

BÀI GIẢNG TOÁN RỜI RẠC

(Discrete Mathematics)

TS. NGUYỄN HOÀNG SƠN

Mục lục

Lời nói đầu	1
Chương 1. Giải tích tổ hợp	2
1.1. Chỉnh hợp lặp	2
1.2. Chỉnh hợp	2
1.3. Hoán vị	3
1.4. Tổ hợp	3
Chương 2. Bài toán đếm	6
2.1. Cơ sở của phép đếm	6
2.1.1. Nguyên lý cộng	6
2.1.2. Nguyên lý nhân	6
2.1.3. Nguyên lý bù trừ	8
2.2. Nguyên lý Dirichlet	10
2.3. Hệ thức truy hồi	12
2.4. Hàm sinh	12
Chương 3. Lý thuyết đồ thị	13
3.1. Khái niệm đồ thị	13
3.1.1. Định nghĩa	13
3.1.2. Đẳng cấu đồ thị	15
3.1.3. Biểu diễn đồ thị bằng ma trận	16
3.1.4. Một số đồ thị tiêu biểu	19
3.2. Các yếu tố cơ bản của đồ thị	19
3.2.1. Đồ thị con, đồ thị bù, đồ thị đối xứng	19
3.2.2. Bậc của đồ thị	21
3.2.3. Đường đi, chu trình và liên thông	23
3.2.4. Bài toán tìm đường đi ngắn nhất	30
3.3. Một số thuật toán tìm kiếm trên đồ thị và ứng dụng	34
3.3.1. Tìm kiếm theo chiều sâu	34
3.3.2. Tìm kiếm theo chiều rộng	36
3.3.3. Bài toán tìm đường đi	37
3.4. Một số loại đơn đồ thị vô hướng	38
3.4.1. Đồ thị đầy đủ	38
3.4.2. Đồ thị chính quy	39
3.4.3. Đồ thị vòng	39
3.4.4. Đồ thị bánh xe	40
3.4.5. Đồ thị lập phương	40

3.4.6.	Đồ thị hai phía	41
3.5.	Hàm Grundy	43
3.5.1.	Định nghĩa	43
3.5.2.	Đồ thị tổng và hàm Grundy	46
3.6.	Chu số của đồ thị	48
3.6.1.	Định nghĩa	48
3.6.2.	Tính chất	48
3.6.3.	Chu trình độc lập	49
3.7.	Các tập hợp đặc biệt trên đồ thị	49
3.7.1.	Tập ổn định trong	50
3.7.2.	Tập ổn định ngoài	52
3.7.3.	Nhân đồ thị	55
3.7.4.	Clique	59
3.7.5.	Phủ đỉnh	60
3.7.6.	Phủ cạnh	61
3.7.7.	Đối sánh	62
3.7.8.	Tập thống trị cạnh	63
3.8.	Đồ thị Euler và đồ thị Hamilton	64
3.8.1.	Đồ thị Euler	64
3.8.2.	Đồ thị Hamilton	71
3.9.	Luồng cực đại	83
3.9.1.	Mạng vận tải, luồng trong mạng vận tải	84
3.9.2.	Định lý Ford-Fulkerson	86
3.9.3.	Thuật toán Ford-Fulkerson tìm luồng cực đại trong mạng vận tải	87
3.10.	Cây	92
3.10.1.	Định nghĩa	92
3.10.2.	Một số tính chất cơ bản	92
3.10.3.	Cây khung	94
3.10.4.	Cây phân cấp	100
3.11.	Đồ thị phẳng	105
3.11.1.	Định nghĩa	106
3.11.2.	Một số tính chất cơ bản	107
3.12.	Tô màu đồ thị	112
3.12.1.	Tô màu bản đồ	112
3.12.2.	Tô màu đồ thị	113
3.12.3.	Tô màu đồ thị phẳng	118

3.12.4. Một số ứng dụng	120
Chương 4. Đại số Boole	122
4.1. Khái niệm đại số Boole	122
4.1.1. Định nghĩa	122
4.1.2. Các hằng đẳng thức của đại số Boole	123
4.2. Hàm Boole	126
4.3. Mạch logic	129
4.3.1. Cổng logic	129
4.3.2. Mạch logic	131
4.4. Cực tiểu hóa các mạch logic	136
4.4.1. Phương pháp bản đồ Karnaugh	137
4.4.2. Phương pháp Quine-McCluskey	140
Tài liệu tham khảo	146

Danh sách hình vẽ

3.1. Đường đi trong đồ thị vô hướng	24
3.2. Đồ thị Petersen	39
3.3. Cách xây dựng dãy tập con $P_0, P_1, \dots, P_k$	45
3.4. Cách xây dựng đồ thị tổng	46
3.5. Bảy chiếc cầu ở thành phố Königsberg và đồ thị tương ứng	64
3.6. Cây lời giải của bài toán người du lịch	83
3.7. Cây phân cấp	100
3.8. Cây có gốc a	100
3.9. Cây phân cấp tổng quát	101
3.10. Cây phi phân	102
3.11. Cây biểu thức tổng quát	103
3.12. Đồ thị ba nhà, ba giếng	106
3.13. Biểu diễn phẳng của hình tứ diện và hình hộp	108
3.14. 5 đỉnh kề với 5 màu	119
3.15. Đồ thị cho bài toán lập lịch	121
4.1. Mạch logic một đường ra	130
4.2. Cổng NOT	130
4.3. Cổng AND	130
4.4. Cổng OR	131
4.5. Cổng NAND	133
4.6. Cổng NAND biểu diễn qua hai cổng AND và NOT	133
4.7. Cổng NOR	134
4.8. Cổng NOR biểu diễn qua hai cổng OR và NOT	134
4.9. Cổng XOR	134
4.10. Cổng XOR vẽ lại	134
4.11. Mạch logic có nhiều đường ra	135
4.12. Mạch logic thực hiện hai hàm $s = x \oplus y$ và $r = x.y$	135
4.13. Mạch cộng hai số 1-bit	136
4.14. Bản đồ Karnaugh hai biến	137
4.15. Ba bản đồ Karnaugh hai biến	138
4.16. Nhóm các tích sơ cấp của các hàm biểu diễn trong bản đồ Karnaugh	138
4.17. Bản đồ Karnaugh ba biến	138
4.18. Bản đồ Karnaugh bốn biến	139

Danh sách bảng

4.1. <i>Bảng cộng hai số 1-bit</i>	135
4.2. <i>Bảng các hàm Boole s, r của x, y</i>	135

Danh sách thuật toán

3.1	(Thuật toán Dijkstra)	32
3.2	Thuật toán DFS	34
3.3	Thuật toán BFS	36
3.4	(Nhận biết đồ thị hai phía)	42
3.5	(Tìm một tập ổn định trong lớn nhất)	52
3.6	(Tìm một tập ổn định ngoài nhỏ nhất)	54
3.7	(Tìm một nhân của đồ thị)	58
3.8	Thuật toán Fleury	68
3.9	Thuật toán Nhánh cận giải bài toán người du lịch	79
3.10	Tìm luồng cực đại trong mạng vận tải nguyên	88
3.11	(Thuật toán Kruskal)	97
3.12	Thuật toán Prim	99
3.13	(Thuật toán Welch-Powell)	115
4.1	(Tìm dạng tổng chuẩn tắc thu gọn)	142
4.2	(Tìm dạng tổng chuẩn tắc cực tiểu)	144

Lời nói đầu

Toán rời rạc là một trong những lĩnh vực của toán học chuyên nghiên cứu về các đối tượng rời rạc. Người ta sử dụng công cụ toán rời rạc khi phải đếm các đối tượng, nghiên cứu quan hệ giữa các tập rời rạc cũng như phân tích các quá trình hữu hạn. Quá trình cất giữ và xử lý thông tin trên máy tính thực chất là quá trình rời rạc, điều này làm toán rời rạc cũng đóng vai trò quan trọng cho ngành khoa học máy tính.

Bài giảng này giới thiệu các kiến thức cơ bản của hai lĩnh vực có nhiều ứng dụng của toán rời rạc là: lý thuyết đồ thị và đại số Boole. Ngoài ra bài giảng còn giới thiệu thêm về giải tích tổ hợp và bài toán đếm, là một phần của lý thuyết tổ hợp trong toán rời rạc.

Cấu trúc của bài giảng được chia làm 4 chương:

Chương 1: giải tích tổ hợp

Chương 2: bài toán đếm

Chương 3: lý thuyết đồ thị

Chương 4: đại số Boole.

Tác giả

Chương 1. Giải tích tổ hợp

Chương này giới thiệu một số cấu hình tổ hợp đơn giản như chỉnh hợp lặp (arrangement with repetition), chỉnh hợp (arrangement), hoán vị (permutation) và tổ hợp (combination). Những cấu hình này thường được dùng làm cơ sở cho phép đếm.

1.1. Chỉnh hợp lặp

Cho tập hợp A gồm n phần tử ($n \neq 0$) và một số tự nhiên k (không nhất thiết $k \leq n$). Một chỉnh hợp lặp chập k của n (của A) là một dãy k phần tử có thể lặp lại của A .

Như vậy có thể thấy một chỉnh hợp lặp chập k của A chính là một phần tử của tích Descartes A^k , hay một ánh xạ từ tập k phần tử vào tập n phần tử. Do đó số chỉnh hợp lặp chập k của n phần tử là n^k .

Ví dụ 1.1. Cho tập $A = \{1, 2, 3\}$. Khi đó các dãy ký tự có hai phần tử có lặp của tập A là 11, 12, 13, 21, 22, 23, 31, 32, 33. Suy ra có tất cả 9 dãy như thế.

1.2. Chỉnh hợp

Cho tập hợp A gồm n phần tử ($n \neq 0$) và một số tự nhiên $k \leq n$. Một chỉnh hợp (không lặp) chập k của n là một dãy k phần tử của A , trong đó mỗi phần tử chỉ được phép có mặt không quá một lần.

Ký hiệu số chỉnh hợp chập k của n phần tử A_n^k .

Ví dụ 1.2. Cho tập $A = \{1, 2, 3\}$. Khi đó các dãy ký tự có hai phần tử không lặp của tập A là 12, 13, 21, 23, 31, 32. Như vậy có tất cả 6 dãy như thế.

Suy ra $A_3^2 = 6$.

Định lý 1.1. Cho tập hợp A có n phần tử và một số tự nhiên $k \leq n$. Khi đó

$$A_n^k = \frac{n!}{(n-k)!}$$

Chứng minh. Trong dãy k phần tử không lặp $a_1 a_2 \dots a_k$ ta có n khả năng chọn a_1 từ tập hợp A . Sau khi a_1 được chọn, ta có $n-1$ khả năng chọn a_2 từ tập hợp

$A \setminus \{a_1\}, \dots$ Tóm lại nếu $a_1, a_2, \dots, a_i$ được chọn thì ta có $n - i$ khả năng chọn a_{i+1} từ tập hợp $A \setminus \{a_1, a_2, \dots, a_i\}$.

Theo tính chất nhân của tổ hợp (sau này chúng ta gọi là nguyên lý nhân) suy ra:

$$A_n^k = n(n-1) \cdots (n-k+1) = \frac{n!}{(n-k)!}$$

□

Ví dụ 1.3. Tính số đơn ánh từ một tập k phần tử A vào một tập n phần tử B ($k \leq n$).

Theo định nghĩa đơn ánh, hai phần tử của A có ảnh khác nhau trong B , suy ra số các đơn ánh phải là A_n^k .

1.3. Hoán vị

Cho tập hợp A gồm n phần tử. Một *hoán vị của A* là một cách sắp xếp các phần tử của A để có được một dãy n phần tử của A .

Ký hiệu P_n là số các hoán vị của tập n phần tử A .

Dễ thấy một hoán vị của A là một dãy n phần tử không lặp của A . Suy ra hoán vị chính là một trường hợp riêng của chỉnh hợp khi $k = n$. Từ đây chúng ta có ngay kết quả sau.

Định lý 1.2. $P_n = 1.2 \dots n = n!$

Ví dụ 1.4. Cho tập $A = \{1, 2, 3\}$. Khi đó $P_3 = 3! = 6$, các hoán vị của A đó là 123, 132, 213, 231, 312, 321.

Có thể đồng nhất một hoán vị của tập A gồm n phần tử với một song ánh từ A lên chính nó. Một song ánh như thế người ta còn gọi là một *phép thế*. Các phép thế có rất nhiều tính chất thú vị, việc nghiên cứu nó đã đóng góp một phần quan trọng trong toán học.

1.4. Tổ hợp

Cho tập hợp A gồm n phần tử và một số tự nhiên k ($k \leq n$). *Tổ hợp chập k của n* là mọi tập hợp con k phần tử của A .

Ví dụ 1.5. Cho tập $A = \{1, 2, 3\}$. Khi đó tổ hợp chập 2 của A là $\{1, 2\}, \{1, 3\}, \{2, 3\}$.

Ký hiệu tổ hợp chập k của n là C_n^k hay $\binom{n}{k}$. Số tổ hợp này gặp khá nhiều trong toán học và nó còn được gọi là *hệ số tổ hợp*.

Định lý 1.3. $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

Chứng minh. Ta tính số các dãy k phần tử không lặp của tập hợp n phần tử A bằng cách chọn ra tập con k phần tử và xét các hoán vị của tập con k phần tử của tập A . Với mỗi tập con k phần tử của A ta có đúng $k!$ dãy k phần tử không lặp của A . Như vậy ta có đẳng thức

$$A_n^k = \binom{n}{k} \cdot k!$$

hay

$$\binom{n}{k} = \frac{A_n^k}{k!} = \frac{n!}{k!(n-k)!}$$

□

Khi nhận xét giá trị của phép chia trong công thức của Định lý 1.3 là một số nguyên, người ta nhận được một kết quả thú vị trong số học là “*tích của k số tự nhiên liên tiếp bao giờ cũng chia hết cho $k!$* .”

Ví dụ 1.6. Có n đội bóng thi đấu vòng tròn. Hỏi chúng ta phải tổ chức bao nhiêu trận đấu tất cả?

Ta có cứ 2 đội bóng thì có một trận. Do đó số trận đấu sẽ bằng số cách chọn 2 đội từ n đội, nghĩa là:

$$\binom{n}{2} = \frac{n(n-1)}{2}.$$

Các hệ số tổ hợp có một số tính chất cơ bản sau.

Mệnh đề 1.4. (1) (Đối xứng) $\binom{n}{k} = \binom{n}{n-k}$

(2) (Điều kiện đầu) $\binom{n}{0} = \binom{n}{n} = 1$

(3) (Công thức đệ quy) $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \quad (0 < k < n)$

Chứng minh. (1) và (2) suy trực tiếp từ định nghĩa tổ hợp.

(3) có thể được chứng minh từ Định lý 1.3. $\square$

Các hệ số tổ hợp còn liên quan chặt chẽ với việc khai triển lũy thừa của một nhị thức, đó là

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k \quad (1.1)$$

Công thức (1.1) này còn được gọi là *khai triển nhị thức Newton*, lúc đó các hệ số tổ hợp cũng còn được gọi là *hệ số nhị thức*. Thông thường khai triển nhị thức Newton được gặp dưới dạng đa thức một ẩn:

$$(x + 1)^n = \binom{n}{0} x^n + \binom{n}{1} x^{n-1} + \cdots + \binom{n}{n-1} x + \binom{n}{n} \quad (1.2)$$

Rất nhiều đẳng thức về hệ số tổ hợp được suy ra từ công thức (1.2). Chẳng hạn trong (1.2) chọn $x = 1$, ta sẽ thu được công thức (1.3) sau:

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n-1} + \binom{n}{n} = 2^n \quad (1.3)$$

Kết quả này có nghĩa số các tập con của một tập có n phần tử bằng 2^n .

Nhiều tính chất của hệ số tổ hợp cũng có thể thu được từ (1.2) bằng cách lấy đạo hàm hoặc tích phân theo x hai vế của đẳng thức này một số hữu hạn lần, sau đó gán cho x những giá trị cụ thể.

Chương 2. Bài toán đếm

2.1. Cơ sở của phép đếm

2.1.1. Nguyên lý cộng

Giả sử có một công đoạn A , mà để thực hiện A ta (chỉ) cần thực hiện một trong k công đoạn rời nhau (tức là không có hai công đoạn nào có thể làm đồng thời) $A_1, A_2, \dots, A_k$, trong đó số cách thực hiện A_i là n_i ($i = 1, 2, \dots, k$). Khi đó số cách thực hiện A là $n_1 + n_2 + \dots + n_k$. Cách tính như thế này còn được gọi là *nguyên lý cộng* hay *quy tắc cộng*.

Ví dụ 2.1. Hãy tính số cách có thể đi từ thành phố Huế ra thành phố Hà nội với ba phương tiện khác nhau là đường bộ, đường thủy và đường hàng không. Giả sử mỗi cách đi từ thành phố Huế ra thành phố Hà nội chỉ có thể là sử dụng một trong ba loại phương tiện trên. Biết rằng có 4 hãng hàng không, 2 hãng tàu thủy và 10 hãng giao thông đường bộ có phương tiện đi lại giữa hai thành phố nói trên.

Rõ ràng bài toán này chúng ta phải sử dụng nguyên lý cộng. Khi đó số cách đi từ thành phố Huế ra thành phố Hà nội là $4 + 2 + 10 = 16$.

Nguyên lý cộng còn có thể phát biểu dưới dạng của ngôn ngữ tập hợp như sau: nếu $A_1, A_2, \dots, A_k$ là các tập hợp đôi một rời nhau thì

$$|A_1 \cup A_2 \cup \dots \cup A_k| = |A_1| + |A_2| + \dots + |A_k|.$$

Một trường hợp riêng hay dùng của nguyên lý cộng là nếu A là một tính chất cho trên tập U thì

$$|A| = U - |\overline{A}|.$$

2.1.2. Nguyên lý nhân

Giả sử có một công đoạn A được thực hiện qua lần lượt k công đoạn $A_1, A_2, \dots, A_k$, trong đó số cách thực hiện A_i là n_i ($i = 1, 2, \dots, k$). Khi đó số cách thực hiện A là thông qua việc thực hiện A_1 , rồi đến thực hiện A_2 , ... cuối cùng đến A_k (kế tiếp nhau) là $n_1 n_2 \dots n_k$. Cách tính như thế này còn được gọi là *nguyên lý nhân* hay *quy tắc nhân*.

Ví dụ 2.2. Giả sử để đi từ thành phố Huế ra đến thành phố Hà nội thì phải đi qua thành phố Vinh. Giả sử rằng đi từ thành phố Huế đến thành phố Vinh có 2

con đường khác nhau, và đi từ thành phố Vinh ra đến thành phố Hà nội có 5 con đường khác nhau. Hãy cho biết số cách có thể đi từ thành phố Huế ra đến thành phố Hà nội.

Bài toán này phải áp dụng nguyên lý nhân. Suy ra số cách đi từ thành phố Huế ra đến thành phố Hà nội là $2.5 = 10$.

Ví dụ 2.3. Có bao nhiêu xâu nhị phân có độ dài n ?

Mỗi một trong n bit của xâu nhị phân có thể chọn bằng hai cách (0 hay 1). Do vậy theo nguyên lý nhân, có tổng cộng 2^n xâu nhị phân khác nhau có độ dài bằng n .

Ví dụ 2.4. Có thể tạo được bao nhiêu ánh xạ từ tập A có k phần tử vào tập B có n phần tử?

Theo định nghĩa ánh xạ, một ánh xạ xác định trên A có giá trị trên B là một phép tương ứng mỗi phần tử của A với một phần tử nào đó của B . Suy ra để chọn ảnh cho mỗi phần tử i ($i = 1, 2, \dots, k$) của A , ta luôn có n cách. Suy ra theo nguyên lý nhân ta có:

$$\underbrace{n.n \dots n}_{k \text{ lần}} = n^k.$$

Ví dụ 2.5. Có bao nhiêu đơn ánh xác định trên tập A có k phần tử và nhận giá trị trên tập B có n phần tử?

Xảy ra hai trường hợp:

- Nếu $k > n$ thì với mọi ánh xạ, có ít nhất hai phần tử của A có cùng một ảnh. Điều này có nghĩa không có đơn ánh nào từ A đến B .
- Nếu $k \leq n$ thì với phần tử thứ nhất của A có n cách chọn ảnh, với phần tử thứ hai có $n - 1$ cách chọn ảnh (do đơn ánh), ... với phần tử thứ j ($j \leq k$) có $n - j + 1$ cách chọn ảnh. Suy ra theo nguyên lý nhân ta có

$$n(n-1)(n-2) \dots (n-k+1) = \frac{n!}{(n-k)!} = A_n^k$$

đơn ánh từ tập A đến tập B .

Nguyên lý nhân cũng có thể phát biểu dưới dạng của ngôn ngữ tập hợp như sau: nếu $A_1, A_2, \dots, A_k$ là các tập hữu hạn, thì

$$|A_1 \times A_2 \times \dots \times A_k| = |A_1| \cdot |A_2| \dots |A_k|.$$

Lưu ý 2.1. Trong nhiều bài toán chúng ta phải áp dụng cả hai nguyên lý kể trên

trong cùng một lúc. Có thể cách dùng rất khác nhau nhưng sẽ luôn có cùng một đáp số.

2.1.3. Nguyên lý bù trừ

Một số bài toán đếm phức tạp hơn phải dựa vào nguyên lý tổng quát của nguyên lý cộng thường được gọi là nguyên lý bù trừ. Ở đây ta phát biểu nguyên lý này theo ngôn ngữ tập hợp. Đầu tiên cho hai tập hợp hữu hạn bất kỳ A và B (không có giả thiết về sự rời nhau giữa hai tập hợp A và B). Khi đó rõ ràng ta có:

$$|A \cup B| = |A| + |B| - |A \cap B|. \quad (2.1)$$

Bằng quy nạp công thức trên có thể được mở rộng cho k tập hợp hữu hạn như sau.

Định lý 2.1. *Giả sử $A_1, A_2, \dots, A_k$ là các tập hữu hạn bất kỳ. Khi đó*

$$|A_1 \cup A_2 \cup \dots \cup A_k| = N_1 - N_2 + \dots + (-1)^{k-1} N_k \quad (2.2)$$

trong đó N_m ($1 \leq m \leq k$) là tổng số phần tử của tất cả các giao m tập lấy từ k tập đã cho, nghĩa là:

$$N_m = \sum_{1 \leq i_1 < i_2 < \dots < i_m \leq k} |A_{i_1} \cap A_{i_2} \cap \dots \cap A_{i_m}|.$$

Bây giờ chúng ta đồng nhất tập A_m ($1 \leq m \leq k$) với tính chất A_m cho trên tập vũ trụ hữu hạn U nào đó và đếm xem có bao nhiêu phần tử của U không thỏa mãn bất kỳ một tính chất A_m nào.

Gọi $\overline{N}$ là số cần đếm và $N = |U|$. Khi đó ta có

$$\begin{aligned} \overline{N} &= N - |A_1 \cup A_2 \cup \dots \cup A_k| \\ &= N - N_1 + N_2 - \dots + (-1)^k N_k \end{aligned} \quad (2.3)$$

trong đó N_m là tổng các phần tử của U thỏa mãn m tính chất lấy từ k tính chất đã cho. Công thức (2.3) này được gọi là *nguyên lý bù trừ*. Nó cho phép tính $\overline{N}$ qua các N_m trong trường hợp các số này dễ tính toán hơn.

Ví dụ 2.6. Có bao nhiêu xâu nhị phân độ dài 10 sao cho hoặc bắt đầu bởi 00 hoặc kết thúc bởi 11?

Dễ thấy số xâu nhị phân độ dài 10 bắt đầu bởi 00 là $2^8 = 256$ và số xâu nhị phân độ dài 10 kết thúc bởi 11 là $2^8 = 256$. Hơn nữa số xâu nhị phân độ dài 10 bắt đầu

bởi 00 và kết thúc bởi 11 là $2^6 = 64$. Khi đó theo nguyên lý bù trừ (2.1) suy ra số xâu nhị phân độ dài 10 sao cho hoặc bắt đầu bởi 00 hoặc kết thúc bởi 11 là $256 + 256 - 64 = 448$.

Ví dụ 2.7. Xét tập số $A = \{1, 2, \dots, 100\}$. Cho biết trong A có bao nhiêu số không chia hết cho bất cứ số nào trong các số 2, 3, 5.

Gọi $B_i = \{x \in A | x \text{ chia hết cho } i\}$, với $i = 1, 2, 3$. Khi đó $B_2 \cup B_3 \cup B_5$ là tập các số trong A chia hết cho ít nhất một trong ba số 2, 3 và 5. Do vậy số lượng các số cần tìm $\overline{N}$ là

$$\begin{aligned}\overline{N} &= N - |B_2 \cup B_3 \cup B_5| \\ &= 100 - (N_1 - N_2 + N_3)\end{aligned}$$

với

$$\begin{aligned}N_1 &= |B_2| + |B_3| + |B_5| \\ &= \left\lfloor \frac{100}{2} \right\rfloor + \left\lfloor \frac{100}{3} \right\rfloor + \left\lfloor \frac{100}{5} \right\rfloor \\ &= 50 + 33 + 20 \\ &= 103\end{aligned}$$

$$\begin{aligned}N_2 &= |B_2 \cap B_3| + |B_2 \cap B_5| + |B_3 \cap B_5| \\ &= \left\lfloor \frac{100}{2 \times 3} \right\rfloor + \left\lfloor \frac{100}{2 \times 5} \right\rfloor + \left\lfloor \frac{100}{3 \times 5} \right\rfloor \\ &= 16 + 10 + 6 \\ &= 32\end{aligned}$$

$$\begin{aligned}N_3 &= |B_2 \cap B_3 \cap B_5| \\ &= \left\lfloor \frac{100}{2 \times 3 \times 5} \right\rfloor \\ &= 3.\end{aligned}$$

Suy ra $\overline{N} = 100 - 103 + 32 - 3 = 26$. Do đó số lượng số không chia hết cho bất cứ số nào trong các số 2, 3, 5 là 26.

2.2. Nguyên lý Dirichlet

Nguyên lý sau được nhà toán học nổi tiếng người Đức J. Dirichlet đề xuất từ thế kỷ 19 và ông đã áp dụng nó để giải rất nhiều bài toán tổ hợp (cũng như các bài toán đếm và bài toán tồn tại).

Mệnh đề 2.2 (Nguyên lý Dirichlet¹). *Nếu có ít nhất $k + 1$ đồ vật được đặt vào trong k hộp thì tồn tại tối thiểu một hộp có ít nhất hai đồ vật.*

Chứng minh. Chứng minh phản chứng. Giả sử không có hộp nào trong k hộp chứa nhiều hơn một đồ vật. Từ đây suy ra tổng số đồ vật được chứa trong các hộp nhiều nhất là k . Điều này mâu thuẫn với giả thiết là có ít nhất $k + 1$ đồ vật. Vậy phải tồn tại tối thiểu một hộp trong k hộp có chứa ít nhất hai đồ vật. $\square$

Lưu ý nguyên lý Dirichlet được ứng dụng trực tiếp nhất là cho các tập hữu hạn (hộp, ngăn kéo, chuồng bồ câu, ...), tuy nhiên nó cũng có thể áp dụng cho các tập vô hạn.

Ví dụ 2.8. Cho biết trong một nhóm bất kỳ có 367 người, có ít nhất bao nhiêu người có cùng ngày sinh.

Vì mỗi người chỉ có duy nhất một ngày sinh, và chúng ta có tất cả 366 ngày sinh khác nhau. Do đó theo nguyên lý Dirichlet sẽ luôn có ít nhất hai người có ngày sinh giống nhau.

Ví dụ 2.9. Trong kỳ thi học sinh giỏi, điểm bài thi được đánh giá bởi một số nguyên trong khoảng từ 0 đến 100. Hỏi phải có ít nhất bao nhiêu học sinh dự thi để chắc chắn có hai học sinh có kết quả thi giống nhau?

Theo nguyên lý Dirichlet số học sinh cần tìm là 102, bởi vì ta có 101 kết quả điểm thi khác nhau.

Thông thường nguyên lý Dirichlet được phát biểu cho trường hợp số đồ vật nhiều hơn số hộp. Khi đó một cách tổng quát nguyên lý Dirichlet được phát biểu lại như sau.

Mệnh đề 2.3 (Nguyên lý Dirichlet tổng quát). *Nếu có n đồ vật được đặt vào trong k hộp ($n > k$) thì sẽ tồn tại tối thiểu một hộp có ít nhất $\lceil \frac{n}{k} \rceil$ đồ vật.*

¹Còn được gọi là *nguyên lý chuồng bồ câu*, *nguyên lý hộp* hay *nguyên lý ngăn kéo Dirichlet*.

² $\lceil \frac{n}{k} \rceil$ là ký hiệu *phần nguyên trần* của $\frac{n}{k}$, đó là số nguyên nhỏ nhất có giá trị lớn hơn hoặc bằng $\frac{n}{k}$, hay còn bằng $\lfloor \frac{n-1}{k} \rfloor + 1$.

Chứng minh. Chứng minh phản chứng.

Giả sử mọi hộp đều chứa ít hơn $\lceil \frac{n}{k} \rceil$ đồ vật. Suy ra tổng số đồ vật trong k hộp là

$$\leq k(\lceil \frac{n}{k} \rceil - 1) < k(\frac{n}{k}) = n.$$

Điều này mâu thuẫn với giả thiết là có n đồ vật. Như vậy phải tồn tại tối thiểu một hộp có ít nhất $\lceil \frac{n}{k} \rceil$ đồ vật. $\square$

Rõ ràng chứng minh của nguyên lý là khá đơn giản, tuy nhiên việc khai thác ứng dụng của nguyên lý lại rất đa dạng và không hề đơn giản. Chúng ta xét một số ví dụ minh họa sau đây.

Ví dụ 2.10. Cho biết trong 50 người, có ít nhất bao nhiêu người sinh cùng một tháng.

Xếp những người sinh cùng một tháng vào một nhóm. Như vậy chúng ta có 12 nhóm tất cả. Do đó theo nguyên lý Dirichlet, tồn tại một nhóm có không ít hơn $\frac{50}{12} = 4,1$ người, nghĩa là 5 người. Suy ra trong 50 người có ít nhất 5 người sinh cùng tháng.

Ví dụ 2.11. Giả sử có 5 loại học bổng khác nhau. Hỏi rằng phải có tối thiểu bao nhiêu sinh viên để chắc chắn có ít nhất 4 người cùng nhận học bổng như nhau.

Số sinh viên ít nhất cần có để đảm bảo chắc chắn có 4 sinh viên cùng nhận học bổng như nhau là số nguyên nhỏ nhất n sao cho $\frac{n}{5} > 3$. Suy ra số nguyên nhỏ nhất đó là $n = 3 \times 5 + 1 = 16$. Như vậy 16 là số sinh viên ít nhất đảm bảo chắc chắn là có 4 sinh viên cùng nhận một loại học bổng.

Lưu ý rằng khi sử dụng nguyên lý Dirichlet để đếm phần tử hay chứng minh sự tồn tại một cấu hình nào đó, thì điều quan trọng là phải xác định được đâu là đồ vật và đâu là cái hộp. Sau đây là một vài ví dụ như vậy.

Ví dụ 2.12. Chứng minh rằng trong một phòng họp (tối thiểu có 2 người) bao giờ cũng tìm được hai người có số người quen trong số những người dự họp là bằng nhau.

Giả sử phòng họp có n người. Suy ra số người quen của mỗi người trong phòng họp nhận các giá trị từ 0 đến $n - 1$. Tuy nhiên rõ ràng trong phòng họp không thể đồng thời có người có số người quen là 0 (tức là không quen ai) và có người có số người quen là $n - 1$ (tức là quen tất cả). Do đó theo số lượng người quen, chúng ta chỉ có thể phân n người ra thành $n - 1$ nhóm. Vì vậy theo nguyên lý Dirichlet tồn tại một nhóm có ít nhất 2 người, tức là luôn tìm được ít nhất 2 người có số người quen là như nhau.

2.3. Hệ thức truy hồi

...

2.4. Hàm sinh

...

Chương 3. Lý thuyết đồ thị

Lý thuyết đồ thị là một lĩnh vực nghiên cứu đã có từ rất lâu, và đặc biệt có nhiều ứng dụng hiện đại. Những tư tưởng cơ bản của lý thuyết đồ thị được đề xuất bởi nhà toán học lỗi lạc người Thụy sĩ Leonhard Euler vào những năm đầu của thế kỷ 18. Chính Euler là người đã sử dụng đồ thị để giải bài toán nổi tiếng về các cây cầu ở thành phố Königsberg.

Đồ thị được sử dụng để giải rất nhiều bài toán trong nhiều lĩnh vực khác nhau.

3.1. Khái niệm đồ thị

3.1.1. Định nghĩa

Đồ thị có hướng (directed graph) là một cặp $G = (V, E)$, trong đó V là tập hữu hạn các phần tử còn gọi là *đỉnh* (vertex) và $E \subseteq V \times V$ là tập các cặp đỉnh còn được gọi là *cạnh có hướng* (directed edge) hay *cung*. Nếu cung $e = (u, v) \in E$ thì ta nói u là *đỉnh đầu* và v là *đỉnh cuối*, hướng của cung e là đi từ đỉnh u đến đỉnh v , hai đỉnh u và v là *liền kề* hay *kề nhau* và cung e *liên thuộc* hay *nối* với các đỉnh u, v .

Đồ thị có hướng cũng có thể định nghĩa bằng ánh xạ kề như sau: $G = (V, F)$ với

- (i) V là tập hữu hạn các đỉnh,
- (ii) $F : V \rightarrow 2^V$ còn được gọi là *ánh xạ kề*.

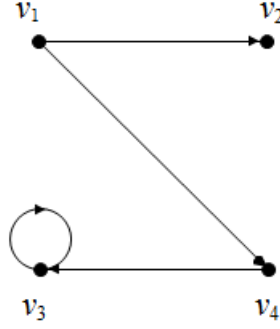
Sự tương đương của hai định nghĩa được thể hiện qua mệnh đề:

$$\forall u, v \in V : (u, v) \in E \Leftrightarrow v \in F(u).$$

Như vậy về bản chất, đồ thị có hướng là một tập các đối tượng được biểu diễn bằng các đỉnh và giữa các đối tượng có một quan hệ hai ngôi biểu diễn bằng các cung. Để trực quan người ta thường biểu diễn đồ thị có hướng trên mặt phẳng như sau: các đỉnh được biểu diễn bằng các điểm, các cung được biểu diễn bằng các đoạn cong với hướng từ đỉnh đầu đến đỉnh cuối. Biểu diễn này còn được gọi là *biểu diễn hình học* của đồ thị.

Ví dụ 3.1. Cho $G = (V, E)$ với $V = \{v_1, v_2, v_3, v_4\}$ và $E = \{(v_1, v_2), (v_1, v_4), (v_4, v_3), (v_3, v_3)\}$.

Khi đó G là một đồ thị có hướng với biểu diễn trên mặt phẳng như sau:



Ảnh xạ kề của G trong trường hợp này là $F(v_1) = \{v_2, v_4\}, F(v_2) = \emptyset, F(v_3) = \{v_3\}, F(v_4) = \{v_3\}$.

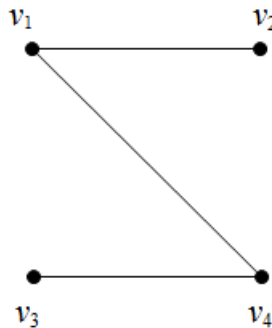
Hai cung bất kỳ được gọi là *kề* nếu chúng có chung một đỉnh. Cung có dạng (v, v) , với $v \in V$, được gọi là *khuyên*. Số đỉnh của G được gọi là *cấp* của G và số cạnh của G được gọi là *cỡ* của G .

Trường hợp $G = (V, E)$ trong đó V là tập hữu hạn các đỉnh và E là tập các tập con có lực lượng là 2 trên V thì G được gọi là *đồ thị vô hướng* (undirected graph). Khi đó các phần tử của E còn được gọi là *cạnh vô hướng* hay *cạnh*. Trường hợp nếu cạnh $e = \{u, v\} \in E$ thì ta nói các đỉnh u, v là các *điểm đầu mút* của e hay e *liên thuộc* với các đỉnh u và v .

Biểu diễn trên mặt phẳng của đồ thị vô hướng tương tự như đồ thị có hướng đó là: các đỉnh được biểu diễn bằng các điểm, các cạnh được biểu diễn bằng các đoạn cong.

Ví dụ 3.2. Cho $G = (V, E)$ với $V = \{v_1, v_2, v_3, v_4\}$ và $E = \{\{v_1, v_2\}, \{v_1, v_4\}, \{v_4, v_3\}\}$.

Khi đó G là một đồ thị vô hướng với biểu diễn trên mặt phẳng là:



Trường hợp nếu tập E chứa cả cạnh lẫn cung thì lúc đó cặp $G = (V, E)$ được gọi là *đồ thị hỗn hợp* hay *đồ thị hỗn tạp*. Rõ ràng mỗi đồ thị vô hướng có thể biểu diễn lại bằng một đồ thị có hướng bằng cách thay mỗi cạnh bằng hai cung. Trong giáo

trình này khi chúng ta đề cập đến đồ thị nhưng không nói rõ là vô hướng hay có hướng thì đồ thị như vậy có thể đồ thị vô hướng và cũng có thể đồ thị có hướng. Ngoài ra để cho gọn từ đây về sau chúng ta vẫn viết mỗi cạnh $\{u, v\}$ trong đồ thị vô hướng bằng cặp (u, v) như đồ thị có hướng.

Đồ thị không có khuyên trong đó mỗi cặp đỉnh được nối với nhau bởi không quá một cạnh được gọi là *đơn đồ thị* (simple graph). Ngược lại nếu đồ thị không có khuyên và có những cặp đỉnh được nối với nhau nhiều hơn một cạnh thì được gọi là *đa đồ thị* (multigraph). Như vậy có thể thấy đồ thị được phân loại dựa theo tính chất cạnh hay cung của chúng. Trường hợp nếu mỗi cạnh hay cung (u, v) của đồ thị $G = (V, E)$ được gán một số nguyên $c(u, v)$, tức là ánh xạ $c : E \rightarrow \mathbb{Z}$, thì đồ thị G như vậy được gọi là *đồ thị có trọng số* và ký hiệu lại là $G = (V, E, c)$.

3.1.2. Đồng cấu đồ thị

Một đồ thị có thể có nhiều biểu diễn trên mặt phẳng khác nhau. Bài toán nhận biết hai biểu diễn trên mặt phẳng khác nhau của cùng một đồ thị là bài toán rất khó. Nếu có hai biểu diễn trên mặt phẳng khác nhau của cùng một đồ thị thì các đỉnh của chúng tương ứng được với nhau, tức là thiết lập được một song ánh giữa hai tập đỉnh của hai biểu diễn trên mặt phẳng, sao cho sự kề nhau của hai đỉnh tương ứng trong hai biểu diễn trên mặt phẳng là hoàn toàn như nhau. Đây chính là khái niệm đồng cấu của đồ thị. Một cách hình thức khái niệm đồng cấu của đồ thị được định nghĩa như sau. Hai đơn đồ thị $G_1 = (V_1, E_1)$ và $G_2 = (V_2, E_2)$ được gọi là *đồng cấu* với nhau, ký hiệu $G_1 \simeq G_2$, nếu tồn tại một song ánh $f : V_1 \rightarrow V_2$ sao cho hai đỉnh u và v kề nhau trong G_1 khi và chỉ khi $f(u)$ và $f(v)$ là kề nhau trong G_2 :

$$\forall u, v \in V_1 : (u, v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2.$$

Khi đó ánh xạ f được gọi là một *phép đồng cấu*.

Như vậy có thể thấy sự đồng cấu của hai đồ thị là dựa trên sự đồng cấu của hai tập đỉnh sao cho sự đồng cấu ấy bảo toàn được các cạnh hay cung của đồ thị. Do đó người ta không phân biệt hai đồ thị đồng cấu với nhau, vì về thực chất chúng chỉ khác nhau về tên gọi của các đỉnh và cách biểu diễn bằng hình vẽ.

Thông thường để chỉ ra hai đơn đồ thị không đồng cấu, chúng ta cần chứng tỏ một trong các điều kiện sau không thỏa mãn:

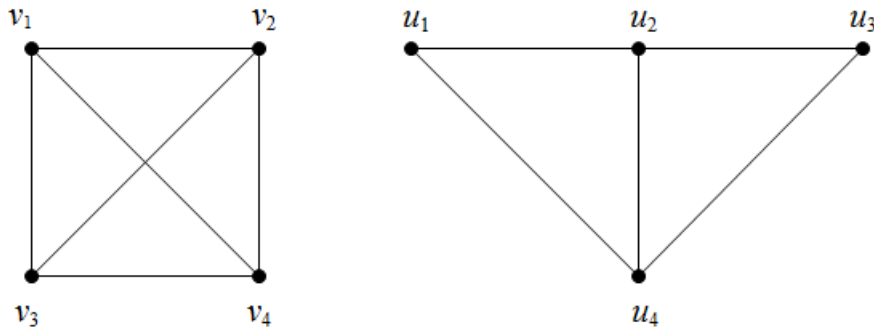
- (1) số đỉnh bằng nhau.
- (2) số cạnh, cung bằng nhau.
- (3) số đỉnh có cùng số cạnh, cung liên thuộc với nó (sau này gọi là *bậc*) là bằng

nhau.

Tuy nhiên đây chỉ là những điều kiện cần, không phải là điều kiện đủ. Để tìm tiêu chuẩn đơn giản và hiệu quả nhận biết hai đồ thị đẳng cấu hiện nay vẫn còn là một bài toán mở của lý thuyết đồ thị.

Ví dụ 3.3. Hai đơn đồ thị vô hướng sau là đẳng cấu với song ánh:

$$f(v_i) = u_i, \forall i = 1, 2, 3, 4$$



3.1.3. Biểu diễn đồ thị bằng ma trận

Người ta thường dùng ma trận để đưa thông tin của đồ thị vào máy tính. Bằng cách này có thể đảm bảo xây dựng lại được đồ thị (đẳng cấu với đồ thị ban đầu) bằng ma trận đó. Có ba cách dùng ma trận để biểu diễn đồ thị.

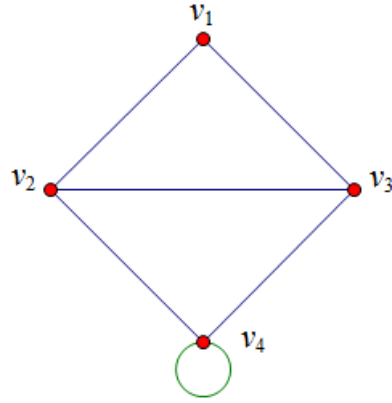
3.1.3.1. Biểu diễn bằng ma trận kề

Cho đồ thị $G = (V, E)$ với $V = \{v_1, v_2, \dots, v_n\}$. Ma trận kề (adjacency matrix) của G là ma trận vuông $A = (a_{ij})_{n \times n} \in M(n, \mathbb{Z})$ trong đó a_{ij} là số cạnh hoặc cung nối từ đỉnh v_i đến đỉnh v_j .

Rõ ràng nếu đồ thị là vô hướng thì ma trận kề tương ứng là ma trận đối xứng. Đặc biệt nếu đồ thị là đơn đồ thị thì ma trận kề tương ứng là một ma trận $(0, 1)$ với các giá trị 0 nằm trên đường chéo chính.

Ưu điểm lớn nhất của phương pháp biểu diễn bằng ma trận kề là để trả lời câu hỏi “hai đỉnh u và v có kề nhau trên đồ thị hay không, chúng ta chỉ thực hiện một phép so sánh”. Tuy nhiên nhược điểm lớn nhất của phương pháp này là “không phụ thuộc vào số cạnh của đồ thị, chúng ta phải sử dụng n^2 đơn vị bộ nhớ để lưu trữ ma trận kề của nó”.

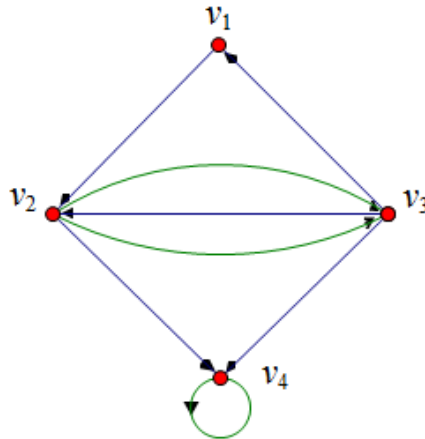
Ví dụ 3.4. (1) Xét đồ thị vô hướng G sau:



Khi đó ma trận kề của G là

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

(2) Xét đồ thị có hướng G như sau:



Khi đó ma trận kề tương ứng là:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

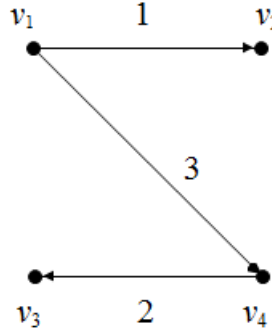
3.1.3.2. Biểu diễn bằng ma trận trọng số

Xét đồ thị có trọng số $G = (V, E, c)$ với $V = \{v_1, v_2, \dots, v_n\}$. Ma trận trọng số của G là ma trận vuông $C = (a_{ij})_{n \times n} \in M(n, \mathbb{Z})$ trong đó:

$$a_{ij} = \begin{cases} c(v_i, v_j) & \text{nếu } (v_i, v_j) \in E \\ \theta & \text{ngược lại} \end{cases}$$

trong đó giá trị $\theta \in \{0, +\infty, -\infty\}$ tùy từng trường hợp cụ thể.

Ví dụ 3.5. Xét đồ thị có hướng có trọng số $G = (V, E, c)$ như sau:



Khi đó ma trận trọng số của G là:

$$C = \begin{pmatrix} 0 & 1 & \infty & 3 \\ \infty & 0 & \infty & \infty \\ \infty & \infty & 0 & \infty \\ \infty & \infty & 2 & 0 \end{pmatrix}$$

ở đây ta cho $a_{ii} = 0$ và $a_{ij} = \infty$ nếu $(v_i, v_j) \notin E$.

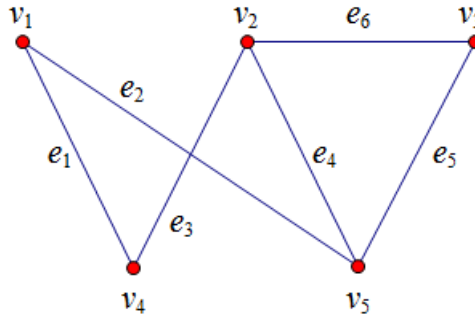
Ưu nhược điểm của biểu diễn này hoàn toàn tương tự như biểu diễn bằng ma trận kề.

3.1.3.3. Biểu diễn bằng ma trận liên thuộc

Cho đồ thị vô hướng $G = (V, E)$ với $V = \{v_1, v_2, \dots, v_n\}$ và $E = \{e_1, e_2, \dots, e_m\}$. Khi đó *ma trận liên thuộc* (incidence matrix) của G là ma trận $I = (a_{ij})_{n \times m} \in M(n \times m, \mathbb{Z})$

$$a_{ij} = \begin{cases} 1 & \text{nếu cạnh } e_j \text{ liên thuộc với đỉnh } v_i \\ 0 & \text{ngược lại.} \end{cases}$$

Ví dụ 3.6. Xét đồ thị vô hướng G sau:



Khi đó ma trận liên thuộc của G là

$$I = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

3.1.4. Một số đồ thị tiêu biểu

Ví dụ 3.7 (Đồ thị lẫn tổ trong sinh thái học). Đồ thị vô hướng được sử dụng trong nhiều mô hình có tính đến sự tương tác của các loài vật. Chẳng hạn sự cạnh tranh của các loài trong một hệ sinh thái có thể mô hình hóa bằng *đồ thị lẫn tổ*, đó là một đồ thị vô hướng trong đó mỗi loài được biểu diễn bằng một đỉnh, một cạnh vô hướng nối hai đỉnh nếu hai loài được biểu diễn bằng các đỉnh này là cạnh tranh với nhau.

Ví dụ 3.8 (Đồ thị ảnh hưởng). Khi nghiên cứu tính cách của một nhóm người, chúng ta nhận thấy một số người có thể có ảnh hưởng lên suy nghĩ của những người khác. Do đó mô hình bài toán này có thể dùng *đồ thị ảnh hưởng*, đó là một đồ thị có hướng trong đó mỗi người của nhóm được biểu diễn bằng một đỉnh và khi một người được biểu diễn bằng đỉnh a có ảnh hưởng lên người được biểu diễn bằng đỉnh b thì có một cung nối từ đỉnh a đến đỉnh b .

3.2. Các yếu tố cơ bản của đồ thị

3.2.1. Đồ thị con, đồ thị bù, đồ thị đối xứng

Cho đồ thị $G = (V, E)$. Đồ thị $G' = (V', E')$ được gọi là *đồ thị con* (subgraph) của G nếu $V' \subseteq V$ và $E' \subseteq E$. Có hai loại đồ thị con quan trọng thường được nghiên cứu, một là đồ thị con bao trùm bởi một tập con các cạnh cho trước và hai là đồ

thị con cảm sinh bởi một tập con các đỉnh cho trước. Các đồ thị con này được định nghĩa một cách hình thức như sau.

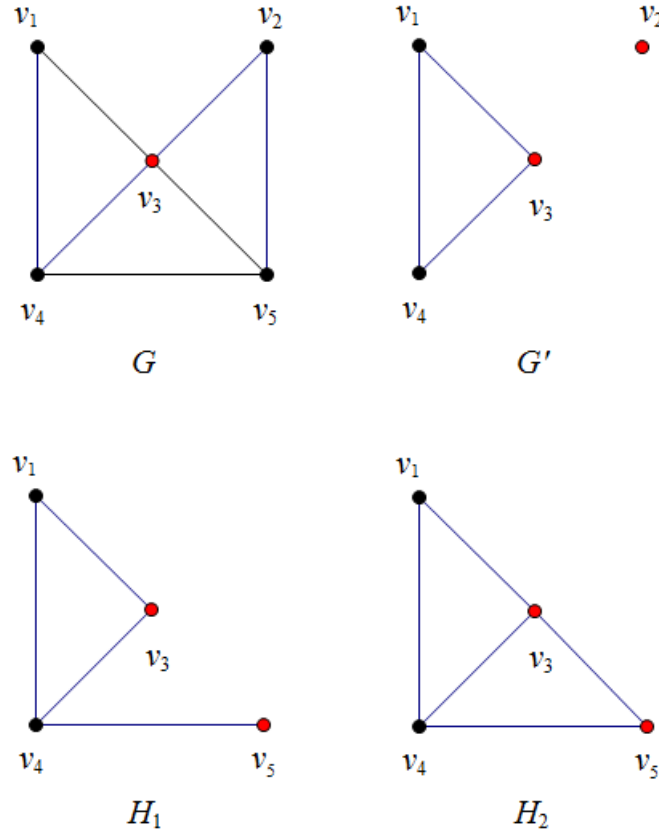
Xét tập con các cạnh $E' \subseteq E$, tập con các đỉnh $V' \subseteq V$ và các tương ứng

$$V_{E'} = \{v \in V : v \text{ là một điểm đầu mút của một cạnh nào đó trong } E'\},$$

$$E_{V'} = \{(u, v) \in E : u, v \in V'\}.$$

Khi đó, đồ thị $H_1 = (V_{E'}, E')$ được gọi là *đồ thị con bao trùm* (spanning subgraph) bởi E' và đồ thị $H_2 = (V', E_{V'})$ được gọi là *đồ thị con cảm sinh* (induced subgraph) bởi V' . Đồ thị con cảm sinh bởi V' của G còn được ký hiệu là $G[V']$. Như vậy, $G[V'] = (V', E_{V'})$. Trường hợp đặc biệt khi $V_{E'} = V$ thì đồ thị con bao trùm $H_1 = (V_{E'}, E')$ còn có thể được hiểu là giữ nguyên số đỉnh và bỏ đi một số cạnh của đồ thị G . Rõ ràng, đồ thị con cảm sinh là đồ thị con bao trùm, tuy nhiên ngược lại không đúng.

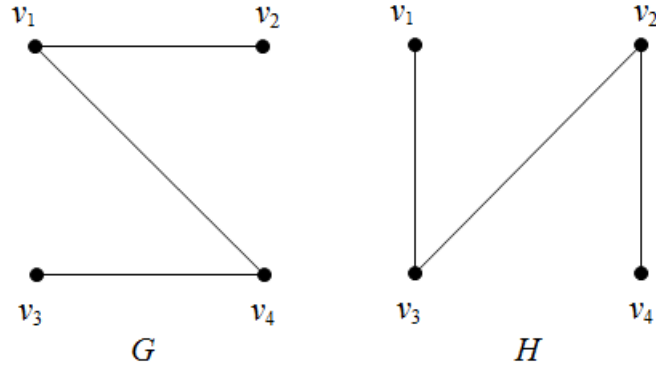
Ví dụ 3.9. Sau đây là đồ thị vô hướng G cùng với các loại đồ thị con của nó:



Rõ ràng, G' là đồ thị con của G , H_1 là đồ thị con bao trùm bởi tập cạnh $\{(v_1, v_3), (v_1, v_4), (v_3, v_4), (v_4, v_5)\}$ và H_2 là đồ thị con cảm sinh bởi tập đỉnh $\{v_1, v_3, v_4, v_5\}$. Ví dụ 3.9 cũng cho thấy không phải mọi đồ thị con là đồ thị con cảm sinh.

Đồ thị bù¹ (complement graph) của đồ thị vô hướng $G = (V, E)$ là một đồ thị vô hướng H có cùng tập đỉnh với G và sao cho hai đỉnh phân biệt bất kỳ của H kề nhau khi và chỉ khi chúng không kề nhau trong G .

Ví dụ 3.10. Đồ thị H sau là đồ thị bù của đồ thị G :



Đồ thị $G = (V, E)$ được gọi là đồ thị đối xứng nếu

$$\forall u, v \in V : (u, v) \in E \Leftrightarrow (v, u) \in E.$$

Như vậy, đồ thị vô hướng là đồ thị đối xứng.

3.2.2. Bậc của đồ thị

Số cạnh liên thuộc với đỉnh v trong đồ thị vô hướng $G = (V, E)$ được gọi là bậc (degree) của đỉnh v và ký hiệu $\deg(v)$. Nói một cách khác

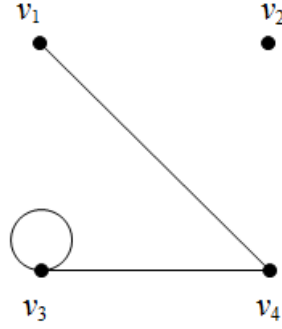
$$\deg(v) = |F(v)|.$$

Trường hợp khuyên tại một đỉnh được tính thêm hai lần cho bậc của đỉnh đó. Đỉnh v gọi là đỉnh treo nếu $\deg(v) = 1$, và gọi là đỉnh cô lập nếu $\deg(v) = 0$. Cạnh có một đỉnh là treo được gọi là cạnh treo. Bậc lớn nhất (tương ứng nhỏ nhất) của các đỉnh trong G được gọi là bậc cực đại (tương ứng cực tiểu) của G và ký hiệu $\Delta(G)$ (tương ứng $\delta(G)$).

Trong đơn đồ thị vô hướng, rõ ràng $\deg(v) \leq |V| - 1$ với mọi $v \in V$.

Ví dụ 3.11. Xét đồ thị vô hướng G như sau:

¹Một số tài liệu còn gọi là đồ thị nghịch đảo (inverse graph).



Ta có $\deg(v_1) = 1, \deg(v_2) = 0, \deg(v_3) = 3, \deg(v_4) = 2$. Suy ra $\Delta(G) = 3$, $\delta(G) = 0$, v_1 là đỉnh treo và v_2 là đỉnh cô lập.

Bậc của đỉnh có một số tính chất cơ bản sau.

Mệnh đề 3.1 (Mệnh đề bắt tay). Cho đồ thị vô hướng $G = (V, E)$. Khi đó

$$\sum_{v \in V} \deg(v) = 2|E|$$

Chứng minh. Rõ ràng, mỗi cạnh $e = (u, v) \in E$ được tính một lần trong $\deg(u)$ và một lần trong $\deg(v)$. Như vậy, tổng tất cả các bậc của các đỉnh phải bằng hai lần số cạnh. $\square$

Hệ quả 3.1. Số đỉnh bậc lẻ của một đồ thị vô hướng luôn là một số chẵn.

Chứng minh. Gọi V_1 và V_2 tương ứng là tập các đỉnh bậc chẵn và tập các đỉnh bậc lẻ của đồ thị vô hướng $G = (V, E)$. Khi đó, theo Mệnh đề 3.1:

$$\sum_{v \in V_1} \deg(v) + \sum_{v \in V_2} \deg(v) = 2|E|.$$

Trong công thức trên ta thấy vế phải là một số chẵn và tổng thứ nhất cũng là một số chẵn nên tổng thứ hai là một số chẵn. Vì $\deg(v)$ là lẻ với mọi $v \in V_2$ nên $|V_2|$ phải là một số chẵn. $\square$

Mệnh đề 3.2. Trong một đơn đồ thị vô hướng, luôn tồn tại hai đỉnh có cùng bậc.

Chứng minh. Xét đơn đồ thị vô hướng $G = (V, E)$ với $|V| = n$. Rõ ràng, phát biểu trên có thể chuyển về bài toán “trong một phòng họp có n người bao giờ cũng tìm được hai người có số người quen trong số những người dự họp là bằng nhau”. Như vậy, tính đúng của bài toán này đã được chứng minh trong Ví dụ 2.12. $\square$

Đối với đồ thị có hướng, chúng ta cũng có những khái niệm tương tự. *Bậc ra* (tương ứng *bậc vào*) của đỉnh v trong đồ thị có hướng $G = (V, E)$, ký hiệu $\deg^+(v)$ (tương ứng $\deg^-(v)$), là số cung của G đi ra khỏi v (tương ứng đi vào v).

Theo quan điểm ánh xạ kề ta có:

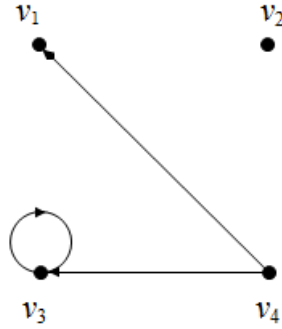
$$\deg^+(v) = |\{w \in V : (v, w) \in E\}| = |F(v)|$$

và

$$\deg^-(v) = |\{u \in V : (u, v) \in E\}| = |\{u \in V : v \in F(u)\}|.$$

Đỉnh v gọi là *đỉnh treo* nếu $\deg^+(v) = 0$ và $\deg^-(v) = 1$. Trường hợp, nếu $\deg^+(v) = \deg^-(v) = 0$, thì v được gọi là *đỉnh cô lập*. Cung có đỉnh treo được gọi là *cung treo*.

Ví dụ 3.12. Xét đồ thị có hướng G như sau:



Ta có $\deg^+(v_1) = 0, \deg^-(v_1) = 1, \deg^+(v_2) = 0, \deg^-(v_2) = 0, \deg^+(v_3) = 1, \deg^-(v_3) = 2, \deg^+(v_4) = 2, \deg^-(v_4) = 0$. Nên v_2 là đỉnh cô lập, v_1 là đỉnh treo và (v_4, v_1) là cung treo.

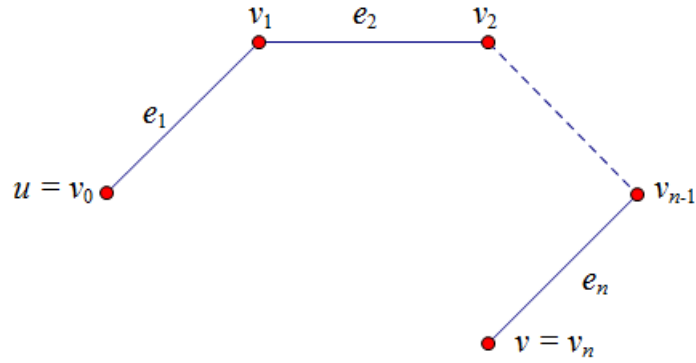
Mệnh đề 3.3. Cho đồ thị có hướng $G = (V, E)$. Khi đó

$$\sum_{v \in V} \deg^+(v) = \sum_{v \in V} \deg^-(v) = |E|.$$

Chứng minh. Do mỗi cung được tính bậc một lần cho đỉnh đầu và một lần cho đỉnh cuối, nên ta có ngay điều cần chứng minh. $\square$

3.2.3. Đường đi, chu trình và liên thông

Đường đi (path) độ dài n ($n \in \mathbb{Z}_0^+$) từ đỉnh u đến đỉnh v trong đồ thị $G = (V, E)$ là một dãy n cạnh hay cung $e_1, e_2, \dots, e_n$ của G sao cho $e_1 = (u, v_1), e_2 = (v_1, v_2), \dots, e_n = (v_{n-1}, v)$ hoặc một dãy $n+1$ đỉnh $v_0, v_1, \dots, v_n$ sao cho $u = v_0, v = v_n$ và $(v_i, v_{i+1}) \in E$ với $i = 0, 1, \dots, n-1$.

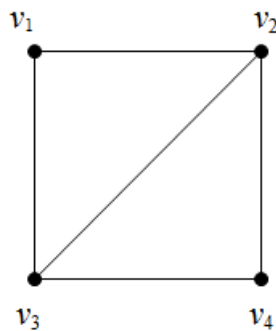


Hình 3.1: Đường đi trong đồ thị vô hướng

Trường hợp nếu đường đi bắt đầu và kết thúc tại cùng một đỉnh thì đường đi đó được gọi là *chu trình* (cycle). Đường đi hay chu trình được gọi là *đơn* nếu nó không chứa cùng một cạnh hay cung quá một lần. Một đường đi (tương ứng chu trình) không đi qua đỉnh nào quá một lần (trừ đỉnh đầu và đỉnh cuối của chu trình là trùng nhau) được gọi là đường đi (tương ứng chu trình) *sơ cấp*. Một chu trình đi qua mọi cạnh hay cung của G được gọi là một *hành trình* trong G . Rõ ràng, một đường đi (tương ứng chu trình) sơ cấp là đường đi (tương ứng chu trình) đơn. Tuy nhiên, ngược lại không đúng.

Hai đường đi (tương ứng chu trình) được gọi là *rời nhau* nếu chúng không có cạnh hay cung chung. Để dễ hình dung người ta thường gọi chu trình có độ dài 3, 4, 5, ..., n trong đồ thị vô hướng là chu trình tam giác, tứ giác, ngũ giác, ..., n -giác. Lưu ý, một số tác giả còn gọi một đường đi trong đồ thị vô hướng là một *xích* hay *dây chuyền*, và một chu trình trong đồ thị có hướng là một *vòng*.

Ví dụ 3.13. Xét đồ thị vô hướng G như sau:



Ta có v_1, v_3, v_4, v_2, v_3 là một đường đi đơn nhưng không sơ cấp độ dài 4; v_1, v_3, v_4, v_2 là một đường đi sơ cấp và do đó là đường đi đơn có độ dài 3. v_1, v_4, v_3, v_2 không phải là đường đi vì (v_1, v_4) không phải là một cạnh; v_1, v_3, v_4, v_2, v_1 là một chu trình sơ cấp và do đó là chu trình đơn. $v_1, v_3, v_4, v_2, v_3, v_1$ là một chu trình không đơn và do đó cũng không phải chu trình sơ cấp.

Xét đồ thị vô hướng $G = (V, E)$. Hai đỉnh của G được gọi là *liên thông*, nếu trong G có một đường đi nối chúng với nhau. Trường hợp, nếu mọi cặp đỉnh của G là liên thông, thì G được gọi là *đồ thị vô hướng liên thông*.

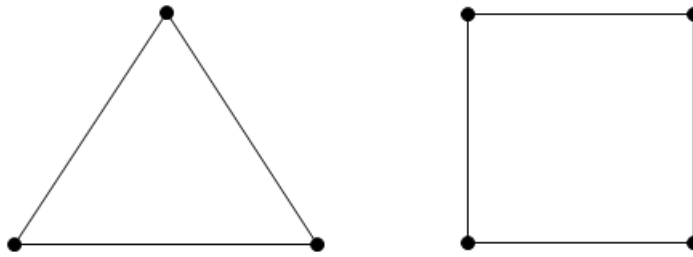
Liên thông là một khái niệm quan trọng trong lý thuyết đồ thị. Rõ ràng từ định nghĩa dễ dàng kiểm tra được quan hệ liên thông giữa các đỉnh là một quan hệ tương đương trên tập các đỉnh. Quan hệ tương đương này chia tập đỉnh của G thành các lớp tương đương thỏa hai tính chất sau:

- (1) Các đỉnh thuộc cùng một lớp thì liên thông với nhau
- (2) Các đỉnh không thuộc cùng một lớp thì không liên thông với nhau.

Các lớp đỉnh tương đương này được gọi là các *thành phần liên thông* (connected component) của G . Ký hiệu $\omega(G)$ là *số thành phần liên thông* của G .

Dễ thấy, G liên thông khi và chỉ khi $\omega(G) = 1$. Ngoài ra, nếu G có n đỉnh cô lập thì $\omega(G) = n$.

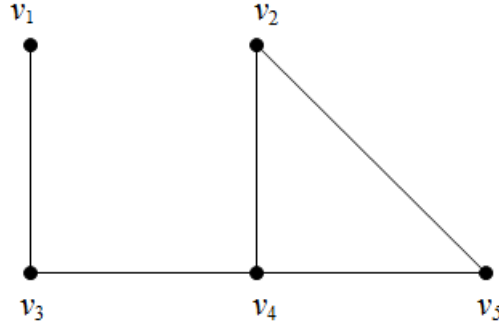
Ví dụ 3.14. Đồ thị vô hướng G sau có 2 thành phần liên thông:



Đỉnh v trong đồ thị G được gọi là *điểm khớp* (articulation point) hay *đỉnh cắt* (cut vertex, *đỉnh rẽ nhánh*) nếu loại bỏ nó cùng với các cạnh liên thuộc với nó khỏi G thì ta nhận được đồ thị con mới có nhiều thành phần liên thông hơn G . Cạnh e của G được gọi là *cầu* (bridge) hay *cạnh cắt* (cut edge) nếu loại bỏ nó khỏi G sẽ làm tăng số thành phần liên thông của G .

Từ định nghĩa có thể thấy các đỉnh cô lập không phải là điểm khớp, một đồ thị có thể không có điểm khớp nào, và số đỉnh không phải điểm khớp trong một đồ thị không liên thông bằng tổng số đỉnh không phải điểm khớp trong các thành phần liên thông của nó. Ngoài ra, một đồ thị bất kỳ có $n \geq 2$ đỉnh thì luôn có ít nhất 2 đỉnh không phải điểm khớp.

Ví dụ 3.15. Xét đồ thị vô hướng G như sau:



Đồ thị G có các điểm khớp là v_3, v_4 và các cầu là $(v_1, v_3), (v_3, v_4)$.

Khái niệm liên thông trong đồ thị vô hướng có một số tính chất cơ bản sau.

Mệnh đề 3.4. *Nếu G là một đồ thị vô hướng liên thông, thì giữa mọi cặp đỉnh phân biệt của G luôn có một đường đi sơ cấp.*

Chứng minh. Giả sử u và v là hai đỉnh phân biệt của đồ thị vô hướng liên thông G . Vì G liên thông nên có ít nhất một đường đi giữa u và v . Gọi $v_0, v_1, \dots, v_n$, với $v_0 = u$ và $v_n = v$, là dãy các đỉnh của đường đi có độ dài ngắn nhất nối u và v . Đây chính là đường đi sơ cấp cần tìm. Thật vậy giả sử nó không phải là đường đi sơ cấp. Suy ra, tồn tại hai đỉnh v_i và v_j sao cho $v_i = v_j$ với $0 \leq i < j \leq n$. Điều này có nghĩa giữa hai đỉnh u và v có đường đi ngắn hơn qua các đỉnh $v_0, v_1, \dots, v_{i-1}, v_j, \dots, v_n$, đường đi ngắn hơn này nhận được bằng cách xóa đi các cạnh tương ứng với dãy các đỉnh $v_i, \dots, v_{j-1}$. Mâu thuẫn. $\square$

Mệnh đề 3.5. *Nếu đơn đồ thị vô hướng G có $n \geq 2$ đỉnh sao cho tổng bậc của hai đỉnh tùy ý không nhỏ hơn n thì G là đồ thị liên thông.*

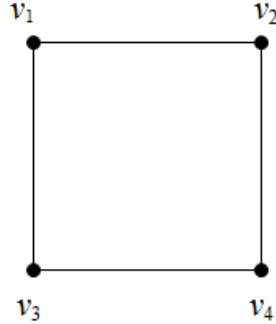
Chứng minh. Giả sử $G = (V, E)$ là một đơn đồ thị vô hướng có n đỉnh ($n \geq 2$) thỏa mãn yêu cầu của bài toán. Chứng minh phản chứng, giả sử G không liên thông, tức là tồn tại hai đỉnh u và v sao cho không có đường đi nối u và v . Suy ra trong G tồn tại hai thành phần liên thông G_1 và G_2 sao cho G_1 có n_1 đỉnh và chứa u , G_2 chứa có n_2 đỉnh và chứa v . Hơn nữa $n_1 + n_2 \leq n$. Khi đó

$$\begin{aligned} \deg(u) + \deg(v) &\leq (n_1 - 1) + (n_2 - 1) \\ &\leq n_1 + n_2 - 2 \\ &\leq n - 2 \\ &< n. \end{aligned}$$

Điều này mâu thuẫn với giả thiết. $\square$

Hệ quả 3.2. Nếu đơn đồ thị vô hướng G có $n \geq 2$ đỉnh sao cho bậc của mỗi đỉnh không nhỏ hơn $n/2$ thì G là đồ thị liên thông.

Ví dụ 3.16. Xét đơn đồ thị vô hướng G có 4 đỉnh như sau:

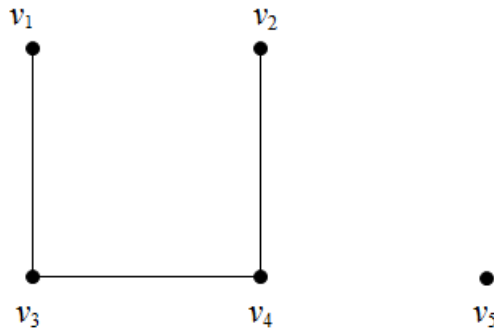


Rõ ràng, G là liên thông theo Mệnh đề 3.5 hoặc Hệ quả 3.2.

Mệnh đề 3.6. Nếu đồ thị vô hướng có đúng hai đỉnh bậc lẻ thì hai đỉnh này phải liên thông.

Chứng minh. Giả sử $G = (V, E)$ là đồ thị vô hướng có đúng hai đỉnh bậc lẻ u và v sao cho chúng không liên thông với nhau. Suy ra G có hai thành phần liên thông G_1 và G_2 sao cho G_1 chứa u và G_2 chứa v . Rõ ràng, bậc của đỉnh u trong G_1 cũng chính là bậc của u trong G , nên trong G_1 đỉnh u vẫn có bậc lẻ và G_1 có duy nhất một đỉnh bậc lẻ. Điều này mâu thuẫn với tính chất số đỉnh bậc lẻ của một đồ thị luôn là chẵn. Vậy hai đỉnh u và v phải liên thông. $\square$

Ví dụ 3.17. Xét đồ thị vô hướng G như sau:



Ta có hai đỉnh v_1 và v_2 là liên thông với nhau theo Mệnh đề 3.6.

Kết quả sau là một điều kiện cần và đủ để một đỉnh trong một đồ thị vô hướng liên thông là điểm khớp.

Mệnh đề 3.7. *Giả sử $G = (V, E)$ là một đồ thị vô hướng liên thông. Khi đó một đỉnh của G là điểm khớp khi và chỉ khi trong G tồn tại hai đỉnh u và v sao cho mỗi đường đi nối u và v đều phải đi qua đỉnh này.*

Chứng minh. ($\Rightarrow$) Giả sử đỉnh x là một điểm khớp trong G . Khi đó, đồ thị con H của G nhận được bằng cách xoá x và các cạnh liên thuộc với x là không liên thông. Giả sử H_1 và H_2 là hai trong số các thành phần liên thông của H . Lấy u là đỉnh trong H_1 và v là đỉnh trong H_2 . Do u và v thuộc hai thành phần liên thông khác nhau nên trong H các đỉnh u và v không liên thông. Nhưng trong G các đỉnh u và v lại liên thông, do đó suy ra mọi đường đi nối u và v đều phải đi qua đỉnh x .

($\Leftarrow$) Giả sử trong G có hai đỉnh u và v sao cho mọi đường đi nối hai đỉnh u và v đều đi qua đỉnh x . Suy ra, nếu bỏ đỉnh x và các cạnh liên thuộc với x thì đồ thị con H nhận được từ G chứa hai đỉnh u, v và hai đỉnh u, v này là không liên thông. Do đó, H là đồ thị không liên thông. Theo định nghĩa điều này có nghĩa đỉnh x là một điểm khớp của G . $\square$

Xét đồ thị G ở Ví dụ 3.15 trên, ta có v_2 là một điểm khớp của G theo Mệnh đề 3.7. Thật vậy, chẳng hạn tồn tại hai đỉnh v_1 và v_3 sao cho mọi đường đi nối v_1 và v_3 đều qua đỉnh v_2 (có đúng hai đường đi như thế là v_1, v_2, v_3 và v_1, v_2, v_4, v_3).

Số đỉnh, số cạnh và số thành phần liên thông trong một đồ thị vô hướng có mối quan hệ sau đây.

Định lý 3.8. *Cho G là một đơn đồ thị vô hướng có n đỉnh, m cạnh và $\omega(G) = k$. Khi đó*

$$n - k \leq m \leq \frac{(n - k)(n - k + 1)}{2}.$$

Chứng minh. Đầu tiên chúng ta chứng minh bất đẳng thức $m \geq n - k$ bằng quy nạp theo số cạnh m . Thật vậy khi số cạnh $m = 0$ thì $k = n$, và do đó rõ ràng bất đẳng thức luôn đúng. Giả sử bất đẳng thức đúng đến $m = i - 1$ cạnh, với $i \geq 2$. Gọi G_1 là đồ thị con bao trùm của G có số cạnh ít nhất i_1 sao cho nó vẫn có k thành phần liên thông. Như vậy, lúc này việc loại bỏ bất kỳ cạnh nào trong G_1 cũng làm tăng số thành phần liên thông của G_1 lên 1 và khi đó đồ thị thu được sẽ có n đỉnh, $k + 1$ thành phần liên thông và $i_1 - 1$ cạnh. Theo giả thiết quy nạp, ta có $i_1 - 1 \geq n - (k + 1)$ hay $i_1 \geq n - k$. Suy ra $i \geq n - k$, tức là bất đẳng thức đúng với $m = i$.

Bây giờ, ta chứng minh bất đẳng thức còn lại của định lý. Thật vậy bổ sung cạnh vào G để nhận được đồ thị G_2 có i_2 cạnh k thành phần liên thông trong đó mỗi thành phần liên thông là những đồ thị mà hai đỉnh bất kỳ nào cũng kề nhau (đồ

thị như thế này còn được gọi là *đồ thị đầy đủ*). Khi đó, $m \leq i_2$, và do đó chỉ cần chứng minh được bất đẳng thức sau là xong

$$i_2 \leq \frac{(n-k)(n-k+1)}{2}.$$

Gọi H_1 và H_2 là hai thành phần liên thông của G_2 với n_1 và n_2 đỉnh sao cho

$$n_1 \geq n_2 > 1 \quad (3.1)$$

Nếu ta thay H_1 và H_2 bằng các đồ thị đầy đủ tương ứng có $n_1 + 1$ và $n_2 - 1$ đỉnh thì tổng số đỉnh không thay đổi nhưng số cạnh tăng thêm một lượng là:

$$\left[\frac{(n_1 + 1)n_1}{2} - \frac{n_1(n_1 - 1)}{2} \right] - \left[\frac{n_2(n_2 - 1)}{2} - \frac{(n_2 - 1)(n_2 - 2)}{2} \right] = n_1 - n_2 + 1.$$

Lặp lại thủ tục này cho khi nào có hai thành phần liên thông thỏa điều kiện (3.1). Vì vậy i_2 là lớn nhất (n, k là cố định) khi đồ thị gồm $k - 1$ đỉnh cô lập và một đồ thị đầy đủ gồm $n - k + 1$ đỉnh. Từ đây, suy ra bất đẳng thức cần chứng minh. $\square$

Hệ quả 3.3. Nếu đơn đồ thị vô hướng G có n đỉnh và m cạnh sao cho

$$m > \frac{(n-2)(n-1)}{2}$$

thì G là liên thông.

Chứng minh. Theo Định lý 3.8 ta có

$$m \leq \frac{(n-k)(n-k+1)}{2}$$

với $\omega(G) = k$. Từ bất đẳng thức này và giả thiết ta thu được:

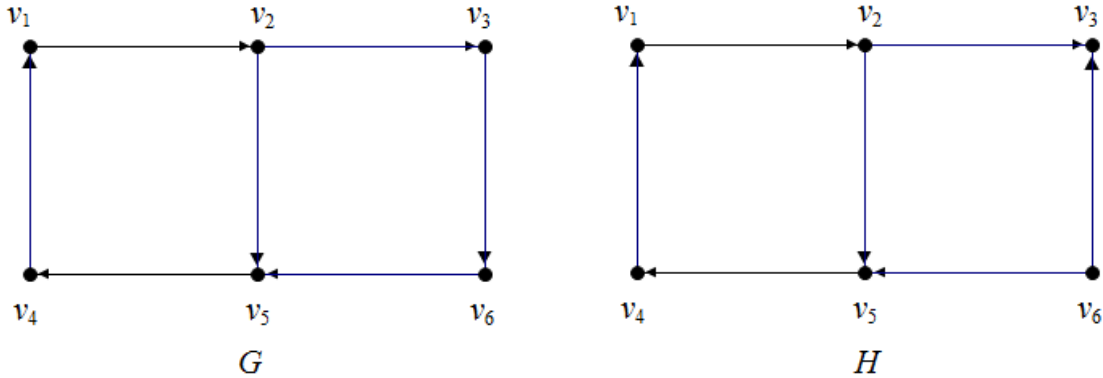
$$\frac{(n-2)(n-1)}{2} < m \leq \frac{(n-k)(n-k+1)}{2}.$$

Bất đẳng thức trên chỉ xảy ra khi $k = 1$. Tức là G liên thông. $\square$

Đối với đồ thị có hướng thì khái niệm liên thông phụ thuộc vào việc ta có xét đến hướng trên các cung hay không. Đồ thị có hướng G được gọi là *liên thông mạnh*

nếu giữa hai đỉnh bất kỳ của G luôn tồn tại một đường đi. Trường hợp, nếu đồ thị vô hướng của đồ thị có hướng G là liên thông thì G được gọi là *liên thông yếu*. Rõ ràng, nếu đồ thị là liên thông mạnh thì cũng là liên thông yếu, tuy nhiên điều ngược lại không đúng.

Ví dụ 3.18. Xét hai đồ thị có hướng G và H sau:



Ta có G là đồ thị liên thông mạnh và H là đồ thị liên thông yếu.

3.2.4. Bài toán tìm đường đi ngắn nhất

3.2.4.1. Phát biểu bài toán

Xét đồ thị có hướng, có trọng số không âm $G = (V, E, c)$ với n đỉnh. Hãy tìm đường đi có trọng số nhỏ nhất từ đỉnh xuất phát $s \in V$ đến đỉnh đích (cuối) $t \in V$. Đường đi có trọng số nhỏ nhất như trên còn được gọi là *đường đi ngắn nhất*. Bài toán tìm đường đi ngắn nhất này thuộc một trong những bài toán tối ưu quan trọng của lý thuyết đồ thị, nó thường được áp dụng để giải quyết rất nhiều bài toán trong thực tế như điều khiển tối ưu, giao thông vận tải, mạng viễn thông, ... Có nhiều thuật toán giải quyết bài toán này như Thuật toán Dijkstra, Thuật toán Ford-Bellman, ... Dưới đây chúng ta sẽ trình bày một số thuật toán như vậy.

3.2.4.2. Thuật toán Dijkstra²

Đường đi ngắn nhất có một tính chất cơ bản sau, tính chất này hàm chứa một nội dung rất sâu sắc và người ta thường gọi nó là nguyên lý tối ưu.

Nguyên lý tối ưu: Mọi đường đi con của đường đi ngắn nhất cũng là đường đi ngắn nhất.

²Edsger Wybe Dijkstra (1930-2002): nhà khoa học máy tính Hà Lan. Ông nhận giải thưởng Turing năm 1972.

Ký hiệu $d(s, v)$ là khoảng cách từ đỉnh s đến đỉnh v . Vì đồ thị G có các cung với trọng số không âm, nên đường đi ngắn nhất từ đỉnh s đến đỉnh t luôn có thể tìm được một cách dễ dàng nếu biết được khoảng cách từ s đến t ($s \neq t$) như sau: luôn tồn tại một đỉnh v đứng ngay trước đỉnh t trong đường đi ngắn nhất từ s đến t sao cho

$$d(s, t) = d(s, v) + c(v, t).$$

Tương tự, có thể tìm được một đỉnh u thỏa tính chất như trên sao cho

$$d(s, v) = d(s, u) + c(u, v).$$

Bởi V hữu hạn và các trọng số của cung là không âm nên rõ ràng phải tồn tại một dãy hữu hạn các đỉnh không có chứa đỉnh lặp lại $t, v, u, \dots, s$. Dãy các đỉnh ngược lại $s, \dots, u, v, t$ chính là đường đi ngắn nhất từ s đến t .

Như vậy để tìm đường đi ngắn nhất từ đỉnh s đến đỉnh t , chúng ta cần phải tìm đường đi ngắn nhất từ s đến tất cả các đỉnh còn lại của G ³. Trên cơ sở này năm 1959 nhà toán học Hà Lan E. Dijkstra đã đề xuất một thuật toán tìm đường đi ngắn nhất từ đỉnh bắt đầu s đến các đỉnh còn lại của đồ thị dựa trên phương pháp tham lam. Thuật toán này thường được gọi là Thuật toán Dijkstra. Ý tưởng của thuật toán này được trình bày ngay sau. Để đơn giản ở đây chúng ta ký hiệu $d(s, v)$ lại bằng $d(v)$.

- Đầu tiên tính khoảng cách $d(v)$ cho mỗi đỉnh $v \in V$. Khoảng cách $d(v)$ lúc này cho biết cận trên của độ dài đường đi ngắn nhất từ s đến v , và gọi là *nhãn tạm thời* của v :

$$d(v) = \begin{cases} 0 & \text{nếu } v = s \\ c(s, v) & \text{nếu } (s, v) \in E \\ \infty & \text{nếu } (s, v) \notin E \end{cases}$$

Sau đó, xác định tập các đỉnh có nhãn tạm thời $T = V \setminus \{s\}$.

- Tiếp theo tại mỗi bước lặp, ta xác định *nhãn chính thức* và *sửa lại nhãn* đối với các nhãn tạm thời trong T nếu $T \neq \emptyset$:

(1) Xác định nhãn chính thức (hay còn gọi *cố định nhãn*):

- Tìm một đỉnh $u \in T$ sao cho $d(u) = \min\{d(z) | z \in T\}$.
- Cho u thành nhãn chính thức: $T = T \setminus \{u\}$.

³Lưu ý rằng hiện nay vẫn chưa thấy có thuật toán nào tìm đường đi ngắn nhất giữa hai đỉnh hiệu quả hơn thuật toán tìm đường đi ngắn nhất từ một đỉnh đến tất cả các đỉnh còn lại.

(2) Sửa lại nhãn: với mỗi $v \in T$, nếu $d(v) > d(u) + c(u, v)$ thì đặt $d(v) = d(u) + c(u, v)$.

Như vậy, có thể thấy các nhãn tạm thời sẽ biến đổi theo mỗi bước lặp sửa lại nhãn. Ở mỗi bước lặp sẽ có một nhãn tạm thời trở thành nhãn chính thức. Khi nhãn của mỗi đỉnh nào đó trở thành nhãn chính thức thì nó sẽ không phải cận trên nữa, mà chính là độ dài của đường đi ngắn nhất từ đỉnh s đến nó. Trong lược đồ tính toán sửa lại nhãn ở trên, chúng ta chưa đề cập đến thứ tự chọn các đỉnh u và v để kiểm tra. Có thể nhận thấy thứ tự chọn này có ảnh hưởng rất lớn đến hiệu quả của thuật toán.

Thuật toán 3.1 (*Thuật toán Dijkstra*)

Vào: Đồ thị có hướng có trọng số không âm $G = (V, E, c)$ và đỉnh xuất phát $s \in V$.
Ra:

- $d[v]$
- $Truoc[v]$: ghi nhận đỉnh đi trước v trong đường đi ngắn nhất từ s đến v .

Phương pháp:

Bước 1. (*Khởi tạo*) Gán nhãn tạm thời cho tất cả các đỉnh trong V :

```

for each ( $v \in V$ ) {
     $d(v) = \begin{cases} 0 & \text{nếu } v = s \\ c[s, v] & \text{nếu } (s, v) \in E; \\ \infty & \text{nếu } (s, v) \notin E \end{cases}$ 
     $Truoc[v] = s;$ 
}
 $T = V \setminus \{s\};$ 

```

Bước 2. (*Lặp*)

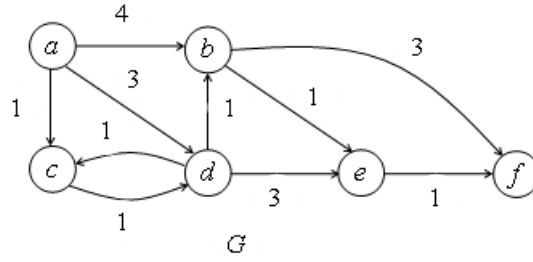
```

while ( $T \neq \emptyset$ ) {
    Tìm đỉnh  $u \in T$  sao cho  $d[u] = \min\{d[z] | z \in T\};$ 
     $T = T \setminus \{u\};$ 
    for each ( $v \in T$ )
        if ( $d[v] > d[u] + c[u, v]$ ) {
             $d[v] = d[u] + c[u, v];$ 
             $Truoc[v] = u;$ 
        }
}

```

Phép toán cơ bản của Thuật toán Dijkstra là phép so sánh. Gọi $T(n)$ là số phép so sánh cần thực hiện của Thuật toán Dijkstra. Có thể thấy Thuật toán Dijkstra có độ phức tạp là $O(n^2)$.

Ví dụ 3.19. Xét đồ thị có hướng, có trọng số G gồm 6 đỉnh như sau:



Hoặc tương đương ma trận có trọng số:

	a	b	c	d	e	f
a	0	4	1	3	∞	∞
b	∞	0	∞	∞	1	3
c	∞	∞	0	1	∞	∞
d	∞	1	1	0	3	∞
e	∞	∞	∞	∞	0	1
f	∞	∞	∞	∞	∞	0

Thuật toán Dijkstra tìm đường đi ngắn nhất từ đỉnh 1 đến các đỉnh còn lại qua bảng sau với quy ước: trong mỗi ô của bảng số đầu tiên là $d[v]$, số tiếp theo là $Truoc[v]$ và cuối cùng là dấu '*' chỉ đỉnh được chọn làm nhãn chính thức (tức là nhãn của nó không biến đổi ở các bước lặp tiếp theo).

	a	b	c	d	e	f
Bước KT	0, a	4, a	1, a^*	3, a	∞ , a	∞ , a
Bước 1	-	4, a	-	2, c^*	∞ , a	∞ , a
Bước 2	-	3, d^*	-	-	5, d	∞ , a
Bước 3	-	-	-	-	4, b^*	6, b
Bước 4	-	-	-	-	-	5, e^*

Lưu ý 3.1. (1) Nếu chỉ cần tìm đường đi ngắn nhất từ đỉnh s đến một đỉnh t nào đó thì trong Thuật toán Dijkstra chúng ta có thể kết thúc thuật toán ngay khi đỉnh t trở thành đỉnh có nhãn chính thức.

(2) Trường hợp nếu đồ thị G là vô hướng, có trọng số không âm thì bài toán tìm đường đi ngắn nhất trên G có thể dẫn về bài toán trên đồ thị có hướng bằng cách thay mỗi cạnh của G bởi hai cung có hướng ngược chiều nhau với cùng một trọng số là trọng số của cạnh tương ứng.

3.3. Một số thuật toán tìm kiếm trên đồ thị và ứng dụng

Rất nhiều thuật toán trên đồ thị được xây dựng dựa trên cơ sở duyệt qua tất cả các đỉnh của đồ thị sao cho mỗi đỉnh của nó được viếng thăm đúng một lần. Vì vậy việc xây dựng các thuật toán cho phép duyệt một cách có hệ thống tất cả các đỉnh của đồ thị là một vấn đề quan trọng. Những thuật toán như vậy thường được gọi là thuật toán duyệt (hay tìm kiếm) trên đồ thị. Các thuật toán này đóng một vai trò quan trọng trong việc thiết kế các thuật toán trên đồ thị. Mục này sẽ trình bày hai thuật toán như vậy: thuật toán tìm kiếm theo chiều sâu và thuật toán tìm kiếm theo chiều rộng.

Trong mục này chúng ta xét đồ thị vô hướng $G = (V, E)$ với n đỉnh và m cạnh.

3.3.1. Tìm kiếm theo chiều sâu

Tìm kiếm theo chiều sâu (Depth-first search - DFS) hay còn hiểu là tìm kiếm ưu tiên chiều sâu. Đây là một thuật toán tìm kiếm hay duyệt trên một đồ thị. Thuật toán khởi đầu tại một đỉnh nào đó (coi như đỉnh gốc) và phát triển xa nhất có thể theo mỗi nhánh.

Thuật toán 3.2 *Thuật toán DFS*

Vào: đồ thị vô hướng $G = (V, E)$ với n đỉnh và m cạnh.

Ra: dãy các đỉnh của G theo thứ tự được xét.

Phương pháp:

Bước 1. (*Khởi tạo*) Xuất phát từ một đỉnh tùy ý $s \in V$. Đánh dấu s đã xét và xem nó là đỉnh hiện hành v .

Bước 2. (*Lặp: tìm đỉnh kề*) Tìm đỉnh u bất kỳ kề với v . Xảy ra các trường hợp sau:

- Nếu đỉnh u chưa xét thì ta xét đỉnh này và xem nó là đỉnh hiện hành mới v . Lặp lại bước này.
 - Còn nếu không có đỉnh u nào kề với v là chưa xét thì ta nói đỉnh v đã duyệt xong, và quay trở lại đỉnh trước đó mà ta đến được đỉnh hiện hành v . Khi quay trở lại nếu $v = s$ thì kết thúc tìm kiếm, còn không lặp lại bước này.
-

Có thể thấy Thuật toán DFS phải xét qua tất cả các cạnh và các đỉnh của G (Bước 2), do đó độ phức tạp của Thuật toán DFS là $O(n + m)$.

Rõ ràng trong Thuật toán DFS các đỉnh được xét càng muộn thì càng sớm trở thành đỉnh đã duyệt xong. Điều này là hệ quả tất yếu của việc các đỉnh được xét sẽ được đưa vào trong ngăn xếp (Stack).

Thuật toán DFS trên có thể được cài đặt như sau:

```

{
1)  for each( $v \in V$ )
2)     $Chuaxet[v]=0$ ;
3)  for each( $v \in V$ )
5)    if( $Chuaxet[v]==0$ )
6)       $DFS(v)$ ;
}

```

Với thủ tục $DFS(v)$ là:

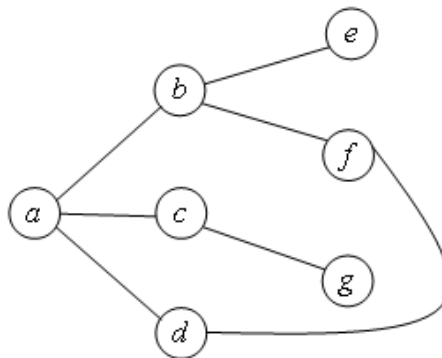
```

 $DFS(v)$ 
{
1)   $Thamdinh(v)$ ;
2)   $Chuaxet[v] = 1$ ;
3)  for each( $u \in Ke(v)$ )
4)    if( $Chuaxet[u] == 0$ )
5)       $DFS(u)$ ;
}

```

Trong đó mảng $Chuaxet[v]$ đánh dấu đỉnh v , bằng 0 nếu v chưa xét, ngược lại bằng 1.

Ví dụ 3.20. Xét đồ thị vô hướng G gồm 6 đỉnh như sau:



Với đỉnh xuất phát a , thì dãy các đỉnh được duyệt theo Thuật toán DFS là a, b, e, f, d, c, g .

Thuật toán DFS vẫn còn đúng cho trường hợp G là đồ thị có hướng, trong đó bước tìm đỉnh bất kỳ u kề với đỉnh hiện hành v được thay lại bằng tìm đỉnh bất kỳ u sao cho cung $(v, u) \in E$, tức là v là đỉnh đầu và u là đỉnh cuối của cung (v, u) trong G .

3.3.2. Tìm kiếm theo chiều rộng

Tìm kiếm theo chiều rộng (Breadth-first search - BFS) cũng là một thuật toán tìm kiếm hay duyệt trên đồ thị. Khác với tìm kiếm theo chiều sâu các đỉnh được xét vào trong ngăn xếp, ở đây tìm kiếm theo chiều rộng sẽ đưa các đỉnh được xét sẽ được đưa vào trong hàng đợi (Queue). Do đó các đỉnh được xét càng sớm thì càng sớm trở thành đỉnh đã duyệt xong. Khi đó một đỉnh sẽ trở thành đã duyệt xong ngay sau khi ta xét xong tất cả các đỉnh kề (chưa được thăm) với nó.

Thuật toán 3.3 *Thuật toán BFS*

Vào: đồ thị vô hướng $G = (V, E)$ với n đỉnh và m cạnh.

Ra: dãy các đỉnh của G theo thứ tự được xét.

Phương pháp:

Bước 1. (*Khởi tạo*) Xuất phát từ một đỉnh bất kỳ $v \in V$. Đưa v vào hàng đợi.

Bước 2. (*Lặp: tìm đỉnh kề*) Lần lượt lấy từng đỉnh ở đầu hàng đợi ra xét. Với mỗi đỉnh lấy ra xét, tìm các đỉnh kề với nó nhưng chưa xét, tất cả đưa vào hàng đợi. Lặp lại bước này cho đến khi hàng đợi rỗng. Thuật toán dừng, tất cả các đỉnh đều được xét.

Tương tự Thuật toán DFS, ở đây Thuật toán BFS cũng phải xét qua tất cả các cạnh và các đỉnh của G (Bước 2), do đó độ phức tạp của thuật toán là $O(n + m)$.

Thuật toán BFS cũng có thể được cài đặt như sau:

```
{  
1)  for each( $v \in V$ )  
2)    Chuaxet[v]=0;  
3)  for each( $v \in V$ )  
5)    if( $Chuaxet[v]==0$ )  
6)      BFS( $v$ );  
}
```

Với thủ tục $BFS(v)$ là:

```
BFS( $s$ )  
{  
1)  InitQueue( $Q$ );  
2)  AddQueue( $v, Q$ );  
3)  Chuaxet[v] = 1;  
4)  while( $Q \neq \emptyset$ ) {  
5)    RemoveQueue( $Q, w$ );  
6)    Thamdinhh( $w$ );
```

```

7)   for each( $u \in Ke(w)$ )
8)       if( $Chuaxet[u] == 0$ ) {
9)           AddQueue( $u, Q$ );
10)         $Chuaxet[u] = 1$ ;
11)    }
12) }
    }
```

Trong đó thủ tục $InitQueue(Q)$ khởi tạo hàng đợi Q (bằng $\emptyset$), thủ tục $AddQueue(v, Q)$ bổ sung đỉnh v vào hàng đợi Q và thủ tục $RemoveQueue(Q, w)$ lấy đỉnh w ra khỏi hàng đợi Q .

Ví dụ 3.21. Xét đồ thị như trong Ví dụ 3.20. Với đỉnh xuất phát a thì dãy các đỉnh được duyệt theo Thuật toán BFS là a, b, c, d, e, f, g .

3.3.3. Bài toán tìm đường đi

3.3.3.1. Phát biểu bài toán

Xét đồ thị vô hướng $G = (V, E)$ và hai đỉnh bất kỳ $s, t \in V$. Hãy tìm đường đi từ s đến t .

3.3.3.2. Thuật toán

Như đã trình bày ở trên, các thủ tục $DFS(s)$ và $BFS(s)$ sẽ cho phép thăm tất cả các đỉnh thuộc cùng một thành phần liên thông với đỉnh s . Vì vậy sau khi thực hiện xong thủ tục, nếu $Chuaxet[t] = 0$ thì điều đó có nghĩa là không có đường đi từ s đến t , còn nếu $Chuaxet[t] = 1$ thì t thuộc cùng thành phần liên thông với s , hay nói cách khác tồn tại một đường đi từ s đến t . Khi tồn tại đường đi, để ghi nhận đường đi chúng ta dùng thêm mảng $Truoc[v]$ để ghi nhận đỉnh đi trước đỉnh v trong đường đi từ s đến v . Khi đó các thủ tục trên cần sửa đổi lại một tí trong câu lệnh *if* như sau:

- Thủ tục $DFS(v)$:

```

if( $Chuaxet[u] == 0$ ) {
     $Truoc[u] = v$ ;
     $DFS(u)$ ;
}
```
- Thủ tục $BFS(v)$:

```

if(Chua_xet[u] == 0){
    AddQueue(u, Q);
    Chua_xet[u] = 1;
    Truoc[u] = w;
}

```

Như vậy có thể thấy trong đồ thị không có trọng số, đường đi tìm được theo thuật toán tìm kiếm theo chiều rộng chính là đường đi ngắn nhất (theo số cạnh) từ đỉnh bắt đầu s đến đỉnh đích t .

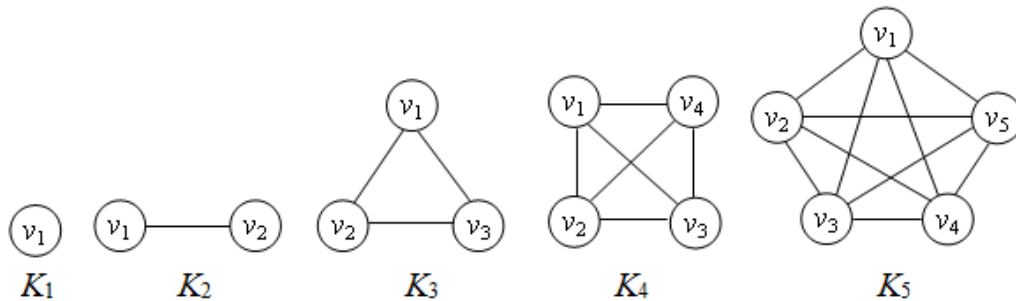
3.4. Một số loại đơn đồ thị vô hướng

Mục này giới thiệu một số dạng đơn đồ thị vô hướng đặc biệt thường xuất hiện nhiều trong nhiều vấn đề ứng dụng thực tế.

3.4.1. Đồ thị đầy đủ

Một đơn đồ thị vô hướng G được gọi là *đầy đủ* (Complete) nếu hai đỉnh bất kỳ của G đều liên kết với nhau. Đồ thị đầy đủ có n đỉnh thường được ký hiệu là K_n .

Ví dụ 3.22. Sau đây là các đơn đồ thị đầy đủ K_1, K_2, K_3, K_4 và K_5 :



Trong đời sống chúng ta hay gặp các đồ thị đầy đủ. Chẳng hạn đồ thị biểu diễn quan hệ quen biết của các giảng viên trong một Khoa của một trường Đại học là một đồ thị đầy đủ, vì đương nhiên các giảng viên trong một Khoa bao giờ cũng quen biết nhau hết. Từ định nghĩa, chúng ta dễ dàng suy ra ngay được tính chất cơ bản sau của đồ thị đầy đủ.

Mệnh đề 3.9. Đồ thị đầy đủ K_n có $\binom{n}{2}$ cạnh và mỗi đỉnh của K_n đều có bậc $n - 1$.

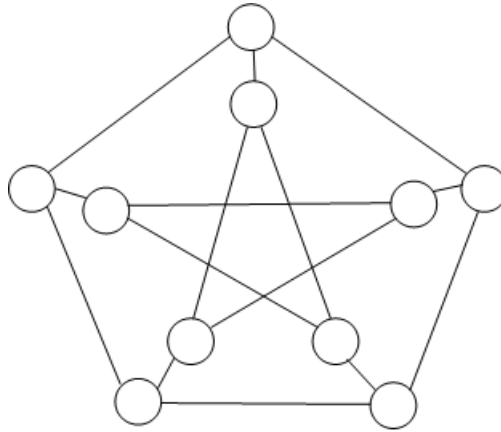
Như vậy, đồ thị đầy đủ K_n là đơn đồ thị có nhiều cạnh nhất trong số các đơn đồ thị có n đỉnh.

3.4.2. Đồ thị chính quy

Một đơn đồ thị vô hướng có bậc của mọi đỉnh đều bằng k được gọi là *đồ thị k -chính quy* (Regular) hay *đồ thị chính quy bậc k* .

Ví dụ 3.23. Một số đồ thị chính quy:

- Đồ thị đầy đủ K_n là đồ thị chính quy bậc $n - 1$.
- Đồ thị chỉ có các đỉnh cô lập là đồ thị 0-chính quy.
- Đồ thị Petersen là đồ thị 3-chính quy:



Hình 3.2: Đồ thị Petersen

Đồ thị chính quy có một số tính chất cơ bản sau.

Mệnh đề 3.10. *Đồ thị 2-chính quy bậc và liên thông có đúng một chu trình.*

Mệnh đề 3.11. *Số đỉnh của một đồ thị chính quy bậc lẻ luôn là một số chẵn.*

Chứng minh. Vì các đỉnh của một đồ thị chính quy bậc lẻ G là có bậc lẻ, nên theo Hệ quả 3.1 chúng ta suy ra ngay số đỉnh của G là một số chẵn. $\square$

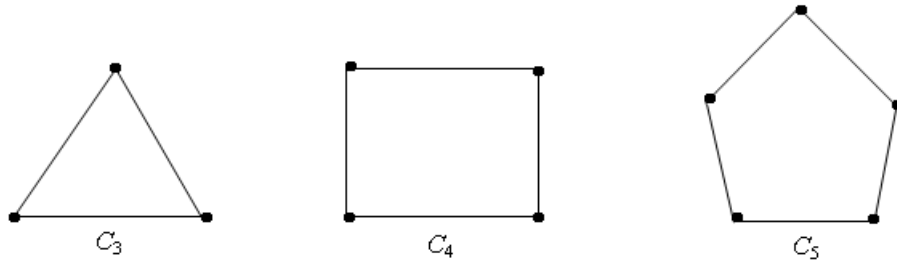
Mệnh đề 3.12. *Số cạnh của một đồ thị k -chính quy có n đỉnh là $\frac{nk}{2}$.*

Chứng minh. Gọi m là số cạnh của đồ thị chính quy bậc k có n đỉnh. Theo Mệnh đề 3.1 suy ra $nk = 2m$ hay $m = \frac{nk}{2}$. $\square$

3.4.3. Đồ thị vòng

Đơn đồ thị vô hướng có $n \geq 3$ đỉnh và một chu trình duy nhất qua tất cả các đỉnh được gọi là *đồ thị vòng* (*chu trình*, cycle graph), ký hiệu C_n . Như vậy, đồ thị C_n có n cạnh và mỗi đỉnh đều có bậc là 2, tức C_n là đồ thị đều bậc 2.

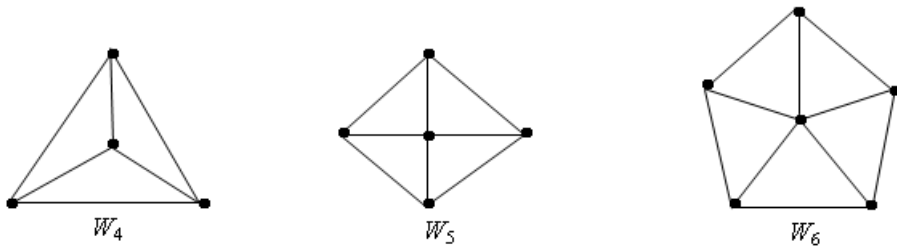
Ví dụ 3.24. Sau đây là các đồ thị vòng:



3.4.4. Đồ thị bánh xe

Đơn đồ thị vô hướng bao gồm một đồ thị vòng C_{n-1} và một đỉnh kề với tất cả các đỉnh của C_{n-1} được gọi là *đồ thị bánh xe* (wheel graph) và ký hiệu W_n . Như vậy, đồ thị bánh xe W_n có n đỉnh ($n \geq 4$), $2n - 2$ cạnh, một đỉnh bậc $n - 1$ và các đỉnh còn lại có bậc là 3.

Ví dụ 3.25. Sau đây là các đồ thị bánh xe:



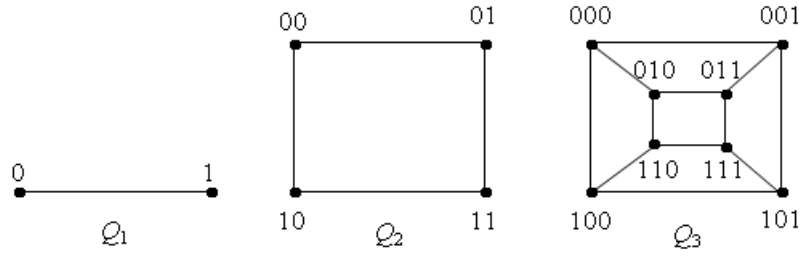
Có thể thấy $W_4 = K_4$.

Người ta chứng minh được số chu trình trong một đồ thị bánh xe W_n là $n^2 - 3n + 3$. Cụ thể, với $n = 4, 5, 6, 7, 8, 9, \dots$ tương ứng có 7, 13, 21, 31, 43, 57, $\dots$ chu trình.

3.4.5. Đồ thị lập phương

Đơn đồ thị vô hướng có 2^n đỉnh, tương ứng với 2^n xâu nhị phân độ dài n , và hai đỉnh kề nhau nếu hai xâu nhị phân tương ứng với hai đỉnh này chỉ khác nhau đúng một bit được gọi là *đồ thị lập phương* (cube graph), ký hiệu Q_n . Như vậy, mỗi đỉnh của đồ thị lập phương Q_n có bậc là n và số cạnh của Q_n là $n2^{n-1}$.

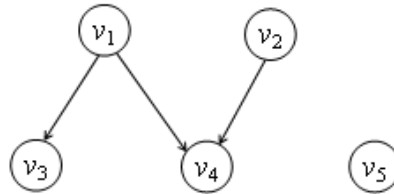
Ví dụ 3.26. Sau đây là một số đồ thị lập phương:



3.4.6. Đồ thị hai phía

Một đơn đồ thị vô hướng $G = (V, E)$ được gọi là *hai phía*⁴ (Bipartite) nếu tập đỉnh V có thể phân hoạch thành hai tập đỉnh X và Y sao cho mỗi cạnh của G chỉ nối một đỉnh nào đó trong X với một đỉnh nào đó trong Y . Đồ thị hai phía G thường được ký là $G = (X \cup Y, E)$.

Ví dụ 3.27. Đồ thị gồm 5 đỉnh sau là đồ thị hai phía:

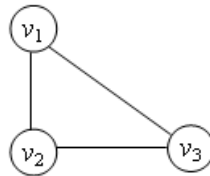


Với $X = \{v_1, v_2\}$ và $Y = \{v_3, v_4, v_5\}$.

Sau đây là một tiêu chuẩn nhận biết một đơn đồ thị vô hướng có phải là đồ thị hai phía hay không.

Mệnh đề 3.13. Một đơn đồ thị vô hướng là đồ thị hai phía khi và chỉ khi nó không chứa chu trình độ dài lẻ.

Ví dụ 3.28. Xét đơn đồ thị G sau:



Đồ thị G không phải là đồ thị hai phía, vì nó chứa chu trình độ dài 3 là v_1, v_2, v_3, v_1 .

Đối với đơn đồ thị vô hướng liên thông chúng ta có thể dùng thuật toán đơn giản sau để xác định xem nó có phải là một đồ thị hai phía hay không.

⁴Còn được gọi là đồ thị phân đôi hay đồ thị lưỡng phân.

Thuật toán 3.4 (*Nhận biết đồ thị hai phía*)

Vào: Đơn đồ thị vô hướng liên thông $G = (V, E)$.

Ra: Cho biết G có phải đồ thị hai phía hay không.

Phương pháp:

Bước 1. (*Khởi tạo*) Chọn một đỉnh bất kỳ v trong G và đặt $X = \{v\}$.

Bước 2. (*Lặp*)

do{

 Tính Y là tập tất cả các đỉnh kề của các đỉnh trong X ;

 Tính Z là tập tất cả các đỉnh kề của các đỉnh trong Y ;

if ($Z \cap Y \neq \emptyset$)

 Thuật toán dừng, kết luận G không phải đồ thị hai phía;

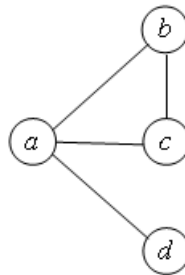
else

$X = X \cup Z$;

while (X còn thay đổi)

 Kết luận G là đồ thị hai phía với phân hoạch của V là $(X, Y = V \setminus X)$.

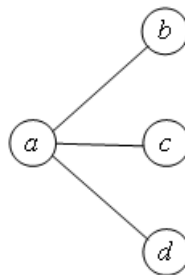
Ví dụ 3.29. Xét đồ thị vô hướng liên thông G sau:



Xuất phát $X = \{b\}$.

Tính $Y = \{a, c\}$ và $Z = \{a, b, c, d\}$. Vì $Z \cap Y \neq \emptyset$, suy ra G không phải đồ thị hai phía.

Ví dụ 3.30. Xét đồ thị vô hướng liên thông G như sau:



Chọn $X = \{b\}$.

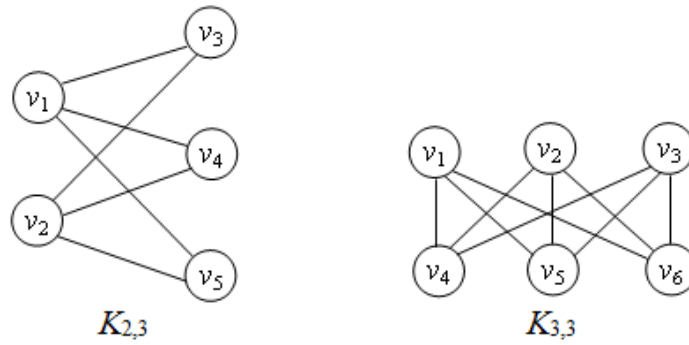
Tính $Y = \{a\}$ và $Z = \{b, c, d\}$.

Tính $X = X \cup Z = \{b, c, d\}$. Suy ra $Y = \{a\}$, $Z = \{b, c, d\}$, và do đó $X = \{b, c, d\}$. Vì X không thay đổi nên G là đồ thị hai phía, với phân hoạch $X = \{b, c, d\}$ và $Y = \{a\}$.

Đồ thị hai phía $G = (X \cup Y, E)$ với $|X| = m$ và $|Y| = n$ được gọi là *đồ thị hai phía đầy đủ* (Complete bipartite), ký hiệu $K_{m,n}$ hay $K_{m \times n}$, nếu mỗi đỉnh trong X kề với mỗi đỉnh trong Y .

Như vậy, đồ thị hai phía đầy đủ $K_{m,n}$ có $m \times n$ cạnh và các đỉnh của X có bậc n , các đỉnh của Y có bậc m .

Ví dụ 3.31. Sau đây là các đồ thị hai phía đầy đủ:



3.5. Hàm Grundy

Hàm Grundy là một hàm toán học xây dựng trên đồ thị, do P. M. Grundy đề xuất thường dùng để trong lý thuyết trò chơi nhằm giúp tìm ra chiến lược giành phần thắng trong các trò chơi đối kháng với hai đối thủ biểu diễn được bằng đồ thị.

3.5.1. Định nghĩa

Cho đồ thị $G = (V, E)$. Ánh xạ $g : V \rightarrow \mathbb{N}$ được gọi là *hàm Grundy* nếu

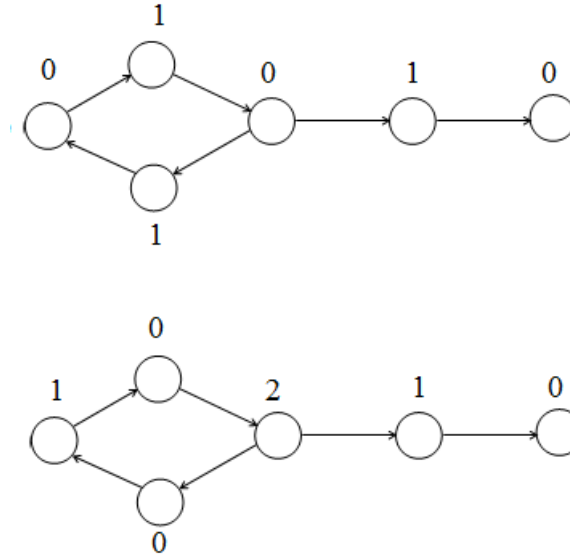
$$\forall v \in V : g(v) = \min\{\mathbb{N} \setminus g(F(v))\}$$

ở đây $g(F(v)) = \{g(u) : u \in F(v)\}$. Từ định nghĩa, chúng ta dễ dàng suy ra hai tính chất đặc trưng của hàm Grundy như sau:

1. $\forall u, v \in V$, nếu $v \in F(u)$ thì $g(u) \neq g(v)$.
2. $\forall x \in \mathbb{N}$, nếu $x < g(v)$ thì $x \in g(F(v))$, tức là $\exists u \in F(v)$ để $g(u) = x$.

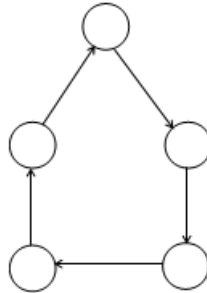
Tính chất 1) chỉ ra rằng $g(v)$ không nằm trong tập giá trị của g trên $F(v)$.
 Tính chất 2) khẳng định $g(v)$ là số nguyên không âm bé nhất trong số các số không thuộc tập giá trị của hàm g trên $F(v)$.

Ví dụ 3.32. Đồ thị sau có hai hàm Grundy:



Như vậy, hàm Grundy có thể không duy nhất. Ngoài ra, hàm Grundy cũng có thể không tồn tại.

Ví dụ 3.33. Đồ thị sau không có hàm Grundy:



Các tính chất cơ bản sau của hàm Grundy là dễ dàng suy ra từ định nghĩa.

Mệnh đề 3.14.

- 1) Nếu $F(v) = \emptyset$ thì $g(v) = 0$.
- 2) Tập hợp $\{v | v \in V \wedge g(v) = 0\}$ luôn khác rỗng.
- 3) $g(v) \leq |F(v)|, \forall v \in V$.
- 4) Đồ thị có khuyên hoặc có chu trình độ dài lẻ thì không có hàm Grundy.

Sau đây là một điều kiện đủ nhận biết khi nào một đồ thị có một hàm Grundy duy nhất.

Định lý 3.15. Mọi đồ thị không có chu trình luôn có duy nhất một hàm Grundy.

Chứng minh. Không mất tính tổng quát giả thiết đồ thị G là liên thông. Xây dựng hai dãy tập con các đỉnh $V_0, V_1, \dots$ và $P_0, P_1, \dots$ lần lượt như sau:

$$V_0 = V,$$

$$P_0 = \{v \mid F(v) = \emptyset\}.$$

Rõ ràng tập $P_0 \neq \emptyset$, vì nếu không mọi đỉnh trong G luôn có đỉnh kề. Điều này là vô lý với giả thiết G không có chu trình. Sau đó tiếp tục xây dựng

$$V_1 = V_0 \setminus P_0,$$

$$P_1 = \{v \mid v \in V_1, F(v) \subseteq V \setminus V_1\},$$

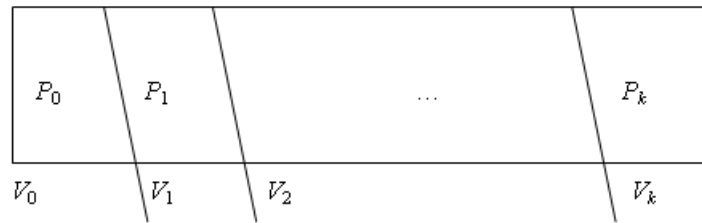
$$V_2 = V_1 \setminus P_1,$$

$\dots$

$$P_i = \{v \mid v \in V_i, F(v) \subseteq V \setminus V_i\}, \text{ với } i \geq 2,$$

$$V_{i+1} = V_i \setminus P_i.$$

Lưu ý rằng, nếu $P_k = \emptyset$ thì $V_k = \emptyset$, nghĩa là đã vét hết các đỉnh của G . Bởi nếu không với mỗi $v \in V_k$ sẽ có $u \in F(v)$ sao cho $u \notin V \setminus V_k$, tức là $u \in V_k$. Điều này có nghĩa, với mọi đỉnh trong V_k luôn có đỉnh kề cũng thuộc V_k , suy ra xuất phát từ một đỉnh trong V_k có thể tạo thành một đường đi dài tùy ý. Điều này vô lý với G không có chu trình.



Hình 3.3: Cách xây dựng dãy tập con $P_0, P_1, \dots, P_k$

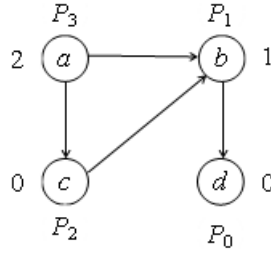
Bây giờ, chúng ta xây dựng hàm $g : V \rightarrow \mathbb{N}$ như sau:

- (1) Với mỗi $v \in P_0$, đặt $g(v) = 0$.
- (2) Với mỗi $v \in P_1$, đặt $g(v) = \min\{\mathbb{N} \setminus g(F(v))\}$. Hàm g trên P_1 xác định duy nhất bởi $F(v) \subseteq P_0$ và g đã được xác định trên P_0 .

Tiếp tục cách làm như trên chúng ta sẽ lần lượt xác định được giá trị hàm g tại các đỉnh của G một cách duy nhất. $\square$

Chúng minh của Định lý 3.15 cũng cho chúng ta thuật toán tìm hàm Grundy của đồ thị phi chu trình.

Ví dụ 3.34. Sau đây là đồ thị có hướng và cách xây dựng hàm Grundy trên nó:

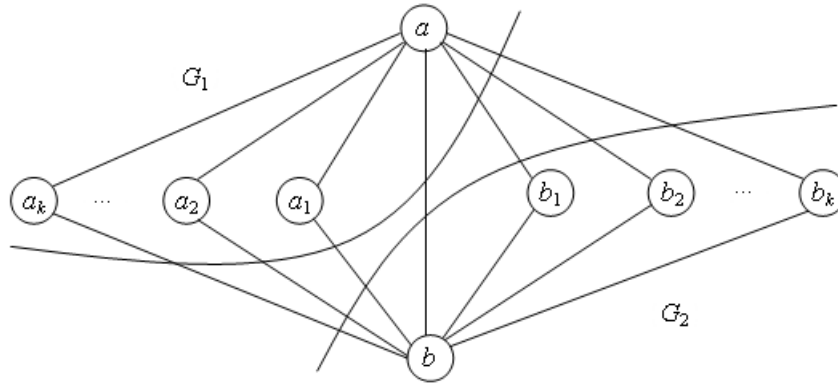


3.5.2. Đồ thị tổng và hàm Grundy

Xét các đồ thị sau dưới dạng ánh xạ kề $G_1 = (V_1, F_1)$ và $G_2 = (V_2, F_2)$. Đồ thị $G = (V, F)$ được gọi là *đồ thị tổng* của G_1 và G_2 , ký hiệu $G_1 + G_2$, nếu

(i) $V = V_1 \times V_2$.

(ii) $(u, v) \in F((a, b)) \Leftrightarrow u = a \wedge v \in F(b)$ hoặc $u \in F(a) \wedge v = b$.



Hình 3.4: Cách xây dựng đồ thị tổng

Đồ thị tổng $G_1 + G_2$ còn được gọi là *đồ thị hợp* và ký hiệu lại là $G_1 \cup G_2$. Phép toán tổng này xuất hiện nhiều trong các lĩnh vực có sự tham gia của lý thuyết đồ thị. Đây là phép toán rất quen thuộc trong lý thuyết trò chơi.

Ví dụ 3.35. Giả sử có hai máy M_1 và M_2 . Mỗi máy có thể ở trong một số trạng thái. Ký hiệu tập trạng thái của máy M_i là V_i và

$$E_i = \{(u_i, v_i) | v_i \text{ là trạng thái tiếp theo của trạng thái } u_i\}$$

($i = 1, 2$). Khi đó, nếu người thợ tại mỗi thời điểm chỉ tác động một trong hai máy thì đồ thị mô tả các tình huống có thể sẽ là $G_1 + G_2$.

Vấn đề đặt ra phép toán tổng trên đồ thị có bảo toàn tính chấp nhận hàm Grundy hay không? Để trả lời câu hỏi này, chúng ta bắt đầu với giới thiệu phép toán d -tổng hay còn gọi tổng digit trên các số nguyên.

Cho các số nguyên không âm $x, y \in \mathbb{N}$ biểu diễn dưới dạng nhị phân như sau:

$$x = x_k x_{k-1} \dots x_1 x_0$$

$$y = y_l y_{l-1} \dots y_1 y_0$$

với x_i, y_i là các chữ số 0 hoặc 1. Giả thiết độ dài của hai số này là bằng nhau, trường hợp nếu không thì thêm các số 0 vô nghĩa vào bên trái của số ngắn hơn. Đặt $z_i = (x_i + y_i) \bmod 2$. Khi đó, số nguyên z có biểu diễn nhị phân là $z_k z_{k-1} \dots z_1 z_0$ được gọi là *tổng digit* của x và y , ký hiệu $z = x \oplus y$.

Ví dụ 3.36. a) $4 \oplus 5 = 1$.

b) $3 \oplus 7 = 4$.

Tổng digit có tính chất cơ bản sau. Kết quả này là dễ dàng suy ra.

Mệnh đề 3.16. $(\mathbb{N}, \oplus)$ là một nhóm giao hoán, tức là:

(1) *Tổng digit có tính kết hợp:* $(x \oplus y) \oplus z = x \oplus (y \oplus z)$

(2) *Tổng digit có phần tử đơn vị 0:* $x \oplus 0 = 0 \oplus x = x$

(3) *Với mọi số x tồn tại phần tử đối $-x$:* $x \oplus -x = 0$

(4) *Tổng digit có tính giao hoán:* $x \oplus y = y \oplus x$.

Sau đây là quy tắc tính hàm Grundy cho tổng các đồ thị.

Định lý 3.17. Nếu các đồ thị G_1 và G_2 chấp nhận các hàm Grundy tương ứng g_1 và g_2 , thì đồ thị tổng $G_1 + G_2$ cũng chấp nhận hàm Grundy $g(u, v) = g_1(u) \oplus g_2(v)$.

Chứng minh. Theo định nghĩa hàm Grundy, đầu tiên chúng ta chứng minh không có đỉnh (u, v) nào kề với đỉnh (a, b) mà $g((u, v)) = g((a, b))$. Thật vậy, lấy bất kỳ đỉnh $(u, v) \in F((a, b))$. Trường hợp nếu $u = a$ và $v \in F_2(b)$ thì $g_2(v) \neq g_2(b)$ và khi đó $g((a, b)) = g_1(a) \oplus g_2(b) = g_1(u) \oplus g_2(b) \neq g_1(u) \oplus g_2(v) = g((u, v))$. Lập luận tương tự, trường hợp nếu $u \in F_1(a)$ và $v = b$ thì ta cũng có $g((a, b)) \neq g((u, v))$.

Bây giờ chỉ cần chứng minh với mọi $x \in \mathbb{N}$, nếu $x < g((a, b))$ thì tồn tại $(u, v) \in F((a, b))$ sao cho $g((u, v)) = x$. Thật vậy nếu ký hiệu $y = g_1(a)$ và $z = g_2(b)$ thì

$x < y \oplus z$. Đặt $t = x \oplus y \oplus z$. Rõ ràng $t \neq 0$ vì $x \neq y \oplus z$. Hơn nữa $t \oplus x = x \oplus y \oplus z \oplus x = y \oplus z > x$. Xét biểu diễn nhị phân của các số x, y, z và t . Giả sử k là chỉ số của bit đầu tiên bằng 1 trong biểu diễn nhị phân của t , tức $t_k = 1$. Rõ ràng nếu $u_k = 1$ thì $(u_k + t_k) \bmod 2 = 0$, và do đó $t \oplus x < x$. Điều này là mâu thuẫn. Như vậy $u_k = 0$. Do $t_k = 1$ nên hoặc $v_k = 1$ hoặc $w_k = 1$. Giả sử $v_k = 1$. Khi đó đặt $s = t \oplus y$. Ta có $s < y = g_1(a)$. Vì g_1 là hàm Grundy của G_1 nên tồn tại $x \in F_1(a)$ sao cho $s = g_1(x)$. Theo định nghĩa đồ thị tổng suy ra $(x, b) \in F((a, b))$ và $g((x, b)) = g_1(x) \oplus g_2(b) = s \oplus z = t \oplus y \oplus z = x$. Trường hợp $w_k = 1$ lập luận hoàn toàn tương tự. $\square$

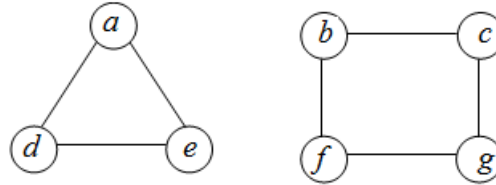
Định lý 3.17 có thể mở rộng cho tổng của nhiều đồ thị.

3.6. Chu số của đồ thị

3.6.1. Định nghĩa

Cho đa đồ thị vô hướng $G = (V, E)$ có n đỉnh, m cạnh và $\omega(G) = p$ thành phần liên thông. Đại lượng $v(G) = m - n + p$ được gọi là *chu số* (cyclomatic number) của G . Khái niệm chu số đề cập ở đây không phụ thuộc vào sự định hướng của đồ thị.

Ví dụ 3.37. Xét đồ thị vô hướng và không liên thông G sau:



G có $n = 7, m = 7$ và $p = 2$. Suy ra chu số $v(G) = 2$.

3.6.2. Tính chất

Định lý 3.18. Nếu thêm một cạnh mới vào đa đồ thị vô hướng G thì chu số tăng lên 1 hoặc không thay đổi.

Chứng minh. Giả sử (a, b) là một cạnh mới thêm vào G . Khi đó m tăng lên 1. Xảy ra hai trường hợp, trường hợp nếu hai đỉnh a và b thuộc cùng một mảng liên thông trong G thì n và p không thay đổi, do đó chu số tăng lên 1. Trường hợp nếu hai đỉnh a và b nằm ở hai mảng liên thông khác nhau trong G thì p giảm xuống 1 (n vẫn thay đổi) và do vậy chu số không thay đổi. $\square$

Hệ quả 3.4. Chu số của đa đồ thị vô hướng là một số nguyên không âm.

Chứng minh. Giả sử đa đồ thị vô hướng G được xây dựng từ đồ thị G_0 gồm n đỉnh và không có cạnh nào. Sau đó, lần lượt thêm các cạnh vào G_0 để được đồ thị G . Vì chu số của G_0 là $v(G_0) = 0 - n + n = 0$ và quá trình thêm cạnh không làm giảm chu số, nên chu số của G lớn hơn hoặc bằng chu số của G_0 . $\square$

3.6.3. Chu trình độc lập

Những kết quả của đại số vectơ có thể vận dụng vào ở đây bằng cách đặt mỗi chu trình trong G bằng một vectơ theo cách như sau:

Mỗi cạnh của G đều được định hướng một cách tùy ý, nếu chu trình μ đi qua cạnh e_k là r_k lần thuận hướng và s_k lần ngược hướng thì ta đặt $c_k = r_k - s_k$ (trường hợp nếu e_k là một khuyên thì ta luôn qui ước $s_k = 0$). Khi đó vectơ m chiều $(c_1, c_2, \dots, c_m)$ được gọi là *vectơ chu trình* tương ứng với μ , ký hiệu $\vec{\mu}$. Các chu trình $\mu_1, \mu_2, \dots$ được gọi là *độc lập* nếu các vectơ chu trình tương ứng là độc lập tuyến tính. Lưu ý rằng định nghĩa này không phụ thuộc vào hướng gán cho các cạnh.

Định lý 3.19. Cho đa đồ thị $G = (V, E)$. Khi đó chu số $v(G)$ bằng số cực đại các chu trình độc lập.

Chứng minh. Tiến hành tương tự như trong Hệ quả 3.4, đầu tiên chúng ta xuất phát từ đồ thị vô hướng không có cạnh với tập đỉnh là V . Sau đó, xây dựng xây dựng đa đồ thị G' bằng cách thêm từng cạnh một vào. Theo Định lý 3.18, chu số sẽ tăng lên một đơn vị nếu cạnh thêm vào lập nên một chu trình mới hoặc chu số không thay đổi trong trường hợp ngược lại. $\square$

Hệ quả sau là dễ dàng suy ra.

Hệ quả 3.5. 1) Đa đồ thị G không có chu trình khi và chỉ khi $v(G) = 0$.
2) Đa đồ thị G có một chu trình duy nhất khi và chỉ khi $v(G) = 1$.

3.7. Các tập hợp đặc biệt trên đồ thị

Mục này tìm hiểu một số tập hợp đặc biệt gồm các đỉnh và các cạnh trên một đồ thị, đó là các tập ổn định trong, tập ổn định ngoài, nhân, clique, phủ đỉnh và phủ cạnh của đồ thị. Các tập hợp này có nhiều ứng dụng quan trọng.

3.7.1. Tập ổn định trong

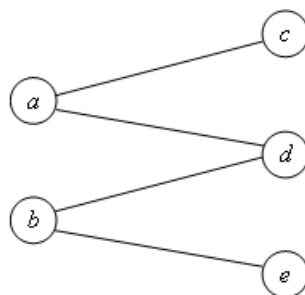
Cho đồ thị $G = (V, E)$. *Tập ổn định trong*⁵ (Internally stable set) của G là một tập con các đỉnh của G sao cho không có hai đỉnh nào thuộc nó là kề nhau.

Như vậy, theo quan điểm ánh xạ kề, tập $I \subseteq V$ là một ổn định trong của G khi và chỉ khi

$$\forall v \in I : I \cap F(v) = \emptyset.$$

Một tập con các đỉnh của G được gọi là *tập ổn định trong cực đại* (hay *tập ổn định trong tối đại*-Maximal internally stable set) nếu thêm vào bất kỳ một đỉnh nào cũng làm mất tính ổn định trong của nó, tức là không tồn tại một tập ổn định trong nào khác chứa nó. Tập ổn định trong có nhiều phần tử nhất được gọi là *tập ổn định trong lớn nhất* (Maximum internally stable set). Rõ ràng, $\emptyset$ là một tập ổn định trong, các tập con của một tập ổn định trong cũng là một tập ổn định trong. Tập ổn định trong lớn nhất là tập ổn định trong cực đại, ngược lại thì chưa chắc. Khái niệm ổn định trong không phụ thuộc vào hướng các cạnh của đồ thị.

Ví dụ 3.38. Xét đồ thị vô hướng G như sau:



Ta có $\{c, d\}, \{d, e\}$ là các tập ổn định trong, nhưng không phải tập ổn định trong cực đại của G . Tập $\{a, c, e\}$ không phải tập ổn định trong của G . Các tập $\{a, b\}$ và $\{c, d, e\}$ là các tập ổn định trong cực đại của G . Tập $\{c, d, e\}$ là tập ổn định trong lớn nhất của G .

Ví dụ 3.38 cũng cho thấy một đồ thị có thể có nhiều tập ổn định trong cực đại và số đỉnh trong chúng không nhất thiết phải bằng nhau.

Ví dụ 3.39. (Gauss) Xét bài toán tám con hậu. Khi đó các ô cần tìm để đặt 8 con hậu sao cho chúng không ăn được lẫn nhau chính là một tập ổn định trong gồm 8 đỉnh của đồ thị vô hướng biểu diễn bàn cờ vua trên. Bài toán có tất cả 92 nghiệm, được suy ra từ 12 tập ổn định trong khác nhau.

⁵Cũng còn được gọi là *tập ổn định* hay *tập độc lập*-Independent set.

Ví dụ 3.40. Trong một đơn vị nào đó, giả sử có quan hệ “xích mích” giữa người với người. Khi đó một nhóm nhiều người nhất không xích mích với nhau chính là một tập ổn định trong cực đại của đồ thị biểu diễn quan hệ xích mích ở trên. Trong trường hợp này tập ổn định trong cực đại được hiểu theo đúng nghĩa xã hội của nó. Do đó để giữ được đoàn kết trong đơn vị, thì cần phải xây dựng nhóm này càng lớn càng tốt.

Số đỉnh của tập ổn định trong lớn nhất của đồ thị G được gọi là *số ổn định trong* của G , ký hiệu $\alpha(G)$.

Như vậy, nếu ký hiệu $\mathcal{I}(G)$ là *tập tất cả tập ổn định trong* của đồ thị G , thì

$$\alpha(G) = \max\{|I| \mid I \in \mathcal{I}(G)\}.$$

Xét Ví dụ 3.38 ở trên ta có $\alpha(G) = 3$.

Lưu ý 3.2. Đối với tập ổn định trong, người ta thường quan tâm đến các tập ổn định trong lớn nhất. Bài toán tìm tập ổn định trong lớn nhất (còn được gọi là *bài toán tập ổn định trong lớn nhất* hay *bài toán tập độc lập lớn nhất*) là một bài toán tối ưu NP-khó. Tuy nhiên, bài toán tìm một tập ổn định trong cực đại có thể giải quyết trong thời gian đa thức bằng một thuật toán tham lam tầm thường. Tìm tất cả các tập ổn định trong cực đại có thể giải quyết bằng một thuật toán có độ phức tạp thời gian hàm mũ là $O(3^{n/3})$. *Bài toán quyết định tập ổn định trong* (có hay không một tập ổn định trong có kích thước bằng $k \in \mathbb{Z}^+$ cho trước) là một bài toán NP-đầy đủ.

Kết quả sau đưa ra chặn dưới cho số ổn định trong của một đồ thị.

Định lý 3.20. Cho đồ thị $G = (V, E)$ có n đỉnh sao cho bậc lớn nhất của các đỉnh là ρ . Số ổn định trong của G thỏa

$$\alpha(G) \geq \frac{n}{\rho + 1}.$$

Chứng minh. Giả sử I là tập ổn định trong lớn nhất với m đỉnh của G . Khi đó với mỗi đỉnh $u \notin I$, có ít nhất một đỉnh của I kề với u . Từ đây, suy ra số cạnh đi ra khỏi I (không kể hướng) không nhỏ hơn $|V \setminus I| = n - m$. Mặt khác, số cạnh đó không lớn hơn ρm . Tóm lại, ta có $n - m \leq \rho m$, hay

$$m \geq \frac{n}{\rho + 1}.$$

□

Sau đây, là thuật toán tìm một tập ổn định trong lớn nhất. Như đã đề cập ở trên thuật toán này là rất khó. Tuy nhiên, tìm một tập ổn định trong cực đại cũng thu được trong thuật toán này chỉ có độ phức tạp đa thức.

Thuật toán 3.5 (*Tìm một tập ổn định trong lớn nhất*)

Vào: Đồ thị $G = (V, E)$.

Ra: Tập ổn định trong lớn nhất I của G .

Phương pháp:

Bước 1. Chọn một đỉnh bất kỳ của G .

Bước 2. Bổ sung dần các đỉnh để thu được một tập ổn định trong cực đại.

Bước 3. Nếu tìm được một tập ổn định trong có m đỉnh và mọi tập có $m + 1$ đỉnh đều không phải là tập ổn định trong, thì tập ổn định trong có m đỉnh này chính là tập ổn định trong lớn nhất của G . Lúc này $\alpha(G) = m$.

Ví dụ 3.41. Xét đồ thị G trong Ví dụ 3.38.

Xuất phát từ đỉnh bất kỳ c . Bổ sung dần các đỉnh vào $\{c\}$ thỏa điều kiện của tập ổn định trong, ta thu được tập ổn định trong cực đại $\{c, d, e\}$. Kiểm tra thấy các tập có 4 đỉnh trở lên không phải là tập ổn định trong, do đó tập $\{c, d, e\}$ cũng chính là tập ổn định trong lớn nhất của G .

3.7.2. Tập ổn định ngoài

Cho đồ thị $G = (V, E)$. *Tập ổn định ngoài*⁶ (Externally stable set) của G là một tập con các đỉnh của G sao cho mọi đỉnh không thuộc nó đều kề với ít nhất một đỉnh trong nó hoặc có cung đi vào ít nhất một đỉnh trong nó (nếu G là đồ thị có hướng).

Nói một cách khác, tập $E \subseteq V$ là một ổn định ngoài của G khi và chỉ khi

$$\forall v \notin E : E \cap F(v) \neq \emptyset$$

hay nói cách khác

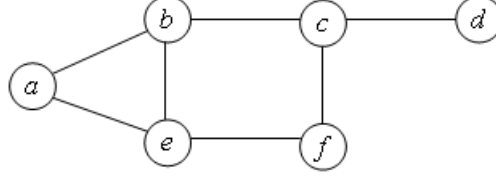
$$\forall v \notin E, \exists u \in E : u \in F(v).$$

Một tập con các đỉnh của G được gọi là *tập ổn định ngoài cực tiểu* (Minimal externally stable set) nếu bớt đi bất kỳ một đỉnh nào thuộc nó cũng làm mất tính ổn định ngoài của nó, tức là nó không chứa một tập con nào là tập ổn định ngoài. Tập ổn định ngoài có ít phần tử nhất được gọi là *tập ổn định ngoài nhỏ nhất*.

⁶Còn được gọi là *tập thống trị*-Dominating set.

(Minimum externally stable set). Rõ ràng, V là một tập ổn định ngoài, và các tập bao hàm một tập ổn định ngoài cũng là một tập ổn định ngoài. Một đồ thị có thể có nhiều tập ổn định ngoài cực tiểu và số phần tử của chúng không nhất thiết là bằng nhau.

Ví dụ 3.42. Xét đồ thị G như sau:



Ta có $\{a, d, f\}$, $\{c, e\}$, $\{b, c\}$, $\{b, c, d\}$ là các tập ổn định ngoài của G . Tập $\{a, b, e\}$ không phải tập ổn định ngoài của G . Tập $\{a, d, f\}$, $\{c, e\}$, $\{b, c\}$ là các tập ổn định ngoài cực tiểu của G . Các tập $\{a, c\}$, $\{b, c\}$, $\{c, e\}$ là các tập ổn định ngoài nhỏ nhất của G .

Ví dụ 3.43. Bài toán thực tế dạng này (phân bổ các tâm phục vụ) có rất nhiều, có thể kể ra một vài bài toán cơ bản như:

- (1) Phân bổ các trung tâm phát thanh, truyền hình trên một lãnh thổ nào đó.
- (2) Phân bổ các căn cứ quân sự kiểm soát một vùng lãnh thổ cho trước.
- (3) Phân bổ các trung tâm bán hàng phục vụ một vùng dân cư.

Lực lượng của tập ổn định ngoài nhỏ nhất của G được gọi là *số ổn định ngoài* của đồ thị G , ký hiệu $\beta(G)$.

Như vậy nếu ký hiệu $\mathcal{E}(G)$ là tập tất cả các tập ổn định ngoài của đồ thị G , thì

$$\beta(G) = \min\{|E| \mid E \in \mathcal{E}(G)\}.$$

Xét Ví dụ 3.42 ở trên ta có $\beta(G) = 2$.

Lưu ý 3.3. Đối với tập ổn định ngoài, người ta thường quan tâm đến các tập ổn định ngoài nhỏ nhất. *Bài toán quyết định tập ổn định ngoài* (có hay không một đồ thị G có $\beta(G) \leq k$ với $k \in \mathbb{Z}^+$ cho trước) là bài toán NP-đầy đủ. Thuật toán tốt nhất hiện nay tìm một tập ổn định ngoài nhỏ nhất (còn được gọi là *bài toán tập ổn định ngoài nhỏ nhất* hay *bài toán tập thống trị nhỏ nhất*) có độ phức tạp thời gian hàm mũ $O(1.4597^n)$. Tuy nhiên, bài toán tìm một tập ổn định ngoài cực tiểu có thể giải quyết trong thời gian đa thức.

Nhận xét 3.1. Từ định nghĩa chúng ta dễ dàng suy ra được một số tính chất cơ bản sau:

- (1) Tập gồm đúng một đỉnh trong đồ thị đầy đủ là tập ổn định ngoài cực tiểu.
- (2) Mọi tập ổn định ngoài chứa ít nhất một tập ổn định ngoài cực tiểu.
- (3) Tập ổn định ngoài cực tiểu có thể hoặc không là tập ổn định trong.
- (4) Một tập ổn định trong là tập ổn định ngoài nếu và chỉ nếu nó là tập ổn định trong cực đại.
- (5) Với mọi đồ thị G bất đẳng thức sau luôn đúng:

$$\beta(G) \leq \alpha(G).$$

Sau đây là thuật toán tìm một tập ổn định ngoài nhỏ nhất của một đồ thị.

Thuật toán 3.6 (*Tìm một tập ổn định ngoài nhỏ nhất*)

Vào: Đồ thị $G = (V, E)$.

Ra: Tập ổn định ngoài nhỏ nhất E của G .

Phương pháp:

Bước 1. Xây dựng ánh xạ $T : V \rightarrow 2^V$ như sau:

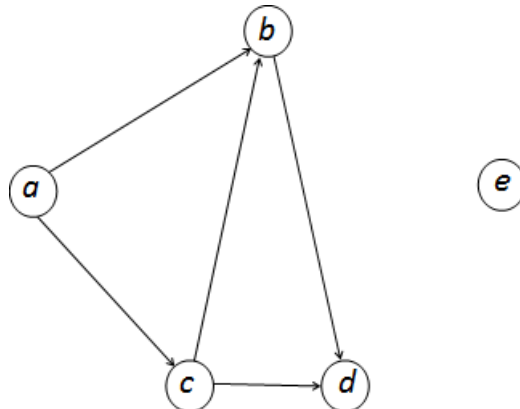
$$T(u) = \{u\} \cup F^{-1}(u), \forall u \in V.$$

Bước 2. Tìm tập con $E \subseteq V$ có số phần tử ít nhất sao cho $\bigcup_{v \in E} T(v) = V$. Tập con E chính là tập ổn định ngoài nhỏ nhất của G cần tìm.

Trong thuật toán trên, ở Bước 2 chúng ta có thể thực hiện nhanh hơn nhờ các nhận xét sau:

- (1) Đỉnh cô lập hoặc đỉnh không có cung đi ra luôn thuộc tập ổn định ngoài nhỏ nhất, nghĩa là các đỉnh này phải được giữ lại.
- (2) Nếu có $T(a) \subseteq T(b)$ thì bỏ không xét đỉnh a .
- (3) Thực hiện việc loại bỏ cho đến khi chỉ còn các đỉnh không thể loại bỏ được nữa. Tập đỉnh sau cùng này chính là tập ổn định ngoài nhỏ nhất cần tìm của G .

Ví dụ 3.44. Xét đồ thị có hướng $G = (V, E)$ như sau:



Ta có

$$T(a) = \{a\}$$

$$T(b) = \{a, b, c\}$$

$$T(c) = \{a, c\}$$

$$T(d) = \{b, c, d\}$$

$$T(e) = \{e\}.$$

Vì e là đỉnh cô lập và d là đỉnh không có cung đi ra nên giữ lại. Vì $T(a), T(c) \subseteq T(b)$ nên bỏ các đỉnh a, c . Vì tập $\{b, d, e\}$ nhỏ nhất thỏa mãn $T(b) \cup T(d) \cup T(e) = V$ nên $\{b, d, e\}$ là tập ổn định ngoài bé nhất của G .

Ngoài ra, loại bỏ các đỉnh a, b ta cũng thu được tập $\{c, d, e\}$ là một tập ổn định ngoài nhỏ nhất của G .

3.7.3. Nhân đồ thị

3.7.3.1. Định nghĩa

Cho đồ thị $G = (V, E)$. *Nhân* (Kernel) của G là một tập con các đỉnh của G sao cho nó vừa là tập ổn định trong và vừa là tập ổn định ngoài của G .

Như vậy, theo quan điểm ánh xạ kề, tập $K \subseteq V$ là một nhân của G khi và chỉ khi

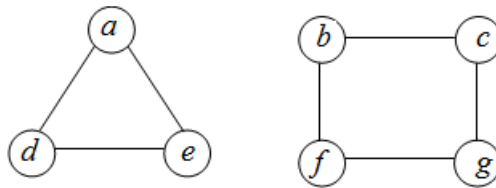
$$\forall v \in K : K \cap F(v) = \emptyset \text{ và } \forall u \notin K : K \cap F(u) \neq \emptyset$$

hay nói một cách khác

$$F^{-1}(K) = V \setminus K.$$

Giả sử K là nhân của G . Suy ra K là tập ổn định trong và do đó K không chứa khuyên. Mặt khác, K cũng là tập ổn định ngoài, nên K phải chứa tất cả các đỉnh cô lập và các đỉnh không có cung đi ra.

Ví dụ 3.45. Xét các đồ thị sau:



Đồ thị đầu tiên không có nhân, đồ thị thứ hai có hai nhân $\{b, g\}$ và $\{c, f\}$.

Mệnh đề 3.21. Nếu g là một hàm Grundy của đồ thị $G = (V, E)$ thì tập hợp

$$K = \{v \in V | g(v) = 0\}$$

là một nhân của G .

Chứng minh. Với mọi $u, v \in K$ thì $g(u) = g(v)$, nên đỉnh u không thể kề với đỉnh v . Suy ra K là một tập ổn định trong.

Mặt khác, nếu $v \notin K$ thì $g(v) > 0$. Khi đó, với $x = 0 < g(v)$ thì sẽ tồn tại $u \in F(v)$ sao cho $g(u) = x = 0$. Suy ra $u \in K$. Điều này có nghĩa K cũng là một tập ổn định ngoài. $\square$

3.7.3.2. Tính chất

Định lý 3.22. Nếu K là nhân của đồ thị G thì K cũng là tập ổn định trong cực đại của G .

Chứng minh. Chứng minh phản chứng. Giả sử K không phải là tập ổn định trong cực đại của G . Suy ra, tồn tại một đỉnh $u \notin K$ sao cho $K \cup \{u\}$ vẫn là tập ổn định trong của G . Vì K là nhân nên u sẽ kề với ít nhất một đỉnh nào đó trong K . Điều này có nghĩa, $K \cup \{u\}$ không thể là tập ổn định trong. Mâu thuẫn với giả sử trên. Vậy K phải là tập ổn định trong cực đại của G . $\square$

Xét đồ thị trong Ví dụ 3.45 (b), ta có $I = \{a\}$ là tập ổn định trong cực đại nhưng không phải là nhân.

Mệnh đề ngược lại của Định lý 3.22 là không đúng. Tuy nhiên, với đồ thị đối xứng thì mệnh đề ngược lại vẫn còn đúng [13].

Định lý 3.23. Trong đồ thị đối xứng không có khuyên, mọi tập ổn định trong cực đại đều là nhân.

Chứng minh. Giả sử I là tập ổn định trong cực đại của đồ thị đối xứng không có khuyên $G = (V, E)$. Theo tính chất cực đại của I , suy ra với mọi $v \in V \setminus I$, tồn tại $u \in I$ sao cho $u \in F(v)$. Điều này có nghĩa, I cũng là tập ổn định ngoài. Do đó, theo định nghĩa I là một nhân của G . $\square$

Xét đồ thị $G = (V, E)$ và $A \subset V$. Ký hiệu $D(A)$ là tập các đỉnh thuộc $V \setminus A$ sao cho mỗi đỉnh này nối với ít nhất một đỉnh thuộc A .

Hệ quả 3.6. Mọi đồ thị đối xứng không có khuyên đều có nhân.

Chứng minh. Giả sử $G = (V, E)$ là đồ thị đối xứng không có khuyên. Suy ra, mỗi đỉnh của G đều lập thành một tập ổn định trong. Xuất phát một đỉnh tùy ý u_1 , ta đặt $K_1 = \{u_1\}$. Sau đó chọn một đỉnh bất kỳ $u_2 \notin D(K_1)$ và đặt $K_2 = K_1 \cup \{u_2\} = \{u_1, u_2\}$. Tiếp tục quá trình như vậy với việc chọn $u_3 \notin D(K_2)$ và bổ sung nó vào K_2 để thu được K_3 . Bởi G hữu hạn nên phải tồn tại một số nguyên dương n sao cho $D(K_n) = V \setminus K_n$. Với cách chọn như thế này rõ ràng K_n là một tập ổn định trong cực đại. Khi đó, theo Định lý 3.23 K_n là nhân của G . $\square$

Chứng minh Hệ quả 3.6 cũng chính là thuật toán tìm một nhân của một đồ thị đối xứng không có khuyên.

Định lý 3.24. *Nếu đồ thị có số ổn định trong nhỏ hơn số ổn định ngoài thì nó không có nhân.*

Chứng minh. Xét đồ thị $G = (V, E)$. Giả sử $\alpha(G) < \beta(G)$ và G có nhân là K . Khi đó theo định nghĩa ta có

$$\begin{aligned}\alpha(G) &= \max\{|I| \mid I \in H(G)\} \\ &\geq |K| \\ &\geq \min\{|D| \mid D \in S(G)\} \\ &= \beta(G).\end{aligned}$$

Điều này là mâu thuẫn với giả sử $\alpha(G) < \beta(G)$. Do đó G không thể có nhân. $\square$

Đồ thị trong Ví dụ 3.45(b) không có nhân theo Định lý 3.24.

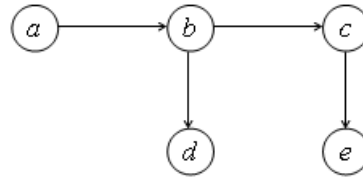
Định lý 3.25. *Mọi đồ thị không có chu trình luôn có nhân.*

Chứng minh. Theo Định lý 3.15 đồ thị không có chu trình có duy nhất một hàm Grundy. Theo Mệnh đề 3.21 nhân của đồ thị này chính là tập các đỉnh có hàm Grundy bằng 0. $\square$

Rõ ràng, câu hỏi tự nhiên đặt ra khi nào đồ thị có chu trình thì có nhân? Để trả lời câu hỏi này chúng ta cần thêm khái niệm sau. Tập con các đỉnh $C \subseteq V$ được gọi là *lõi* của đồ thị $G = (V, E)$ nếu C thỏa mãn các điều kiện sau:

- (i) Với mọi $u, v \in C, u \neq v$, không tồn tại đường đi nối u với v .
- (ii) Với mọi $v \notin C$, tồn tại đường đi từ v đến C .

Ví dụ 3.46. Xét đồ thị G như sau:



Ta có tập đỉnh $\{a, d, e\}$ là lõi và cũng là nhân của G .

Đầu tiên chúng ta có kết quả sau:

Bổ đề 3.1. Mọi đồ thị đều có lõi.

Chứng minh. Chúng ta chứng minh quy nạp theo số đỉnh của đồ thị. Trường hợp, nếu đồ thị G có một đỉnh duy nhất thì rõ ràng đỉnh đó cũng chính là lõi của G . $\square$

Định lý 3.26. Mọi đồ thị không có chu trình độ dài lẻ luôn có nhân.

Định lý 3.27. Nếu mỗi đồ thị con của đồ thị G đều có nhân thì G có hàm Grundy.

Hệ quả 3.7. Đồ thị đối xứng có hàm Grundy khi và chỉ khi nó không có khuyên.

Chứng minh. Chúng ta đã biết đồ thị có hàm Grundy thì không có khuyên.

Ngược lại giả sử đồ thị đối xứng G không có khuyên. Khi đó theo Hệ quả 3.6 mọi đồ thị con của G đều có nhân. Suy ra G có hàm Grundy theo Định lý 3.27. $\square$

3.7.3.3. Thuật toán

Sau đây là thuật toán tìm một nhân của một đồ thị.

Thuật toán 3.7 (Tìm một nhân của đồ thị)

Vào: Đồ thị $G = (V, E)$.

Ra: Nhân của G .

Phương pháp:

Bước 1. Tìm một tập ổn định ngoài nhỏ nhất của G .

Bước 2. Kiểm tra xem nó có phải tập ổn định trong hay không. Nếu đúng thì đó là nhân bé nhất.

Bước 3. Tăng dần số phần tử của tập ổn định ngoài và lặp lại phép kiểm tra, để nhận được các nhân khác.

Ví dụ 3.47. Xét đồ thị G trong Ví dụ 3.44. Ta đã biết $\{b, d, e\}$ là một tập ổn định ngoài nhỏ nhất. Hơn nữa, dễ thấy $\{b, d, e\}$ cũng là một tập ổn định trong, do đó $\{b, d, e\}$ là một nhân của G .

3.7.3.4. Ứng dụng nhân vào lý thuyết trò chơi

Đầu tiên chúng ta giới thiệu trò chơi hai đấu thủ, thường được gọi là *trò chơi Nim*. Trò chơi này được mô tả như sau:

- (1) Có một tập hữu hạn các *hình trạng*.
- (2) Cho phép chuyển một hình trạng sang một hình trạng khác, gọi là *nước đi*.
- (3) Có một tập con các hình trạng gọi là tập các *hình trạng kết thúc*.
- (4) Xuất phát từ một hình trạng, hai đấu thủ lần lượt chọn nước đi. Người nào rơi vào hình trạng kết thúc là người thua cuộc.

Trò chơi Nim có thể biểu diễn bằng đồ thị $G = (V, F)$ như sau:

- (1) Tập đỉnh V chính là tập các hình trạng.
- (2) Với mỗi hình trạng v , thì $F(v)$ là tập các hình trạng có thể chuyển đến trực tiếp từ v bằng các nước đi.

Nếu đồ thị biểu diễn trò chơi có nhân và tất cả các hình trạng kết thúc đều nằm trong nhân thì ta có chiến lược chắc chắn thắng sau đây:

- (1) Tìm cách đưa đối thủ vào nhân.
- (2) Khi đối thủ đã ở trong nhân, thì đối thủ chọn nước đi nào cũng đều đi đến một hình trạng nằm ngoài nhân.
- (3) Đến lượt ta đi, khi đang ở ngoài nhân ta luôn chọn được nước đi để đưa đối thủ trở vào nhân. Khi đó đối thủ chắc chắn bị thua.

Ví dụ 3.48 (Trò chơi bốc que). Giả sử ta có m que và một số nguyên dương cho trước k ($k \leq m$). Hai người tham gia cuộc chơi bốc que. Đến lượt đi, người chơi bốc một số que không vượt quá k . Ai bốc được chiếc que cuối cùng thì người đó thắng cuộc.

3.7.4. Clique

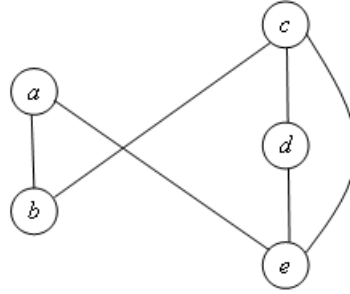
Cho đồ thị vô hướng $G = (V, E)$. Một *clique* của G là một tập con các đỉnh của G sao cho hai đỉnh bất kỳ thuộc nó đều kề nhau.

Một tập con các đỉnh của G được gọi là *clique cực đại* (Maximal clique) nếu không thể thêm bất kỳ đỉnh nào vào nó để tạo ra clique có số đỉnh lớn hơn, tức là nó không chứa trong bất kỳ một clique nào. Clique có nhiều phần tử nhất của G được gọi là *clique lớn nhất* (Maximum clique) của G .

Rõ ràng, một clique lớn nhất cũng là một clique cực đại, tuy nhiên điều ngược lại không đúng. Từ định nghĩa, chúng ta có thể thấy một đồ thị con sinh bởi một

clique của G là một đồ thị con đầy đủ của G . Do đó, trong một số trường hợp người ta cũng xem thuật ngữ clique như là một đồ thị con đầy đủ (tuy không thuyết phục, bởi clique chỉ gồm các đỉnh).

Ví dụ 3.49. Xét đồ thị G như sau:



Đây là đồ thị bù của đồ thị G trong Ví dụ 3.38 ở trên. Ta có $\{c, d\}$, $\{d, e\}$, $\{c, d, e\}$ là các clique của G . Tập $\{c, d, e\}$ là clique cực đại và cũng là clique lớn nhất của G .

Số đỉnh của clique lớn nhất của đồ thị G được gọi là *chỉ số clique* (clique number) của G , ký hiệu $\omega(G)$. Chẳng hạn, đồ thị G ở Ví dụ 3.49 ngay trên có $\omega(G) = 3$. Tập ổn định trong và clique có mối quan hệ sau. Kết quả này dễ dàng suy ra từ định nghĩa.

Định lý 3.28. *Một tập con các đỉnh trong đồ thị G là một tập ổn định trong khi và chỉ khi nó là một clique trong đồ thị bù của G .*

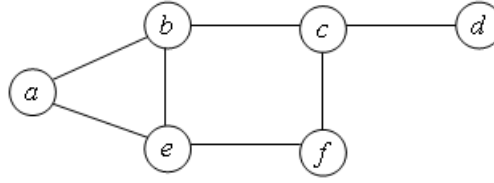
Từ kết quả này, chúng ta thấy hai khái niệm tập ổn định trong và clique có thể xem là bù nhau. Bởi Lưu ý 3.2 và Định lý 3.28 suy ra *bài toán quyết định clique* (có hay không một clique với kích thước cho trước trong một đồ thị) cũng là một bài toán NP-đầy đủ.

3.7.5. Phủ đỉnh

Cho đồ thị vô hướng $G = (V, E)$. Một *phủ đỉnh*⁷ (Vertex cover) của G là một tập con các đỉnh của G sao cho mỗi cạnh của G đều liên thuộc (hay kề) với ít nhất một đỉnh trong nó. Một phủ đỉnh có ít phần tử nhất được gọi là *phủ đỉnh nhỏ nhất* (Minimum vertex cover). Rõ ràng, tập tất cả các đỉnh của một đồ thị là một phủ đỉnh. Đồ thị hai phía đầy đủ $K_{m,n}$ có một phủ đỉnh nhỏ nhất với kích thước là $\tau(K_{m,n}) = \min(m, n)$.

⁷Một số tác giả còn gọi là *tập đỉnh tựa*.

Ví dụ 3.50. Xét đồ thị G sau đây:



Ta có $\{a, c, d, e\}$ là một phủ đỉnh của G và $\{b, c, e\}$ là phủ đỉnh nhỏ nhất của G .

Tập ổn định trong và một phủ đỉnh trong một đồ thị có mối quan hệ sau đây.

Định lý 3.29. Một tập các đỉnh của một đồ thị là một phủ đỉnh nếu và chỉ nếu phần bù của nó là một tập ổn định trong.

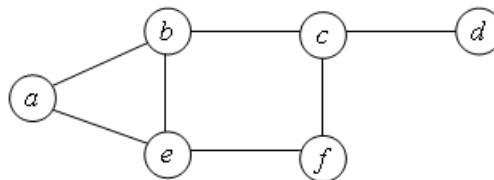
Định lý 3.30 (Gallai, 1959 [3]). Số đỉnh của một đồ thị bằng số đỉnh của phủ đỉnh nhỏ nhất cộng với số đỉnh của tập ổn định trong lớn nhất của đồ thị đó.

Lưu ý 3.4. Bài toán phủ đỉnh nhỏ nhất là một bài toán tối ưu NP-khó. Bài toán quyết định phủ đỉnh (có hay không một phủ đỉnh có kích thước bằng $k \in \mathbb{Z}^+$ cho trước) là một trong 21 bài toán NP-đầy đủ của Karp chứng minh năm 1972. Trường hợp một cây thì bài toán phủ đỉnh nhỏ nhất có thể giải bằng một thuật toán tham lam trong thời gian đa thức [10].

3.7.6. Phủ cạnh

Cho đồ thị $G = (V, E)$. Một *phủ cạnh* (Edge cover) của G là một tập con các cạnh của G sao cho mỗi đỉnh trong G kề với ít nhất một cạnh trong nó. Một phủ cạnh có ít phần tử nhất được gọi là *phủ cạnh nhỏ nhất* (Minimum edge cover). Rõ ràng, tập tất cả cạnh của một đồ thị là một phủ cạnh.

Ví dụ 3.51. Xét đồ thị G sau:



Ta có $\{(a, b), (b, c), (b, e), (c, d), (c, f)\}$ là một phủ cạnh và $\{(a, b), (c, d), (e, f)\}$ là một phủ cạnh nhỏ nhất của G .

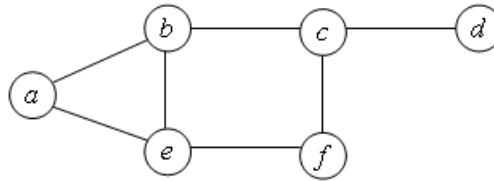
Số cạnh trong một phủ cạnh nhỏ nhất của G được gọi là *số phủ cạnh* (Edge cover number) của G , ký hiệu $\rho(G)$. Chẳng hạn, đồ thị G trong Ví dụ 3.51 có $\rho(G) = 3$. Đồ thị hai phía đầy đủ $K_{m,n}$ có $\rho(K_{m,n}) = \max(m, n)$.

Lưu ý 3.5. Bài toán phủ cạnh nhỏ nhất (tìm một phủ cạnh có kích thước nhỏ nhất) là một bài toán tối ưu thuộc lớp *bài toán phủ*⁸, và có thể giải quyết bằng một thuật toán thời gian đa thức.

3.7.7. Đối sánh

Cho đồ thị $G = (V, E)$. Một *đối sánh*⁹ (Matching) trong G là một tập con các cạnh của G sao cho không có hai cạnh bất kỳ nào của nó là kề nhau. Một tập con các cạnh của G được gọi là một *đối sánh cực đại* (Maximal matching) nếu không thể thêm bất kỳ cạnh nào vào nó để tạo ra đối sánh có số cạnh lớn hơn, tức là nó không chứa trong bất kỳ một đối sánh nào. Một đối sánh có nhiều cạnh nhất được gọi là *đối sánh lớn nhất* (Maximum matching). Rõ ràng, đối sánh lớn nhất là một đối sánh cực đại, tuy nhiên ngược lại thì không đúng. Trong một đồ thị, đối sánh lớn nhất có thể có nhiều.

Ví dụ 3.52. Xét đồ thị G sau đây:



Ta có $\{(a, b), (e, f)\}$ là một đối sánh, nhưng $\{(a, b), (b, c)\}$ không phải là một đối sánh. Tập $\{(b, c), (c, f)\}$ là đối sánh cực đại nhưng không phải đối sánh lớn nhất. Tập $\{(a, b), (e, f), (c, d)\}$ là đối sánh lớn nhất.

Số cạnh của một đối sánh lớn nhất của đồ thị G được gọi là *số đối sánh* (Matching number) của G , ký hiệu $\nu(G)$. Chẳng hạn, đồ thị G ở Ví dụ 3.52 có $\nu(G) = 3$.

Mệnh đề 3.31. Nếu M_1 và M_2 là hai đối sánh cực đại thì $|M_1| \leq 2|M_2|$ hoặc $|M_2| \leq 2|M_1|$.

Định lý 3.32 (Gallai, 1959 [3]). Trong một đồ thị bất kỳ không có đỉnh cô lập, tổng số đối sánh và số phủ cạnh là bằng số đỉnh của đồ thị.

⁸Bài toán phủ tập (Set cover): cho một tập vũ trụ $U = \{1, 2, \dots, m\}$ và một họ S các tập con của U sao cho $\bigcup S = U$. Xác định một họ con nhỏ nhất $C \subset S$ sao cho $\bigcup C = U$. Bài toán quyết định phủ tập là một bài toán NP-đầy đủ, trong khi đó bài toán tối ưu phủ tập là một bài toán NP-khó.

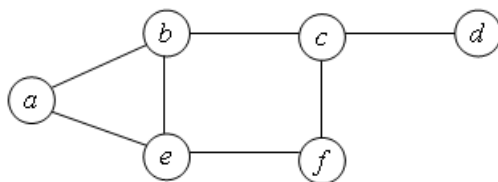
⁹Một tác giả còn gọi là *cặp ghép*, hay *tập cạnh độc lập*-Independent edge set.

Lưu ý 3.6. Một đối sánh cực đại có thể tìm thấy bằng một thuật toán tham lam đơn giản thời gian đa thức. Ta biết rằng một đối sánh lớn nhất cũng là một đối sánh cực đại, do đó có thể tìm thấy một đối sánh cực đại lớn nhất trong thời gian đa thức. Tuy nhiên, không có thuật toán thời gian đa thức để tìm thấy một *đối sánh cực đại nhỏ nhất* (đối sánh cực đại có ít cạnh nhất).

3.7.8. Tập thống trị cạnh

Cho đồ thị $G = (V, E)$. Một *tập thống trị cạnh*¹⁰ (Edge dominating set) của G là một tập con các cạnh của G sao cho mỗi cạnh không thuộc nó đều kề với ít nhất một cạnh trong nó. Một tập thống trị cạnh có ít cạnh nhất được gọi là *tập thống trị cạnh nhỏ nhất* (Minimum edge dominating set).

Ví dụ 3.53. Xét đồ thị G như sau:



Ta có $\{(a, e), (c, d)\}$ là một tập thống trị cạnh. Tập $\{(a, b), (a, e), (b, e)\}$ không phải là tập thống trị cạnh. Tập $\{(a, e), (c, d)\}$ cũng là tập thống trị cạnh nhỏ nhất.

Định lý 3.33 (Yannakakis and Gavril, 1980[14]). *Trong một đồ thị bất kỳ, một đối sánh cực đại là một tập thống trị cạnh.*

Định lý 3.34. *Trong một đồ thị bất kỳ, kích thước của tập thống trị cạnh nhỏ nhất và đối sánh cực đại nhỏ nhất là bằng nhau.*

Định lý 3.35 (Yannakakis and Gavril, 1980 [14]). *Trong một đồ thị bất kỳ, một đối sánh cực đại nhỏ nhất là một tập thống trị cạnh nhỏ nhất.*

Mệnh đề ngược lại của Định lý 3.35 là không đúng, tức là một tập thống trị cạnh nhỏ nhất chưa chắc là một đối sánh cực đại nhỏ nhất. Chẳng hạn, xét Ví dụ 3.53 ta có $\{(b, c), (b, e)\}$ là một tập thống trị cạnh nhỏ nhất nhưng không phải là một đối sánh, và do đó không phải là một đối sánh cực đại nhỏ nhất. Tuy vậy, nếu cho trước một tập thống trị cạnh nhỏ nhất D thì dễ dàng tìm được một đối sánh cực đại nhỏ nhất gồm $|D|$ cạnh [14].

¹⁰Còn được gọi là *tập thống trị dòng*-Line dominating set.

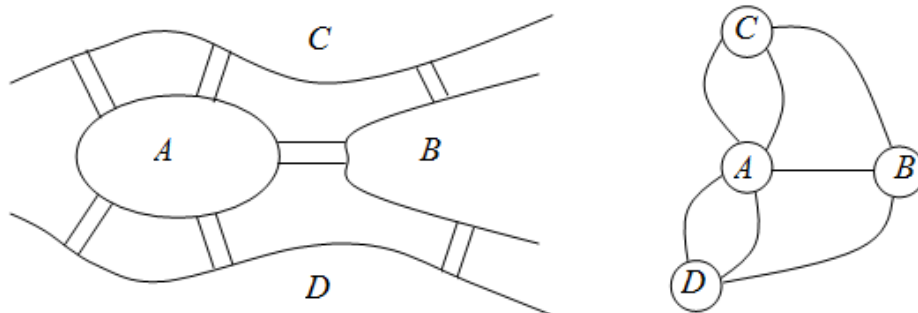
Lưu ý 3.7. Bài toán quyết định tập thống trị cạnh là bài toán NP-đầy đủ, và do đó bài toán tập thống trị cạnh nhỏ nhất là một bài toán tối ưu NP-khó. Yannakakis và Gavril [14] chứng minh rằng bài toán tập thống trị cạnh nhỏ nhất vẫn là bài toán NP-đầy đủ trong trường hợp đồ thị là hai phía với bậc lớn nhất là 3 hay đồ thị phẳng với bậc lớn nhất là 3.

3.8. Đồ thị Euler và đồ thị Hamilton

Mục này giới thiệu về hai dạng đồ thị đặc biệt là đồ thị Euler và đồ thị Hamilton.

3.8.1. Đồ thị Euler

Có thể xem năm 1736 là năm khai sinh lý thuyết đồ thị với việc công bố lời giải “bài toán về các cầu ở Königsberg” của nhà toán học lỗi lạc Euler (1707-1783). Thành phố Königsberg thuộc Phổ (nay gọi là Kaliningrad thuộc Nga) được chia thành bốn vùng bằng các nhánh sông Pregel, các vùng này gồm hai vùng bên bờ sông, đảo Kneiphof và một vùng nằm giữa hai nhánh của sông Pregel. Vào thế kỷ 18, người ta xây 7 chiếc cầu nối các vùng này với nhau. Có thể hình dung sơ đồ của thành phố Königsberg với 7 chiếc cầu như sau:



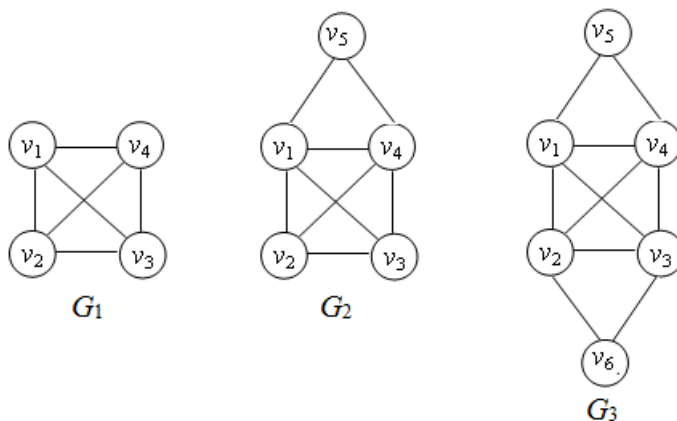
Hình 3.5: Bảy chiếc cầu ở thành phố Königsberg và đồ thị tương ứng

Dân ở thành phố Königsberg từng thắc mắc: “Có thể nào đi dạo qua tất cả bảy chiếc cầu, mỗi cầu chỉ qua một lần thôi hay không?”. Như vậy, nếu ta coi mỗi vùng A, B, C, D như một đỉnh và mỗi chiếc cầu qua lại hai vùng là một cạnh nối hai đỉnh thì ta có sơ đồ của Königsberg là một đa đồ thị G như hình trên. Khi đó bài toán tìm đường đi qua tất cả chiếc cầu, mỗi cầu chỉ qua một lần có thể được phát biểu lại theo mô hình đồ thị như sau: “có tồn tại chu trình đơn trong đa đồ thị G chứa tất cả các cạnh hay không?” Euler chứng minh được không tồn tại một chu trình đơn trong đồ thị G như trên.

3.8.1.1. Định nghĩa

Cho đồ thị $G = (V, E)$. Một đường đi đơn, đi qua tất cả các cạnh của G được gọi là *đường đi Euler* trong G . Trường hợp đường đi Euler có đỉnh cuối trùng với đỉnh xuất phát thì được gọi là *chu trình Euler*. Như vậy, chu trình Euler là một chu trình đơn qua tất cả các cạnh. Đồ thị G được gọi là *đồ thị Euler* nếu G có chứa một chu trình Euler, và gọi là *đồ thị nửa Euler* nếu G có chứa một đường đi Euler. Rõ ràng, đồ thị Euler là nửa Euler, nhưng ngược lại không đúng. Đường đi Euler có thể tìm thấy trong các bài toán vui vẽ một nét, đó là vẽ một hình nào đó mà không nhấc bút khỏi mặt giấy và không tô lại cạnh nào hai lần.

Ví dụ 3.54. Xét các đồ thị sau:



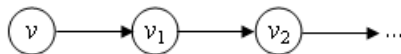
Đồ thị G_1 không phải là đồ thị nửa Euler vì nó không chứa một đường đi Euler nào. Đồ thị G_2 là đồ thị nửa Euler nhưng không phải đồ thị Euler vì nó có chứa một đường đi Euler, chẳng hạn đường đi $P = [v_2, v_1, v_5, v_4, v_1, v_3, v_2, v_4, v_3]$. Đồ thị G_3 là đồ thị Euler vì nó có chứa một chu trình Euler (có thể xuất phát từ bất kỳ đỉnh nào trong G_3 chúng ta luôn đi qua tất cả các cạnh, mỗi cạnh đúng một lần và quay trở lại đỉnh xuất phát).

3.8.1.2. Một số điều kiện cần và đủ

Điều kiện cần và đủ để một đồ thị là đồ thị Euler được Euler tìm ra vào năm 1736 khi ông giải quyết bài toán học búa nổi tiếng bảy chiếc cầu ở thành phố Königsberg và đây cũng chính là định lý đầu tiên của lý thuyết đồ thị. Đầu tiên, chúng ta tìm hiểu trên các đồ thị vô hướng.

Bổ đề 3.2. Cho G là một đồ thị vô hướng liên thông. Nếu bậc mỗi đỉnh của G không nhỏ hơn 2 thì G chứa ít nhất một chu trình đơn.

Chứng minh. Rõ ràng, nếu G có cạnh bội hoặc khuyên thì khẳng định của bổ đề là hiển nhiên. Vì vậy, giả sử G là một đơn đồ thị. Gọi v là một đỉnh bất kỳ của G . Ta sẽ xây dựng theo quy nạp một đường đi như sau:



trong đó đỉnh v_1 là kề với đỉnh v , đỉnh v_{i+1} ($i \geq 1$) kề với đỉnh v_i nhưng $v_{i+1} \neq v_{i-1}$ (luôn chọn được vì $\deg(v_i) \geq 2$ với mỗi v_i của G) và $v_0 = v$. Vì G hữu hạn, nên sau một số hữu hạn bước ta phải quay trở lại một đỉnh đã chọn trước đó. Gọi k là số nguyên dương đầu tiên để $v_k = v_i$ ($0 \leq i < k$). Khi đó, đường đi $v_i, v_{i+1}, \dots, v_{k-1}, v_k$ là một chu trình đơn cần tìm. $\square$

Một số đồ thị quen thuộc là đồ thị Euler: đồ thị đầy đủ K_n với n lẻ, các đồ thị đều bậc chẵn, đồ thị vòng, đồ thị hai phía đầy đủ $K_{m,n}$ với m, n chẵn.

Định lý 3.36 (Euler, 1736). *Đồ thị vô hướng liên thông G là đồ thị Euler khi và chỉ khi mọi đỉnh của G đều có bậc chẵn.*

Chứng minh. Giả sử G là đồ thị Euler. Suy ra, tồn tại một chu trình Euler P trong G . Biết rằng mỗi lần chu trình P đi qua một đỉnh nào đó của G thì bậc của đỉnh đó tăng lên 2. Mặt khác, mỗi cạnh của G đều xuất hiện trong P đúng một lần. Do đó mỗi đỉnh của G đều có bậc chẵn.

Giả sử mọi đỉnh của G đều có bậc chẵn. Từ đây và giả thiết G liên thông, suy ra mỗi đỉnh của G có bậc không nhỏ hơn 2. Khi đó, theo Bổ đề 3.2 G phải chứa một chu trình đơn C . Rõ ràng, nếu C đi qua tất cả các cạnh của G thì nó chính là chu trình Euler. Giả sử C không đi qua tất cả các cạnh của G . Khi đó, loại bỏ khỏi G các cạnh thuộc C , ta sẽ thu được một đồ thị mới H (không nhất thiết là liên thông). Suy ra, số cạnh trong H ít hơn trong G và mỗi đỉnh của H vẫn có bậc là chẵn. Theo giả thiết quy nạp, trong mỗi thành phần liên thông của H đều tìm được chu trình Euler. $\square$

Hệ quả 3.8. *Đồ thị vô hướng liên thông G là nửa Euler (nhưng không phải Euler) khi và chỉ khi G có đúng hai đỉnh bậc lẻ.*

Chứng minh. Giả sử G là nửa Euler. Suy ra tồn tại một đường đi Euler trong G từ đỉnh u đến đỉnh v . Gọi H là đồ thị thu được từ G bằng cách thêm vào cạnh (u, v) . Khi đó rõ ràng H là đồ thị Euler. Theo Định lý 3.36 suy ra mọi đỉnh của H đều có bậc chẵn, kể cả hai đỉnh u và v . Điều này có nghĩa trong G hai đỉnh u và v là đỉnh bậc lẻ duy nhất.

Giả sử G có đúng hai đỉnh bậc lẻ u và v . Gọi H là đồ thị thu được từ G bằng cách thêm vào cạnh (u, v) . Khi đó mọi đỉnh của H đều có bậc chẵn. Suy ra H là đồ thị Euler theo Định lý 3.36. Bỏ cạnh (u, v) đã thêm vào ra khỏi chu trình Euler trong H ta có được đường đi Euler từ u đến v trong G , hay G là nửa Euler. $\square$

Đồ thị hai phía đầy đủ $K_{m,n}$ là nửa Euler khi $m = 2$ và n lẻ, hoặc m lẻ và $n = 2$. Với đồ thị có hướng chúng ta cũng có các điều kiện cần và đủ cho đồ thị Euler và nửa Euler.

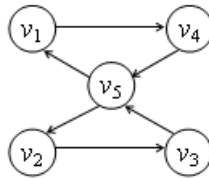
Bổ đề 3.3. Cho G là một đồ thị có hướng liên thông. Khi đó nếu bậc vào và bậc ra của mỗi đỉnh của G không nhỏ hơn 1 thì G chứa chu trình đơn.

Định lý 3.37. Đồ thị có hướng liên thông mạnh G là đồ thị Euler khi và chỉ khi mọi đỉnh của G đều có bậc vào bằng bậc ra.

Chứng minh. Điều kiện cần chứng minh tương tự Định lý 3.36. Điều kiện đủ suy ra từ $\square$

Chứng minh của Bổ đề 3.3 và Định lý 3.37 là tương tự như Bổ đề 3.2 và Định lý 3.36.

Ví dụ 3.55. Đồ thị có hướng liên thông mạnh sau là đồ thị Euler theo Định lý 3.37:



Với chu trình Euler: $C = [v_1, v_4, v_5, v_2, v_3, v_5, v_1]$.

Hệ quả 3.9. Đồ thị có hướng liên thông yếu $G = (V, E)$ là nửa Euler (nhưng không phải Euler) khi và chỉ khi tồn tại hai đỉnh x và y sao cho:

$$\begin{aligned} \deg^-(x) &= \deg^+(x) + 1, \deg^+(y) = \deg^-(y) + 1, \\ \deg^+(v) &= \deg^-(v), \forall v \in V, v \neq x, v \neq y \end{aligned}$$

Chứng minh. Chứng minh tương tự như Hệ quả 3.8. $\square$

3.8.1.3. Thuật toán Fleury

Ta có thể vạch được một chu trình Euler trong đồ thị vô hướng liên thông G có bậc của mọi đỉnh là chẵn theo Thuật toán Fleury sau đây.

Thuật toán 3.8 Thuật toán Fleury

Vào: Đồ thị vô hướng liên thông $G = (V, E)$ với mọi đỉnh của G đều có bậc chẵn.

Ra: Một chu trình Euler.

Phương pháp:

Bước 1. (*Khởi tạo*) Xuất phát từ một đỉnh bất kỳ v của G .

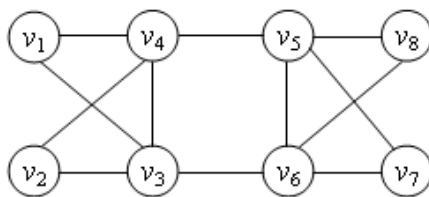
Bước 2. (*Lắp*) Từ đỉnh v ta đi theo các cạnh liên thuộc của nó một cách tùy ý chỉ cần tuân thủ hai quy tắc sau:

(1) Mỗi khi đi qua một cạnh nào thì xóa nó đi, sau đó xóa đỉnh cô lập nếu có.

(2) Không bao giờ đi qua một cầu, trừ phi không còn cách đi nào khác.

Bước 3. (*Kết luận*) Các cạnh theo thứ tự đã xóa chính là một chu trình Euler cần tìm.

Ví dụ 3.56. Tìm chu trình Euler của đồ thị G sau:



Bước 1: xuất phát từ đỉnh v_1 có thể đi theo cạnh (v_1, v_4) hoặc (v_1, v_3) , giả sử là cạnh (v_1, v_4) (xóa cạnh (v_1, v_4)).

Bước 2: từ đỉnh v_4 có thể đi theo một trong các cạnh (v_4, v_5) , (v_4, v_2) , (v_4, v_3) , giả sử là cạnh (v_4, v_5) (xóa cạnh (v_4, v_5)).

Bước 3: từ đỉnh v_5 có thể đi theo một trong các cạnh (v_5, v_8) , (v_5, v_6) , (v_5, v_7) , giả sử là cạnh (v_5, v_8) (xóa cạnh (v_5, v_8)).

Bước 4: từ đỉnh v_8 , đi theo cạnh (v_8, v_6) (xóa (v_8, v_6) và đỉnh v_8).

Bước 5: vì (v_6, v_3) là cầu nên có thể đi theo một trong hai cạnh (v_6, v_5) , (v_6, v_7) , giả sử là cạnh (v_6, v_5) (xóa cạnh (v_6, v_5)).

Bước 6: đi theo (v_5, v_7) (xóa cạnh (v_5, v_7) và đỉnh v_5).

Bước 7: đi theo (v_7, v_6) (xóa cạnh (v_7, v_6) và đỉnh v_7).

Bước 8: tiếp tục đi theo cạnh (v_6, v_3) (xóa cạnh (v_6, v_3) và đỉnh v_6).

Bước 9: vì (v_3, v_1) là cầu nên đi theo cạnh (v_3, v_4) hoặc (v_3, v_2) , giả sử là cạnh (v_3, v_4) (xóa cạnh (v_3, v_4)).

Bước 10: tiếp tục đi theo (v_4, v_2) (xóa cạnh (v_4, v_2) và đỉnh v_4).

Bước 11: theo cạnh (v_2, v_3) (xóa cạnh (v_2, v_3) và đỉnh v_2).

Bước 12: Cuối cùng đi theo cạnh (v_3, v_1) (xóa cạnh (v_3, v_1) , đỉnh v_3 và đỉnh v_1).

Như vậy một chu trình Euler là $C = [v_1, v_4, v_5, v_8, v_6, v_5, v_7, v_6, v_3, v_4, v_2, v_3, v_1]$.

3.8.1.4. Bài toán người đưa thư Trung Hoa

Bài toán người đưa thư Trung Hoa (Chinese postman problem) được đưa ra bởi nhà toán học Trung Hoa Mei-Ku Guan lần đầu tiên vào năm 1962. Bài toán này được phát biểu là: “*một người đưa thư xuất phát từ bưu điện phải đến một số con đường để phát thư rồi quay trở về điểm xuất phát. Hỏi người đó phải đi như thế nào để số đường đi là ít nhất*”.

Bài toán này có thể giải bằng phương pháp đồ thị như sau: dựng một đồ thị liên thông G có các cạnh tương ứng với các con đường mà người đưa thư phải đi qua. Khi đó đi tìm một hành trình ngắn nhất của đồ thị G .

Xét hai trường hợp:

- G là đồ thị Euler (mọi đỉnh đều có bậc chẵn). Khi đó theo Định lý Euler rõ ràng chu trình Euler trong G chính là hành trình ngắn nhất cần tìm.
- G có một số đỉnh bậc lẻ. Khi đó mọi hành trình trong G phải đi qua ít nhất hai lần một số cạnh nào đó. Rõ ràng một hành trình qua một cạnh nào đó quá hai lần thì không phải là hành trình ngắn nhất trong G . Vì vậy chỉ cần xét những hành trình T đi qua hai lần một số cạnh nào đó của G . Ta quy ước mỗi hành trình T trong G là một hành trình trong đồ thị Euler G_T , là đồ thị có được từ G bằng cách vẽ thêm một cạnh song song đối với những cạnh mà T đi qua hai lần. Khi đó bài toán người đưa thư Trung Hoa có thể đưa về bài toán sau: “trong số các đồ thị Euler G_T , tìm đồ thị có số cạnh ít nhất (chu trình Euler trong đồ thị này chính là hành trình ngắn nhất cần tìm)”. Trong trường hợp này, bài toán có thể giải dựa vào định lý sau đây.

Định lý 3.38 (Goodman và Hedetniemi, 1973). *Nếu G là một đồ thị liên thông có q cạnh thì hành trình ngắn nhất trong G có chiều dài*

$$q + m(G)$$

trong đó $m(G)$ là số cạnh mà hành trình đi qua đúng hai lần và được xác định như sau:

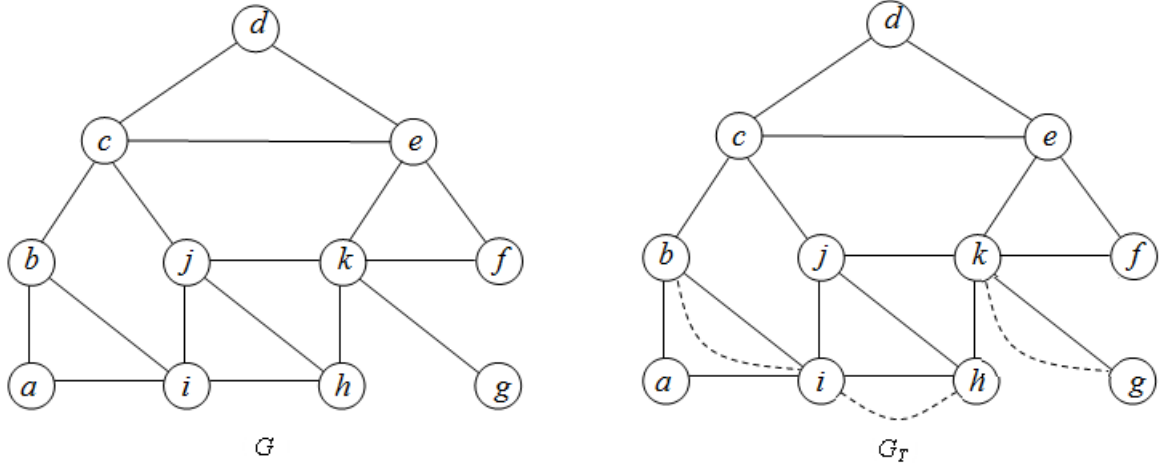
Gọi $V_0(G)$ là tập hợp các đỉnh bậc lẻ của G , giả sử có $2k$ đỉnh như vậy. Ta phân hoạch $2k$ đỉnh này thành k cặp, trong đó mỗi tập hợp k cặp được gọi là một phân hoạch cặp P_i của $V_0(G)$. Ký hiệu $d_{\min}(u, v)$ là độ dài đường đi ngắn nhất từ đỉnh

u đến đỉnh v . Khi đó

$$m(G) = \min(d(P_i))$$

với $d(P_i) = \sum_{(u,v) \in P_i} d_{\min}(u,v)$ (gọi là độ dài đường đi ngắn nhất của phân hoạch P_i).

Ví dụ 3.57. Giải bài toán người đưa thư Trung Hoa cho bởi đồ thị G như sau:



Đồ thị G không phải là đồ thị Euler. Do đó ta sử dụng Định lý Goodman-Hedetniemi để tìm hành trình ngắn nhất trong G . Số cạnh của G là $q = 15$ và tập các đỉnh bậc lẻ là $V_0(G) = \{b, g, h, k\}$. Có 3 cách phân hoạch tập $V_0(G)$ thành 2 cặp như sau:

$$P_1 = \{(b, g), (h, k)\}, \text{ với } d(P_1) = d(b, g) + d(h, k) = 4 + 1 = 5$$

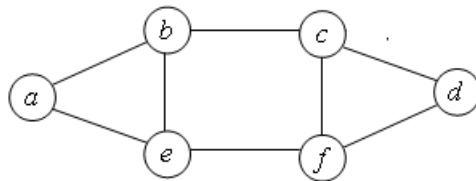
$$P_2 = \{(b, h), (g, k)\}, \text{ với } d(P_2) = d(b, h) + d(g, k) = 2 + 1 = 3$$

$$P_3 = \{(b, k), (g, h)\}, \text{ với } d(P_3) = d(b, k) + d(g, h) = 3 + 2 = 5.$$

Như vậy $m(G) = \min(d(P_1), d(P_2), d(P_3)) = 3$. Suy ra hành trình ngắn nhất trong G có độ dài là $q + m(G) = 15 + 3 = 18$.

Vẽ đồ thị G_T từ G bằng cách thêm vào 3 cạnh (b, i) , (i, h) và (g, k) . Lúc này G_T là đồ thị Euler và hành trình ngắn nhất cần tìm là đi theo chu trình Euler trong G_T , chẳng hạn $C = [a, b, c, d, e, f, g, k, e, c, j, k, h, j, i, h, i, b, i, a]$.

Ví dụ 3.58. Giải bài toán người đưa thư Trung Hoa cho bởi đồ thị G như sau:



Đồ thị G không phải là đồ thị Euler. Do đó ta sử dụng Định lý Goodman-Hedetniemi để tìm hành trình ngắn nhất trong G . Số cạnh của G là $q = 8$ và tập các đỉnh bậc lẻ là $V_0(G) = \{b, c, e, f\}$. Có 3 cách phân hoạch tập $V_0(G)$ thành 2 cặp như sau:

$P_1 = \{(b, c), (e, f)\}$, với $d(P_1) = d(b, c) + d(e, f) = 1 + 1 = 2$

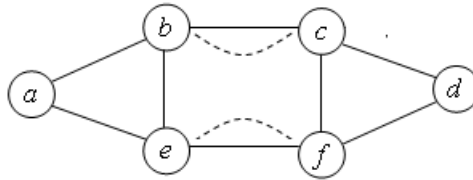
$P_2 = \{(b, e), (c, f)\}$, với $d(P_2) = d(b, e) + d(c, f) = 1 + 1 = 2$

$P_3 = \{(b, f), (c, e)\}$, với $d(P_3) = d(b, f) + d(c, e) = 2 + 2 = 4$.

Như vậy $m(G) = \min(d(P_1), d(P_2), d(P_3)) = 2$. Suy ra hành trình ngắn nhất trong G có độ dài là $q + m(G) = 8 + 2 = 10$.

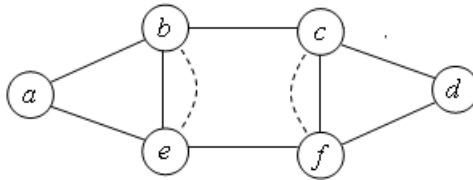
Có hai hành trình ngắn nhất 10 cạnh như thế:

(1) Vẽ thêm 2 cạnh phụ (b, c) và (e, f) vào đồ thị G :



Đồ thị mới là đồ thị Euler và khi đó chu trình Euler trong đồ thị mới này chính là hành trình ngắn nhất cần tìm: $C = [a, b, c, d, f, e, b, c, f, e, a]$.

(2) Vẽ thêm 2 cạnh phụ (b, e) và (c, f) vào đồ thị G :



Đồ thị mới là đồ thị Euler và chu trình Euler trong đồ thị mới này chính là hành trình ngắn nhất cần tìm: $C = [a, b, c, d, f, c, f, e, b, e, a]$.

3.8.2. Đồ thị Hamilton

Năm 1859 nhà toán học người Ailen William Rowan Hamilton (1805-1865) đưa ra trò chơi *đi vòng quanh thế giới* như sau “cho một khối thập nhị diện đều (khối có 12 mặt, 20 đỉnh và 30 cạnh), trong đó mỗi đỉnh của khối tên một thành phố nổi tiếng và mỗi cạnh của khối (nối hai đỉnh) là đường đi lại giữa hai thành phố tương ứng. Xuất phát từ một thành phố, hãy tìm đường đi thăm tất cả các thành phố khác, mỗi thành phố chỉ một lần rồi quay trở về thành phố ban đầu”.

Trước Hamilton, có thể là từ thời Euler, người ta cũng đã biết đến một câu đố hóc búa về *đường đi của con mã trên bàn cờ* như sau “cho bàn cờ có 8×8 ô vuông. Hãy tìm đường đi của con mã qua tất cả các ô của bàn cờ, mỗi ô chỉ một lần rồi trở về ô xuất phát”. Đây là bài toán được nhiều nhà toán học quan tâm, đặc biệt là Euler, De Moivre, Vandermonde, ...

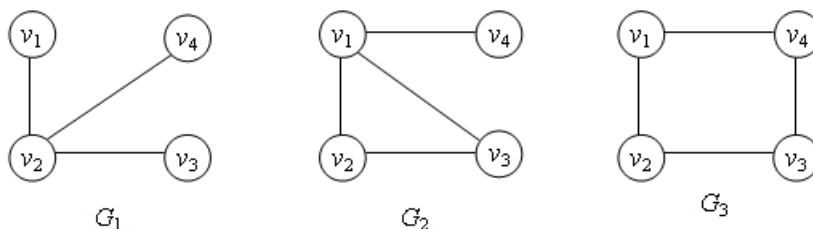
Chính trò chơi và câu đố trên đã dẫn tới việc khảo sát một lớp đồ thị đặc biệt có tên gọi đồ thị Hamilton.

3.8.2.1. Định nghĩa

Cho đồ thị $G = (V, E)$. Một đường đi sơ cấp qua tất cả các đỉnh của G được gọi là *đường đi Hamilton*. Trường hợp đường đi Hamilton có đỉnh cuối cùng trùng với đỉnh xuất phát thì được gọi là *chu trình Hamilton*. Như vậy, chu trình Hamilton là một chu trình sơ cấp qua tất cả các đỉnh. Đồ thị G được gọi là *đồ thị Hamilton* nếu nó có chứa một chu trình Hamilton và gọi là *đồ thị nửa Hamilton* nếu nó có chứa một đường đi Hamilton.

Rõ ràng, đồ thị Hamilton là đồ thị nửa Hamilton, tuy nhiên ngược lại không đúng. Có thể thấy ngay, đồ thị đầy đủ K_3 là đồ thị Hamilton và K_2 là đồ thị nửa Hamilton.

Ví dụ 3.59. Xét các đồ thị sau:



Đồ thị G_1 không phải nửa Hamilton, đồ thị G_2 là nửa Hamilton (đường đi Hamilton $P = [v_4, v_1, v_3, v_2]$) và đồ thị G_3 là Hamilton (chu trình Hamilton $C = [v_1, v_2, v_3, v_4, v_1]$).

Dễ thấy, nếu G là đồ thị Hamilton thì G không có điểm khớp và mỗi đỉnh có bậc không nhỏ hơn 2.

Có thể thấy, đường đi Hamilton gần như tương tự đường đi Euler trong cách phát biểu. Tuy nhiên, nếu như bài toán tìm đường đi Euler trong một đồ thị đã được giải quyết trọn vẹn, dấu hiệu nhận biết một đồ thị Euler là khá đơn giản và dễ sử dụng thì các bài toán tìm đường đi Hamilton và xác định đồ thị Hamilton lại khó hơn rất nhiều. Cho đến bây giờ việc tìm một tiêu chuẩn nhận biết đồ thị Hamilton vẫn còn là mở, mặc dù đây là một vấn đề trung tâm của lý thuyết đồ thị. Phần

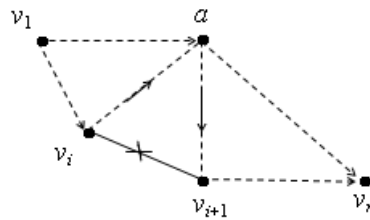
lớn các kết quả thu được chỉ là một vài điều kiện đủ để nhận biết một lớp rất nhỏ các đồ thị Hamilton và đồ thị nửa Hamilton. Phần lớn chúng đều có dạng “*nếu đồ thị G có số cạnh đủ lớn thì G là Hamilton*”. Và do đó cũng chưa có thuật toán hiệu quả nào để kiểm tra một đồ thị có phải là đồ thị Hamilton hay không.

3.8.2.2. Một số tính chất cơ bản

Định lý 3.39 (Rédei). *Đồ thị có hướng đầy đủ là đồ thị nửa Hamilton.*

Chứng minh. Xét đồ thị có hướng đầy đủ G . Ta chứng minh bằng qui nạp theo số đỉnh n của G . Định lý là hiển nhiên cho trường hợp $n = 1, 2$. Giả sử định lý đúng với $n = k$ ($k \geq 3$). Ở đây giả sử G có $k + 1$ đỉnh. Xây dựng đồ thị H từ G bằng cách bỏ đi một đỉnh a và các cạnh liên thuộc với a . Rõ ràng, H còn lại k đỉnh và cũng là đồ thị đầy đủ, nên theo giả thiết qui nạp H là đồ thị nửa Hamilton. Giả sử đường đi Hamilton trong H là $P = [v_1, v_2, \dots, v_k]$. Khi đó, xảy ra các trường hợp sau:

- Nếu trong G có cung (v_k, a) thì đường đi (P, a) sẽ là đường đi Hamilton trong G .
- Nếu trong G có cung (a, v_1) thì đường đi (a, P) sẽ là đường đi Hamilton trong G .
- Còn ngược lại G có các cung ngược hướng nhau (a, v_k) và (v_1, a) . Khi đó sẽ phải có ít nhất một cặp cung sát nhau nhưng ngược hướng nhau là (v_i, a) và (a, v_{i+1}) ($1 \leq i \leq k - 1$):



Khi đó đường đi $P = [v_1, v_2, \dots, v_i, a, v_{i+1}, \dots, v_k]$ là đường đi Hamilton trong G . Tóm lại G luôn là đồ thị nửa Hamilton. $\square$

Chứng minh của Định lý Rédei cũng chính là thuật toán tìm đường đi Hamilton trong đồ thị đầy đủ có hướng. Vận dụng thuật toán này chúng ta sẽ tìm được ngay đường đi Hamilton trong đồ thị có hướng đầy đủ K_3 .

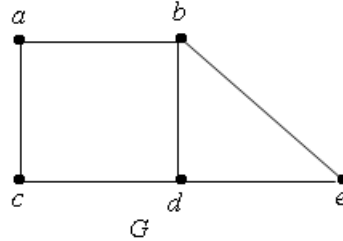
Bây giờ chúng ta tìm hiểu một số điều kiện đủ của đồ thị nửa Hamilton và Hamilton.

Định lý 3.40 (Dirac, 1952). *Đơn đồ thị vô hướng có $n \geq 3$ đỉnh, và bậc của mỗi đỉnh không nhỏ hơn $n/2$ là đồ thị Hamilton.*

Từ Định lý Dirac suy ra ngay mọi đồ thị đầy đủ K_n có nhiều hơn 2 đỉnh là đồ thị Hamilton. Xét đồ thị G_3 ở Ví dụ 3.59. Ta có G_3 là đồ thị Hamilton theo Định lý Dirac.

Mệnh đề ngược lại của Định lý Dirac là không đúng, chẳng hạn đồ thị sau.

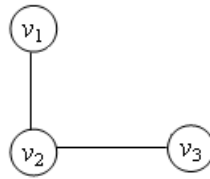
Ví dụ 3.60. Xét đồ thị Hamilton G như sau:



Đỉnh a của G có $\deg(a) = 2$.

Hệ quả 3.10. Đơn đồ thị vô hướng có $n \geq 3$ đỉnh, và bậc của mỗi đỉnh không nhỏ hơn $(n - 1)/2$ là đồ thị nửa Hamilton.

Ví dụ 3.61. Đồ thị sau là đồ thị nửa Hamilton theo Hệ quả 3.10:



Với đường đi Hamilton $P = [v_1, v_2, v_3]$.

Định lý 3.41 (Ore, 1960). Đơn đồ thị vô hướng có $n \geq 3$ đỉnh, và tổng bậc của hai đỉnh bất kỳ không kề nhau luôn không nhỏ hơn n là đồ thị Hamilton.

Xét đồ thị G_3 ở Ví dụ 3.59. Ta cũng có G_3 là đồ thị Hamilton theo Định lý Ore.

Mệnh đề ngược lại của Định lý Ore là không đúng, chẳng hạn đồ thị Hamilton G trong Ví dụ 3.60 có $\deg(a) + \deg(d) = 4$.

Định lý 3.42. Đồ thị hai phía $G = (X \cup Y, E)$ có $|X| = |Y| = n$ ($n \geq 2$), và bậc của mỗi đỉnh lớn hơn $n/2$ là một đồ thị Hamilton.

Như vậy các đồ thị hai phía đầy đủ $K_{n,n}$ với $n \geq 2$ đều là các đồ thị Hamilton.

Đối với đồ thị có hướng chúng ta cũng có một điều kiện đủ như sau.

Định lý 3.43 (Dirac tổng quát). Đồ thị có hướng liên thông mạnh có $n \geq 3$ đỉnh và bậc vào, bậc ra của mỗi đỉnh không nhỏ hơn $n/2$ là đồ thị Hamilton.

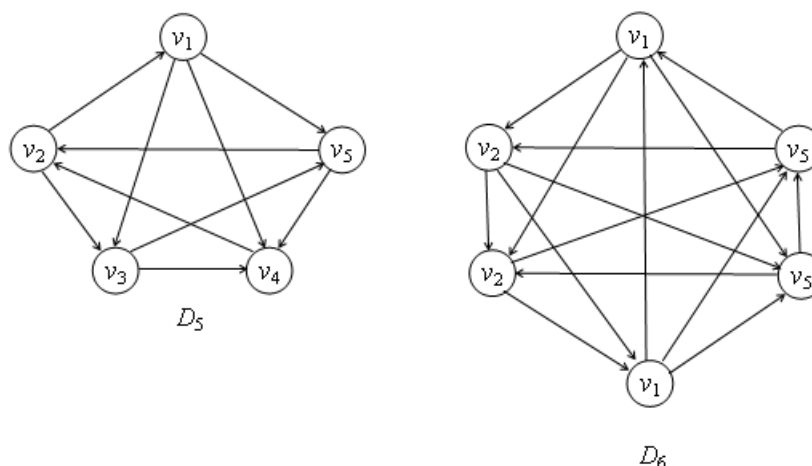
Định lý 3.44 (Meyniel, 1973). *Đồ thị liên thông mạnh có n đỉnh sao cho tổng bậc của hai đỉnh không kề nhau không nhỏ hơn $2n - 1$ là đồ thị Hamilton.*

Có một số dạng đồ thị mà chúng ta có thể biết khi nào nó là đồ thị Hamilton, chẳng hạn đồ thị đấu loại. *Đồ thị đấu loại* là một đồ thị có hướng sao cho giữa hai đỉnh bất kỳ luôn được nối với nhau bởi đúng một cung. Đồ thị này được dùng để biểu diễn kết quả thi đấu của các trò chơi (như bóng chuyền, bóng bàn, ...) nhưng không cho phép hòa. Tên đấu loại xuất hiện là vì ý nghĩa đó.

Định lý 3.45. (1) *Đồ thị đấu loại là nửa Hamilton.*

(2) *Đồ thị đấu loại liên thông mạnh là Hamilton.*

Ví dụ 3.62. Xét các đồ thị có hướng sau:



Ta có D_5 là đồ thị đấu loại, D_6 là đồ thị đấu loại liên thông mạnh.

Thuật toán liệt kê tất cả các chu trình Hamilton của đồ thị được xây dựng dựa trên cơ sở thuật toán quay lui. Trường hợp đồ thị không có quá nhiều cạnh thì thuật toán này có thể sử dụng để kiểm tra xem đồ thị có phải là Hamilton hay không.

3.8.2.3. Bài toán người du lịch

Bài toán người du lịch còn được gọi là bài toán người bán hàng rong (Travelling Salesman Problem-TSP). Đây là bài toán nổi tiếng được phát biểu bởi nhà toán học William Rowan Hamilton vào thế kỷ 19, thuộc một trong những bài toán nghiên cứu sâu nhất trong tối ưu hóa. Nó thường được dùng làm thước đo cho nhiều phương pháp tối ưu hóa. Nhiều nhà toán học nổi tiếng đã tham gia nghiên cứu bài toán này như William Rowan Hamilton, George Dantzig, Delbert Ray Fulkerson,

... Bài toán này đã được nhà toán học Richard M. Karp chứng minh là NP-đầy đủ vào năm 1972. Điều này chính là một lý giải toán học cho sự khó khăn trong việc tìm kiếm lời giải. Bài toán nổi tiếng này được phát biểu như sau: “Giả sử có n thành phố với chi phí từ thành phố i đến thành phố j là $c_{ij} \geq 0$ ($i, j = 1, 2, \dots, n$). Một người du lịch xuất phát từ một thành phố nào đó muốn tới thăm $n - 1$ thành phố còn lại sao cho mỗi thành phố đến đúng một lần, rồi quay về thành phố ban đầu. Hỏi người du lịch nên đi theo trình tự như thế nào để chi phí là nhỏ nhất (ở đây chi phí giữa hai thành phố có thể hiểu là cự ly thông thường hoặc thời gian cần đi hoặc chi phí của hành trình, ...).

Bài toán người du lịch có thể mô hình hóa bằng đồ thị như sau: xét đồ thị đầy đủ có hướng, có trọng số không âm $G = (V, E, c)$ với $V = \{v_1, v_2, \dots, v_n\}$ tương ứng mỗi thành phố i với đỉnh v_i , và $c(v_i, v_j) = c_{ij}$ với mọi $i, j = 1, 2, \dots, n$. Ở đây có thể $c_{ij} \neq c_{ji}$. Tìm một chu trình Hamilton có trọng số nhỏ nhất trong đồ thị G .

Bài toán này đã có lời giải bằng cách sử dụng *phương pháp nhánh cận*. Ý tưởng của phương pháp này là, giả sử trong một tập hữu hạn các phương án của bài toán, ta phải chọn ra được một phương án tối ưu theo một tiêu chuẩn nào đó (thí dụ làm cho hàm mục tiêu đạt giá trị nhỏ nhất). Ta sẽ tìm cách phân chia tập phương án đang xét thành hai tập con không giao nhau. Với mỗi tập con này, ta sẽ tính “cận dưới” (chặn dưới đủ tốt) của các giá trị hàm mục tiêu ứng với các phương án trong đó. Mang so sánh hai cận dưới vừa tính được, ta có thể phán đoán xem tập con nào có nhiều triển vọng chứa phương án tối ưu và tiếp tục phân chia tập con đó thành hai tập con khác không giao nhau, lại tính các cận dưới tương ứng, ... Lặp lại quá trình này thì sau một số hữu hạn bước, cuối cùng sẽ được một phương án tốt, nói chung là tối ưu. Nếu không thì lặp lại quá trình phân chia để kiểm tra và sau một vài bước, ta sẽ được phương án tối ưu. Người ta thường mô tả quá trình phân chia đó bằng một “cây có gốc” mà gốc sẽ tượng trưng cho tập toàn bộ các phương án, còn các đỉnh ở phía dưới lần lượt tượng trưng cho các tập con trong quá trình “phân nhánh nhị phân”. Vì vậy, phương pháp này mang tên nhánh cận. Rõ ràng, nếu không xác định thành phố xuất phát thì có $n!$ hành trình, trong đó mỗi hành trình ứng với một hoán vị nào đó của tập $\{1, 2, \dots, n\}$. Còn nếu cho trước thành phố xuất phát thì có tất cả là $(n - 1)!$ hành trình. Giả sử $h = (\pi(1), \pi(2), \dots, \pi(n), \pi(1))$ (với π là một hoán vị) là một hành trình qua các thành phố $\pi(1), \pi(2), \dots, \pi(n)$ theo thứ tự đó rồi quay về $\pi(1)$. Khi đó, hàm mục tiêu

$$f(h) = c_{\pi(1)\pi(2)} + c_{\pi(2)\pi(3)} + \dots + c_{\pi(n-1)\pi(n)} + c_{\pi(n)\pi(1)} = \sum_{(i,j) \in h} c_{ij}$$

sẽ biểu thị chi phí của hành trình h , trong đó (i, j) ký hiệu một chặng đường trong hành trình h , tức một cặp thành phố kề nhau trong h .

Ma trận thu được từ ma trận trọng số không âm $C = (c_{ij})$ bằng cách trừ đi các phần tử trong mỗi dòng (tương ứng cột) với phần tử nhỏ nhất tương ứng của dòng (cột) đó được gọi là *ma trận rút gọn* của ma trận C . Quá trình tạo ra ma trận rút gọn được gọi là *rút gọn*. Tổng các phần tử nhỏ nhất đem trừ đi trong quá trình rút gọn được gọi là *hằng số rút gọn*.

Như vậy, ma trận rút gọn cũng không âm và có ít nhất một số 0 trên mỗi dòng và cột. Có thể thấy ma trận rút gọn là không duy nhất.

Ví dụ 3.63. Xét ma trận sau:

$$C = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 6 & 5 \\ 9 & 8 & 7 \end{pmatrix}$$

Quá trình rút gọn ma trận C dẫn đến ma trận rút gọn C^1 là:

$$C = \begin{array}{ccc|c} 1 & 2 & 3 & 1 \\ 4 & 6 & 5 & 4 \\ 9 & 8 & 7 & 7 \end{array} \Rightarrow \begin{array}{ccc|c} 0 & 1 & 2 & \\ 0 & 2 & 1 & \\ 2 & 1 & 0 & \\ \hline 0 & 1 & 0 & \end{array} \Rightarrow C^1 = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 1 & 1 \\ 2 & 0 & 0 \end{pmatrix}$$

Với hằng số rút gọn là $s = 1 + 4 + 7 + 1 = 13$. Chúng ta cũng có thể rút gọn ma trận C theo cách khác:

$$C = \begin{array}{ccc|c} 1 & 2 & 3 & \\ 4 & 6 & 5 & \\ 9 & 8 & 7 & \\ \hline 1 & 2 & 3 & \end{array} \Rightarrow \begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 3 & 4 & 2 & 2 \\ 8 & 6 & 4 & 4 \end{array} \Rightarrow C^1 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 2 & 0 \\ 4 & 2 & 0 \end{pmatrix}$$

Với hằng số rút gọn là $s = 1 + 2 + 3 + 2 + 4 = 12$. Người ta chứng minh được quá trình tính toán trên ma trận ban đầu C có thể thay bằng quá trình tính toán trên ma trận rút gọn để được đơn giản hơn.

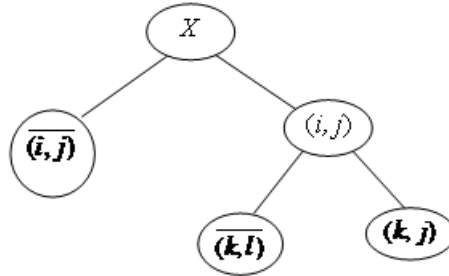
Mệnh đề 3.46. *Phương án tối ưu xét trên ma trận trọng số ban đầu $C = (c_{ij})$ cũng là phương án tối ưu xét trên ma trận rút gọn, và ngược lại.*

Như vậy nếu ký hiệu $f^k(h) = \sum_{(i,j) \in h} c_{ij}^k$ là giá trị hàm mục tiêu ứng với ma trận rút gọn (lần thứ k) $C^k = (c_{ij}^k)$ ($k = 1, 2, \dots$) và s^k là hằng số rút gọn, thì ta có

$$f(h) = f^k(h) + s^k.$$

Rõ ràng $f^k(h) \geq 0$ nên s^k là cận dưới cho $f(h)$.

Quá trình phân nhánh có thể được mô tả qua mô tả hình học của phương pháp nhánh cận như sau, trong đó quá trình phân hoạch tập các chu trình thành các tập con không giao nhau được mô tả hình học bằng một cây với các đỉnh giao bởi các cặp cho phép (i, j) và cặp không cho phép $\overline{(i, j)}$ (tức cấm đi lại giữa thành phố i và thành phố j):



Thuật toán sử dụng các ký hiệu sau đây:

- X : là đỉnh từ đó bắt đầu tiến hành phân nhánh
- Y : là tập chứa các đỉnh (i, j) có triển vọng của nhánh
- $\overline{Y}$: là tập chứa các đỉnh $\overline{(i, j)}$ ứng với cặp thành phố cấm đi lại.

Thuật toán 3.9 *Thuật toán Nhánh cận giải bài toán người du lịch*

Vào: Ma trận trọng số không âm $C = (c_{ij})$ với $i, j = 1, 2, \dots, n$.

Ra: Chu trình Hamilton với chi phí nhỏ nhất.

Phương pháp:

Bước 1. (*Khởi tạo*) Đặt $k = 1$. Tạo ma trận rút gọn C^k từ ma trận trọng số ban đầu C . Tính hằng số rút gọn s^k . Đặt cận dưới của đỉnh ban đầu X là $\omega(X) = s^k$.

Bước 2. (*Lập: xác định cặp thành phố để rẽ nhánh*) Chọn thành phần để đưa vào tập Y . Đó là chọn các cặp (i, j) sao cho $c_{ij}^k = 0$ với $i, j = 1, 2, \dots, n$ và $i \neq j$. Với mỗi thành phần được chọn (i, j) ở trên, tính: $\theta(i, j) = \min_{j' \neq j} (c_{ij'}^k) + \min_{i' \neq i} (c_{i'j}^k)$.

Xác định $\theta(x, y) = \max(\theta(i, j))$. Suy ra $(x, y) \in Y$ và $(x, y) \in \bar{Y}$.

Bước 3. (*Tính cận dưới cho $\bar{Y}$*) Đặt cận dưới cho $\bar{Y}$ là: $\omega(\bar{Y}) = \omega(X) + \theta(x, y)$.

Bước 4. (*Xác định ma trận cắt*) Loại hàng x , cột y (vì mỗi thành phố x chỉ đi vào một lần, thành phố y đi qua một lần). Sau đó đặt $c_{yx} = \infty$ (cấm đi từ thành phố y đến thành phố x để tránh tạo thành chu trình con). Ma trận thu được ở bước này được gọi là *ma trận cắt*.

Bước 5. (*Tính cận dưới cho Y , và kiểm tra*)

$$\omega(Y) = \begin{cases} \omega(X) & \text{nếu ma trận cắt là ma trận rút gọn} \\ \omega(X) + s^{k+1} & \text{ngược lại, với } s^{k+1} \text{ là hằng số rút gọn.} \end{cases}$$

Trường hợp nếu ma trận rút gọn có kích thước 2×2 thì sang Bước 6, còn không $k = k + 1$ và quay lại Bước 2.

Bước 6. (*Kiểm tra điều kiện tối ưu*) Nếu đánh giá của chu trình kín không lớn hơn các đánh giá của các tập để phân nhánh thì chu trình thu được là tối ưu, còn ngược lại nếu tồn tại ít nhất một tập với đánh giá nhỏ hơn thì tiếp tục phân nhánh từ tập với đánh giá nhỏ nhất đó.

Lưu ý 3.8. (1) (Ngăn chặn hành trình con) Một đường đi hoặc chu trình Hamilton không thể chứa chu trình con với số cạnh tạo thành nhỏ hơn n . Do đó ta sẽ đặt $c_{ii} = \infty$ với mọi $i = 1, 2, \dots, n$ để tránh các khuyên. Trường hợp nếu cặp (i, j) được chọn thì ta đặt $c_{ji}^k = \infty$ trong ma trận rút gọn (với $i \neq j$). Còn nếu đã chọn (i, j) , (j, k) và $n > 3$ thì ta đặt $c_{ji}^k = c_{kj}^k = c_{ki}^k = \infty$. Ở đây việc đặt $c_{ij}^k = \infty$ tương đương với việc xóa ô (i, j) trong bảng hoặc xem (i, j) là ô cấm, nghĩa là hai thành phố i và j không được kề nhau trong hành trình định kiến thiết. Ở mỗi giai đoạn của quá trình đều phải tiến hành thủ tục ngăn chặn này trước khi tiếp tục rút gọn ma trận.

(2) Cận dưới của đỉnh cuối cùng trong cây phân nhánh chính là độ dài của hành trình vừa kiến thiết. Nếu cận dưới của đỉnh này không lớn hơn các cận dưới của mọi đỉnh treo trên cây phân nhánh thì hành trình đó là tối ưu. Còn ngược lại thì phải xuất phát từ đỉnh treo nào có cận dưới nhỏ hơn để phân nhánh tiếp tục và kiểm tra xem điều kiện ở trên có thỏa mãn không.

Ví dụ 3.64. Giải bài toán người du lịch với ma trận trọng số cho bởi:

$$C = \begin{pmatrix} \infty & 27 & 43 & 16 & 30 & 26 \\ 7 & \infty & 14 & 1 & 30 & 25 \\ 20 & 13 & \infty & 35 & 5 & 0 \\ 21 & 16 & 25 & \infty & 18 & 28 \\ 12 & 46 & 27 & 48 & \infty & 5 \\ 23 & 5 & 5 & 9 & 5 & \infty \end{pmatrix}$$

Lần 1:

- Rút gọn:

$$C^1 = \begin{array}{c|cccccc|c} & 1 & 2 & 3 & 4 & 5 & 6 & \\ \hline 1 & \infty & 11 & 27 & 0 & 14 & 10 & 16 \\ 2 & 1 & \infty & 13 & 0 & 29 & 24 & 1 \\ 3 & 15 & 13 & \infty & 35 & 5 & 0 & 0 \\ 4 & 0 & 0 & 9 & \infty & 2 & 12 & 16 \\ 5 & 2 & 41 & 22 & 43 & \infty & 0 & 5 \\ 6 & 13 & 0 & 0 & 4 & 0 & \infty & 5 \\ \hline & 5 & 0 & 0 & 0 & 0 & 0 & \end{array}$$

Với hằng số rút gọn $s^1 = 16 + 1 + 16 + 5 + 5 + 5 = 48$. Đặt $w(X) = 48$.

- Chọn cặp (i, j) để phân nhánh: $c_{14}^1 = c_{24}^1 = c_{36}^1 = c_{41}^1 = c_{42}^1 = c_{56}^1 = c_{62}^1 = c_{63}^1 = c_{65}^1 = 0$. Khi đó tính $\theta(1, 4) = 10, \theta(2, 4) = 1, \theta(3, 6) = 5, \theta(4, 1) = 1, \theta(4, 2) = 0, \theta(5, 6) = 2, \theta(6, 2) = 0, \theta(6, 3) = 9, \theta(6, 5) = 2$. Suy ra $\text{Max}(\theta(i, j)) = \theta(1, 4) = 10$. Do đó chọn $(1, 4)$ là điểm phân nhánh.
- Tính $\omega(\overline{1, 4}) = 48 + 10 = 58$.
- Loại hàng 1, cột 4. Đặt $c_{41}^1 = \infty$. Ma trận thu được là:

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 5 & 6 \\ \hline 2 & 1 & \infty & 13 & 29 & 24 \\ 3 & 15 & 13 & \infty & 5 & 0 \\ 4 & \infty & 0 & 9 & 2 & 12 \\ 5 & 2 & 41 & 22 & \infty & 0 \\ 6 & 13 & 0 & 0 & 0 & \infty \end{array}$$

Ma trận cắt này có kích thước 5×5 , và không phải ma trận rút gọn.

- Rút gọn ma trận cắt:

$$C^2 = \begin{array}{c|ccccc|c} & 1 & 2 & 3 & 5 & 6 & \\ \hline 2 & 0 & \infty & 12 & 28 & 23 & 1 \\ 3 & 15 & 13 & \infty & 5 & 0 & 0 \\ 4 & \infty & 0 & 9 & 2 & 12 & 0 \\ 5 & 2 & 41 & 22 & \infty & 0 & 0 \\ 6 & 13 & 0 & 0 & 0 & \infty & 0 \end{array}$$

Với hằng số rút gọn $s^2 = 1$. Suy ra $\omega(1, 4) = 48 + 1 = 49$.

Lần 2:

- Chọn cặp để phân nhánh: $c_{21}^2 = c_{36}^2 = c_{42}^2 = c_{56}^2 = c_{62}^2 = c_{63}^2 = c_{65}^2 = 0$ và $\theta(2, 1) = 14, \theta(3, 6) = 5, \theta(4, 2) = 2, \theta(5, 6) = 2, \theta(6, 2) = 0, \theta(6, 3) = 9, \theta(6, 5) = 2$. Suy ra $\text{Max}(\theta(i, j)) = \theta(2, 1) = 14$. Do đó chọn $(2, 1)$ làm điểm phân nhánh.
- Tính $\omega(\overline{2, 1}) = 49 + 14 = 63$.
- Loại hàng 2, cột 1. Đặt $c_{42}^2 = \infty$. Ma trận thu được là:

$$\begin{array}{c|cccc} & 2 & 3 & 5 & 6 \\ \hline 3 & 13 & \infty & 5 & 0 \\ 4 & \infty & 9 & 2 & 12 \\ 5 & 41 & 22 & \infty & 0 \\ 6 & 0 & 0 & 0 & \infty \end{array}$$

Ma trận cắt này có kích thước 4×4 , không phải ma trận rút gọn.

- Rút gọn ma trận cắt:

$$C^3 = \begin{array}{c|cccc|c} & 2 & 3 & 5 & 6 & \\ \hline 3 & 13 & \infty & 5 & 0 & 0 \\ 4 & \infty & 7 & 0 & 10 & 2 \\ 5 & 41 & 22 & \infty & 0 & 0 \\ 6 & 0 & 0 & 0 & \infty & 0 \end{array}$$

Hằng số rút gọn: $s^3 = 2$. Suy ra $\omega(2, 1) = 49 + 2 = 51$.

Lần 3:

- Chọn cặp để phân nhánh: $c_{36}^3 = c_{45}^3 = c_{56}^3 = c_{62}^3 = c_{63}^3 = c_{65}^3 = 0$ và $\theta(3, 6) = 5, \theta(4, 5) = 0, \theta(5, 6) = 22, \theta(6, 2) = 13, \theta(6, 3) = 7, \theta(6, 5) = 0$. Suy ra $\text{Max}(\theta(i, j)) = \theta(5, 6) = 22$. Do đó chọn $(5, 6)$ làm điểm phân nhánh.
- Tính $\omega(\overline{5, 6}) = 51 + 22 = 73$.
- Loại hàng 5, cột 6. Đặt $c_{65}^3 = \infty$. Ma trận thu được là:

	2	3	5
3	13	∞	5
4	∞	7	0
6	0	0	∞

Ma trận cắt này có kích thước 3×3 , cũng không phải ma trận rút gọn.

- Rút gọn ma trận cắt:

$$C^4 = \begin{array}{c|ccc|c} & 2 & 3 & 5 & \\ \hline 3 & 8 & \infty & 0 & 5 \\ 4 & \infty & 7 & 0 & 0 \\ 6 & 0 & 0 & \infty & 0 \end{array}$$

Hằng số rút gọn: $s^4 = 5$. Suy ra $\omega(5, 6) = 51 + 5 = 56$.

Lần 4:

- Chọn cặp để phân nhánh: $c_{35}^4 = c_{45}^4 = c_{62}^4 = c_{63}^4 = 0$ và $\theta(3, 5) = 8, \theta(4, 5) = 7, \theta(6, 2) = 8, \theta(6, 3) = 7$. Suy ra $\max(\theta(i, j)) = \theta(3, 5) = 8$. Do đó chọn $(3, 5)$ làm điểm phân nhánh.
- Tính $\omega(\overline{3, 5}) = 56 + 8 = 64$.
- Loại hàng 3, cột 5. Đặt $c_{63}^4 = \infty$. Ma trận thu được là:

	2	3
4	∞	7
6	0	∞

Ma trận cắt này có kích thước 2×2 , cũng không phải ma trận rút gọn.

- Rút gọn ma trận cắt:

$$\begin{array}{c|cc|c} & 2 & 3 & \\ \hline 4 & \infty & 7 & 7 \\ 6 & 0 & \infty & 0 \end{array} \Rightarrow C^5 = \begin{array}{c|cc|c} & 2 & 3 & \\ \hline 4 & \infty & 0 & \\ 6 & 0 & \infty & \end{array}$$

Hằng số rút gọn: $s^5 = 7$. Suy ra $\omega(3, 5) = 56 + 7 = 63$. Ma trận rút gọn có kích thước 2×2 , thuật toán dừng. Như vậy cuối cùng ta thu được 6 cặp thành phố được chọn trong hành trình là:

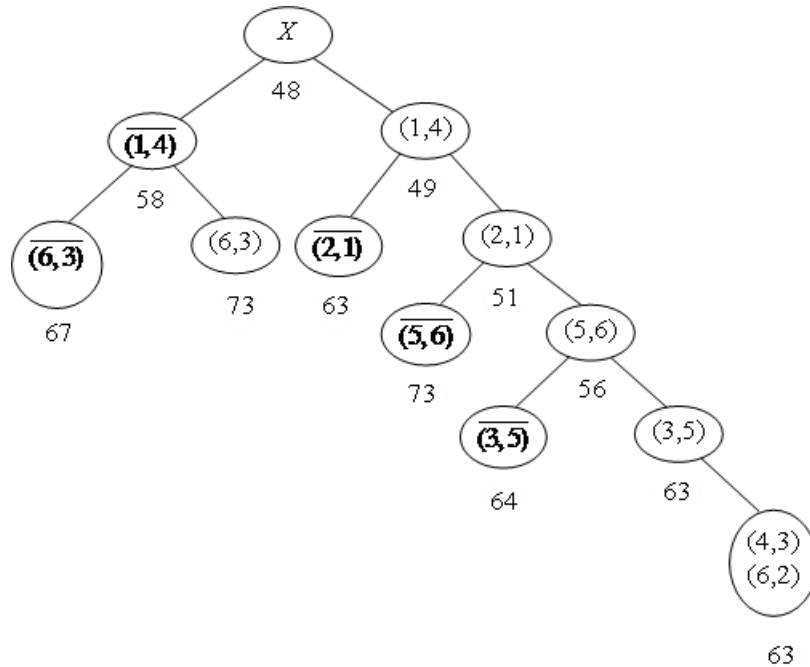
$$(1, 4), (2, 1), (5, 6), (3, 5), (4, 3), (6, 2).$$

Do đó hành trình thu được sẽ là $h_0 = (1, 4, 3, 5, 6, 2, 1)$ với $f(h_0) = 63$ là cận dưới của đỉnh cuối cùng.

Tuy nhiên cận dưới của đỉnh cuối cùng là 63, trong khi đỉnh treo $\overline{(1, 4)}$ có cận dưới là 58 (< 63), nên phải tiếp tục phân nhánh tại điểm $\overline{(1, 4)}$ để kiểm tra. Tuy vậy sau sự phân nhánh tại điểm này thì mọi đỉnh treo đều có cận dưới không nhỏ hơn 63, nên có thể khẳng định rằng hành trình $h_0 = (1, 4, 3, 5, 6, 2, 1)$ là tối ưu.

Ở đây sự phân nhánh tại điểm $\overline{(1, 4)}$ được thực hiện như sau: trong ma trận rút gọn C^1 , ta đặt $c_{14}^1 = \infty$ vì xem $(1, 4)$ là ô cấm. Khi đó không còn $\theta(1, 4)$ nữa, suy ra $\text{Max}(\theta(i, j)) = \theta(6, 3) = 9$. Lúc này chọn $(6, 3)$ làm điểm phân nhánh. Nên $\omega(\overline{6, 3}) = 58 + 9 = 67$. Đặt $c_{36}^1 = \infty$. Rút gọn ma trận với hằng số rút gọn là 15. Suy ra $\omega(6, 3) = 58 + 15 = 73$.

Tóm lại cây lời giải của bài toán là:



Hình 3.6: Cây lời giải của bài toán người du lịch

3.9. Luồng cực đại

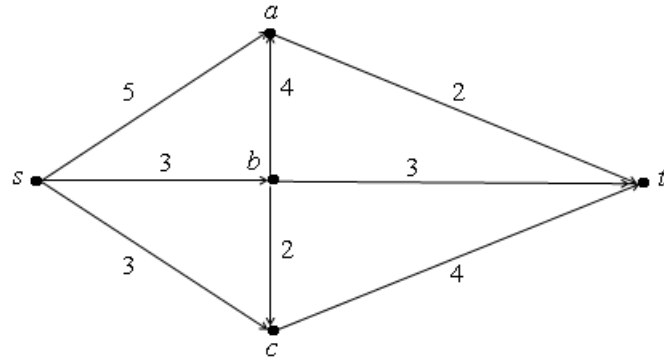
Một trong số những bài toán tối ưu trên đồ thị có nhiều ứng dụng trong thực tiễn cũng như những ứng dụng thú vị trong tối ưu tổ hợp là bài toán luồng cực đại. Bài toán này được đề xuất vào đầu những năm 1950, gắn liền với tên tuổi của hai nhà toán học Mỹ là Ford và Fulkerson.

3.9.1. Mạng vận tải, luồng trong mạng vận tải

Mạng vận tải (Network) là một đồ thị có hướng, có trọng số, không có khuyên $G = (V, E, c)$ trong đó có duy nhất một đỉnh s không có cung đi vào gọi là *đỉnh phát* (source), duy nhất một đỉnh t không có cung đi ra gọi là *đỉnh thu* (sink) và hàm $c : E \rightarrow \mathbb{R}^+$ gán mỗi cung $e = (u, v) \in E$ bởi một số thực không âm $c(e) = c(u, v)$ gọi là *khả năng thông qua* (capacity) của cung e . Trường hợp $e \notin E$, thì để đơn giản người ta quy ước $c(e) = 0$.

Mạng vận tải như thế này thường được ký hiệu lại là $G = (V, E, c, s, t)$.

Ví dụ 3.65. Sau đây là ví dụ đơn giản về một mạng vận tải:



với đỉnh phát là s và đỉnh thu là t .

Cho mạng vận tải $G = (V, E, c, s, t)$. *Luồng*¹¹ (hay *lưu lượng-Flow*) trong G là một hàm số thực $f : E \rightarrow \mathbb{R}^+$ thỏa các điều kiện sau:

(i) (*Ràng buộc khả năng thông qua-Capacity constraint*)

$$\forall e \in E : 0 \leq f(e) \leq c(e)$$

(ii) (*Bảo toàn luồng-Flow conservation*)

$$\forall v \in V \setminus \{s, t\} : \sum_{w \in \Gamma^+(v)} f(v, w) = \sum_{u \in \Gamma^-(v)} f(u, v),$$

trong đó $\Gamma^-(v)$ là ký hiệu tập các đỉnh của G có cung đến đỉnh v và $\Gamma^+(v)$ là ký hiệu tập các đỉnh của G có cung từ đỉnh v đến nó:

$$\Gamma^+(v) = \{w \in V | (v, w) \in E\},$$

$$\Gamma^-(v) = \{u \in V | (u, v) \in E\}.$$

Giá trị $f(e)$ được gọi là *luồng trên cung e* . Để đơn giản, ta xem $f(e) = 0$ nếu cung $e \notin E$.

¹¹Còn được gọi là *dòng chảy*.

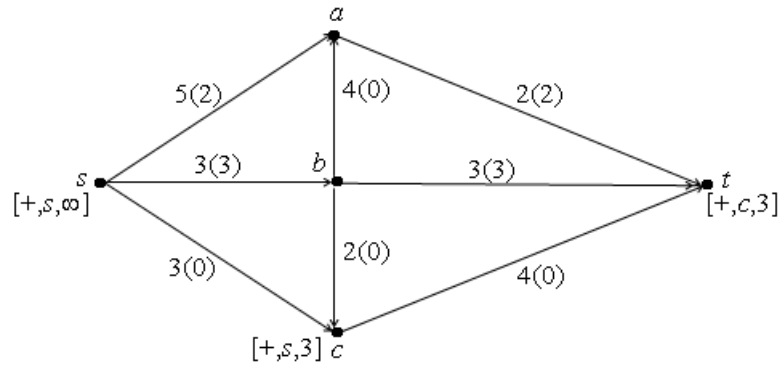
Như vậy, điều kiện (i) có nghĩa luồng trên mỗi cung không vượt quá khả năng thông qua của nó, điều kiện (ii) có nghĩa tổng luồng trên các cung đi vào một đỉnh bằng tổng luồng trên các cung đi ra khỏi đỉnh đó (không tính đỉnh phát và đỉnh thu). Điều kiện (ii) này còn được gọi là *điều kiện cân bằng luồng trên mỗi đỉnh*.

Rõ ràng, một mạng vận tải có thể có nhiều luồng f , và tổng luồng trên các cung đi ra từ điểm phát s bằng tổng luồng trên các cung đi vào điểm thu t , và tổng luồng này được gọi là *giá trị của luồng f* , thường ký hiệu $value(f)$ hoặc $f(G)$.

Tức là

$$value(f) = \sum_{v \in \Gamma^+(s)} f(s, v) = \sum_{u \in \Gamma^-(t)} f(u, t).$$

Ví dụ 3.66. Dưới đây là một luồng f trên mạng vận tải G ở Ví dụ 3.65, trong đó các số ghi trên mỗi cung có ý nghĩa: “khả năng thông qua (giá trị luồng)”.



Luồng f trên mạng vận tải G có giá trị là $value(f) = 2 + 3 = 5$.

Luồng trên cung $e \in E$ được gọi là *bão hòa* nếu $f(e) = c(e)$ và *chưa bão hòa* nếu $f(e) < c(e)$. Luồng có giá trị lớn nhất (maximum flow) trong một mạng vận tải được gọi là *luồng cực đại*.

Bài toán quan trọng được đặt ra là thiết kết mạng vận tải sao cho luồng qua mạng vận tải từ đỉnh phát đến đỉnh thu đạt được giá trị lớn nhất. Như đã nói ở trên, đây là bài toán tối ưu trên đồ thị có rất nhiều ứng dụng ý nghĩa trong thực tiễn cũng như trong lý thuyết tổ hợp. Một số các ứng dụng thực tiễn của bài toán này có thể kể đến như xác định cường độ lớn nhất cho phép trong một mạng điện, lượng dòng chảy lớn nhất cho phép trong một mạng sông ngoài. Bài toán này được phát biểu một cách tường minh như sau.

Bài toán 3.1 (*Bài toán luồng cực đại*). Cho mạng vận tải $G = (V, E, c, s, t)$. Hãy tìm luồng f trong G sao cho giá trị luồng $value(f)$ là lớn nhất.

Sau đây là một số bài toán luồng cực đại quan trọng trong thực tiễn.

Ví dụ 3.67 (*Mạng giao thông*). Xét một mạng giao thông $G = (V, E, c, s, t)$, trong đó V là các nút giao thông (ngã ba, ngã tư, ...), s là nút vào, t là nút ra, đoạn đường giữa hai nút gần nhất được gọi là cung của G và số lượng xe tối đa đi được trên đoạn đường này là khả năng thông qua của cung. Hãy tìm số lượng tối đa các xe đi từ nút vào s qua mạng đến nút ra t sao cho không gây tắc nghẽn giao thông. Đây chính là bài toán tìm luồng cực đại trong mạng vận tải giao thông đang xét.

Ví dụ 3.68 (*Đường ống dẫn dầu*). Xét một mạng đường ống dẫn dầu $G = (V, E, c, s, t)$, trong đó V là tập các nút giao nhau của các đoạn đường ống, s là nơi cung cấp dầu, t là nơi tiếp nhận dầu, lượng dầu tối đa qua các đoạn đường ống trong một đơn vị thời gian được gọi là khả năng thông qua của cung. Hãy tìm lượng dầu tối đa có thể bơm trong một đơn vị thời gian để đưa dầu đến nơi tiếp nhận t sao cho dầu không bị tràn ra khỏi nơi cung cấp dầu s .

Đây cũng chính là bài toán tìm luồng cực đại trong mạng vận tải đường ống dẫn dầu đang xét.

Để giải quyết bài toán trên chúng ta cần đến khái niệm lát cắt và khả năng thông qua của lát cắt trong một mạng vận tải.

3.9.2. Định lý Ford-Fulkerson

Cho mạng vận tải $G = (V, E)$. Một lát cắt (Cut) của G là một phân hoạch (X, Y) của V sao cho $s \in X$ và $t \in Y$.

Tổng trọng số của tất cả các cung đi từ một đỉnh của X đến một đỉnh của Y được gọi là khả năng thông qua của lát cắt (X, Y) , ký hiệu $c(X, Y)$.

Một cách hình thức:

$$c(X, Y) = \sum_{u \in X, v \in Y} c(u, v).$$

Lát cắt có khả năng thông qua nhỏ nhất được gọi là lát cắt nhỏ nhất.

Chẳng hạn xét mạng tải G ở Ví dụ 3.65 với lát cắt (X, Y) là $X = \{1, 2, 3, 6\}$ và $Y = \{4, 5\}$ thì

$$c(X, Y) = c(2, 5) + c(3, 4) + c(6, 5) = 3 + 6 + 6 = 15.$$

Bổ đề 3.4. Giá trị của luồng cực đại f trong mạng vận tải $G = (V, E)$ không lớn hơn khả năng thông qua của lát cắt nhỏ nhất trong G .

Dựa vào Bổ đề 3.4 chúng ta có thể chứng minh được rằng luồng cực đại và lát cắt nhỏ nhất trong một mạng vận tải (hữu hạn) $G = (V, E)$ bất kỳ luôn tồn tại.

Tuy vậy năm 1956 Ford và Fulkerson đã chứng minh được một khẳng định mạnh hơn khẳng định nhận được trong Bổ đề 3.4. Đó là kết quả sau đây.

Định lý 3.47 (Ford & Fulkerson, 1956). *Giá trị luồng cực đại trong một mạng vận tải bằng khả năng thông qua của lát cắt nhỏ nhất trong mạng vận tải đó.*

Nếu khả năng thông qua của một số cung là vô cùng nhưng giá trị của luồng cực đại trong mạng vận tải đó vẫn là hữu hạn, thì Định lý 3.47 vẫn còn đúng.

Mạng vận tải $G = (V, E)$ được gọi là *mạng vận tải nguyên* nếu khả năng thông qua của mỗi cung trong G là một số nguyên không âm. Một luồng f trong G được gọi là *luồng nguyên* nếu mọi cung e trong G có giá trị $f(e)$ là một số nguyên không âm. Trường hợp $f(e) = 0$ với mọi $e \in E$ thì ta nói f là một *luồng không*.

Định lý 3.48. *Nếu mạng vận tải $G = (V, E)$ là một mạng nguyên thì luồng cực đại trong G cũng là một luồng nguyên.*

Hai Định lý 3.47 và 3.48 trên được sử dụng trong việc ứng dụng luồng cực đại vào giải rất nhiều bài toán trong tối ưu tổ hợp.

3.9.3. Thuật toán Ford-Fulkerson tìm luồng cực đại trong mạng vận tải

Thuật toán bắt đầu từ một luồng nguyên bất kỳ nào đó, chẳng hạn luồng không. Sau đó tìm các đường có hướng từ đỉnh phát s đến đỉnh thu t để tăng luồng. Để làm điều này Ford và Fulkerson đưa ra phương pháp đánh dấu các đỉnh bằng nhãn. Nhãn của các đỉnh sẽ chỉ cho ta theo các cung nào để có thể tăng luồng và tăng được bao nhiêu. Mỗi khi tìm được đường có hướng để tăng luồng ta lại tăng luồng theo đường đó. Sau đó xóa tất cả các nhãn và sử dụng luồng mới nhận được để đánh dấu lại các đỉnh. Thuật toán sẽ kết thúc khi nào không tìm được đường có hướng nào nữa để tăng luồng. Trong thuật toán này *nhãn của đỉnh* $v \in V$, ký hiệu $\bar{v}$, được định nghĩa là bộ 3 $\bar{v} = [\alpha(v), \beta(v), \sigma(v)]$, với $\alpha(v) \in \{+, -\}$, $\beta(v) \in V$ và $\sigma(v) \in \mathbb{N}$. Lưu ý mỗi đỉnh của G trong quá trình thực hiện thuật toán chỉ có thể ở một trong ba trạng thái: *chưa có nhãn*, *có nhãn nhưng chưa xét* và *có nhãn đã xét xong*.

Sau đây là thuật toán tìm luồng cực đại trong một mạng vận tải nguyên. Thuật toán bao gồm hai bước chính: bước gán nhãn cho các đỉnh và bước tăng luồng. Hai bước đó thực hiện luân phiên nhau cho tới khi tìm được luồng cực đại trong mạng vận tải nguyên.

Thuật toán 3.10 *Tìm luồng cực đại trong mạng vận tải nguyên*

[2, 8]

Vào: mạng vận tải nguyên $G = (V, E)$.

Ra: một luồng cực đại của G .

Phương pháp:

Bước 1. (*Khởi tạo*) Gán luồng không cho G .

Bước 2. (*Gán nhãn cho các đỉnh*) Bao gồm 3 bước:

Bước 2.1. Gán nhãn cho đỉnh phát s : $\bar{s} = [+ , s, \infty]$.

Lúc này đỉnh s được coi là ở trạng thái đã có nhãn nhưng chưa xét. Các đỉnh còn lại được coi là ở trạng thái chưa có nhãn.

Bước 2.2. (*Xét nhãn*) Chọn một đỉnh đã có nhãn nhưng chưa được xét ra xét, giả sử đỉnh v với nhãn $\bar{v} = [\alpha(v), \beta(v), \sigma(v)]$. Quá trình xét là quá trình gán nhãn cho một số đỉnh w như ở (a) hoặc (b) như sau:

(a) Nếu $w \in \{x \in \Gamma^+(v) : f(v, x) < c(v, x)\}$ và w chưa có nhãn, thì nhãn là $\bar{w} = [+ , v, \sigma(w)]$ với $\sigma(w) = \min\{\sigma(v), c(v, w) - f(v, w)\}$.

(b) Nếu $w \in \{x \in \Gamma^-(v) : f(x, v) > 0\}$ và w chưa có nhãn, thì nhãn là $\bar{w} = [- , v, \sigma(w)]$ với $\sigma(w) = \min\{\sigma(v), f(w, v)\}$.

Sau khi thực hiện xong Bước (a) hoặc Bước (b), đỉnh v được coi là ở trạng thái có nhãn và đã xét xong. Bằng cách nào đó hãy đánh dấu đỉnh v để ta có thể nhận biết v thuộc trạng thái nói trên. Các đỉnh w vừa được gán nhãn ở Bước (a) hoặc Bước (b) sẽ được coi là trạng thái đã có nhãn nhưng chưa xét. Còn các đỉnh còn lại thì có trạng thái như nó đã có trước đây.

Bước 2.3. Xét 3 trường hợp:

(a) Nếu đỉnh thu t chưa có nhãn và vẫn còn những đỉnh đã có nhãn nhưng chưa xét, thì quay lại Bước 2.2.

(b) Nếu đỉnh thu t chưa có nhãn và không còn đỉnh nào đã có nhãn nhưng chưa xét, thì kết thúc thuật toán. Luồng cực đại của G chính là luồng hiện tại (luồng sau cùng).

(c) Nếu đỉnh thu t được gán nhãn $\bar{t} = [\alpha(t), \beta(t), \sigma(t)]$, thì sang Bước 3.

Bước 3. (*Tăng luồng*)

Bước 3.1. Xuất phát từ đỉnh thu $u = t$.

Bước 3.2. (*Tăng/giảm luồng*) Xét hai trường hợp:

(a) Nếu nhãn $\bar{u} = [+ , \beta(u), \sigma(u)]$, thì ta tăng luồng qua cung $(\beta(u), u)$ từ $f((\beta(u), u))$ lên $f((\beta(u), u)) + \sigma(u)$.

(b) Nếu nhãn $\bar{u} = [- , \beta(u), \sigma(u)]$, thì ta giảm luồng qua cung $(u, \beta(u))$ từ $f((\beta(u), u))$ xuống $f((\beta(u), u)) - \sigma(u)$.

Bước 3.3. Xét hai trường hợp:

(a) Nếu $\beta(u) \neq s$, thì ta gán $u = \beta(u)$ và quay lại Bước 3.2.

(b) Nếu $\beta(u) = s$, thì xóa tất cả nhãn của các đỉnh và quay lại Bước 2 để gán lại nhãn cho các đỉnh với luồng mới vừa được điều chỉnh.

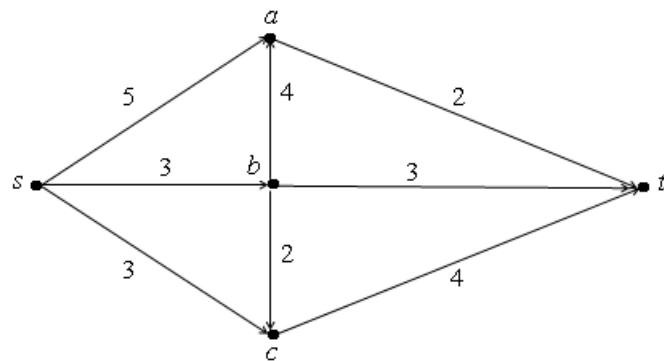
Thuật toán 3.10 có độ phức tạp là $O(n^3)$ với n là số đỉnh của mạng vận tải. Do đó nó là thuật toán tương đối hiệu quả. Ngoài ra khi thuật toán kết thúc ở Bước 2.3(b) thì cặp (X, Y) với X là tập các đỉnh có nhãn ở lần gán nhãn cuối cùng và $Y = V \setminus X$, chính là lát cắt nhỏ nhất của mạng vận tải.

Lưu ý 3.9. Trong thực tế phần lớn các mạng vận tải có khả năng thông qua của

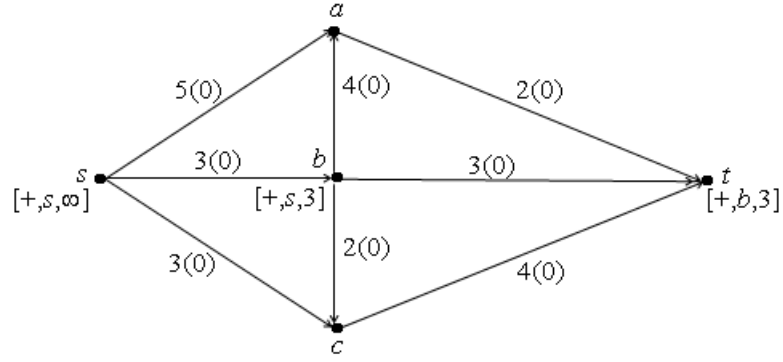
các cung là các số hữu tỷ không âm. Khi đó bằng cách quy đồng mẫu số chúng ta có thể dẫn bài toán tìm luồng cực đại trong mạng vận tải với khả năng thông qua của các cung là hữu tỷ không âm về bài toán tương tự cho mạng vận tải nguyên. Như vậy Thuật toán 3.10 có thể sử dụng cho phần lớn các bài toán thực tế. Trường hợp nếu các khả năng thông qua là các số nguyên không âm rất lớn thì giá trị của luồng cực đại cũng có thể rất lớn và khi đó Thuật toán 3.10 đòi hỏi thực hiện rất nhiều bước để tăng luồng. Còn trường hợp nếu các khả năng thông qua là các số vô tỷ thì người ta đã xây dựng được ví dụ để Thuật toán 3.10 không dừng.

Để thuật toán làm việc hiệu quả, chúng ta cần hết sức cẩn thận khi lựa chọn đường để tăng luồng. Edmonds và Karp chỉ ra rằng nếu đường để tăng luồng được chọn là đường ngắn nhất từ đỉnh s đến đỉnh t (sử dụng tìm kiếm theo chiều rộng) thì thuật toán sẽ kết thúc sau không quá $\frac{mn}{2}$ lần sử dụng đường tăng luồng (m và n tương ứng là số cung và số đỉnh của mạng vận tải). Như vậy với độ phức tạp của tìm kiếm theo chiều rộng trên đồ thị là $O(n + m)$, thì độ phức tạp của thuật toán sẽ là $O(nm^2)$. Nhờ cách tổ chức tìm đường để tăng luồng tốt, người ta đã xây dựng được các thuật toán với độ phức tạp tốt hơn như: $O(n^2m)$ (Dinic, 1970), $O(n^3)$ (Karzanov, 1974), $O(n^2m^{1/2})$ (Cherkasky, 1977) và $O(nm \log n)$ (Sleator-Tarjan, 1980).

Ví dụ 3.69. Xét mạng vận tải $G = (V, E, s, t, c)$ như sau:

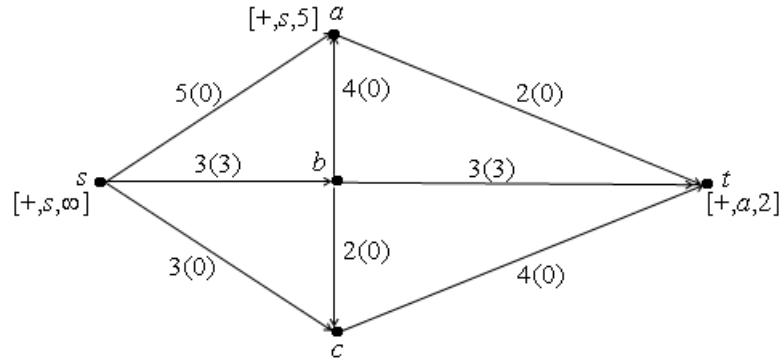


Đầu tiên khởi tạo luồng không, sau đó gán nhãn cho các đỉnh lần thứ nhất ta được mạng vận tải sau:



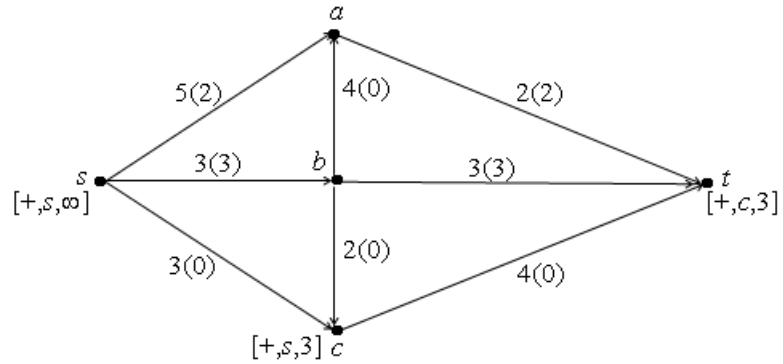
Đường tăng luồng lần 1 thu được: $P_1 = [s, b, t]$

Tăng luồng lần 1 dọc theo P_1 và gán nhãn lần 2:



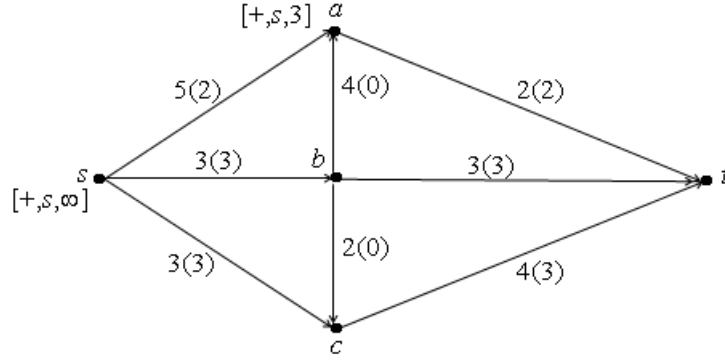
Đường tăng luồng lần 2: $P_2 = [s, a, t]$

Tăng luồng lần 2 dọc theo P_2 và gán nhãn lần 3:



Đường tăng luồng lần 3: $P_3 = [s, c, t]$

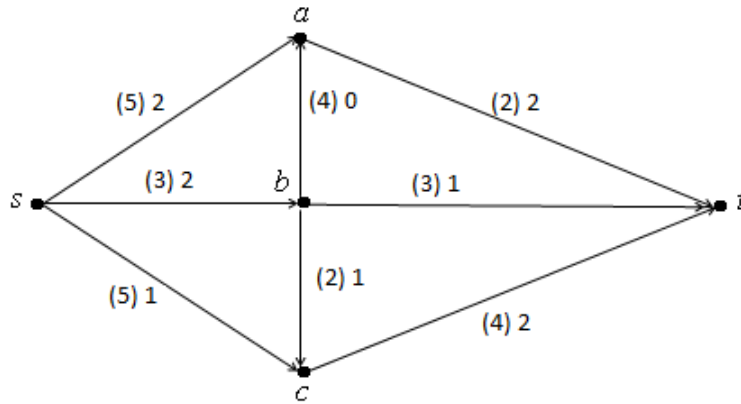
Tăng luồng lần 3 dọc theo P_3 và gán nhãn lần 4:



Ở lần gán nhãn thứ 4, chúng ta không thể gán nhãn được cho đỉnh thu t , trong khi đó tất cả các đỉnh có nhãn đều đã ở trạng thái đã có nhãn và xét xong. Thuật toán kết thúc, như vậy luồng hiện tại là luồng cực đại f cần tìm với giá trị của luồng là $value(f) = 2 + 3 + 3 = 8$.

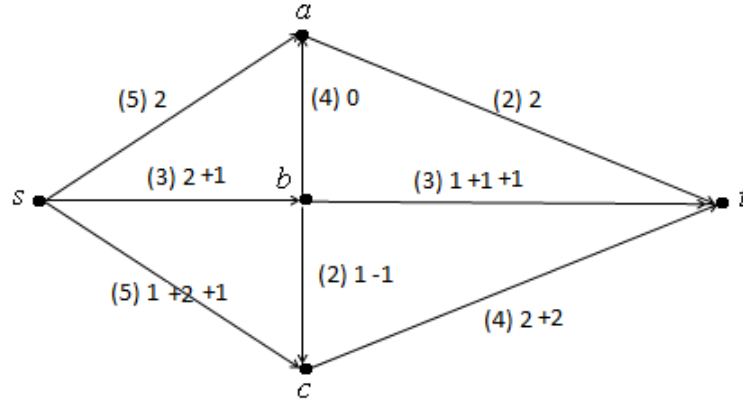
Lưu ý, lát cắt hẹp nhất của G ở trên là (X, Y) với $X = \{s, a\}$ và $Y = \{b, c, t\}$. Khả năng thông qua của lát cắt này là $c(X, Y) = 2 + 3 + 3 = 8 = value(f)$.

Ví dụ 3.70. Tìm luồng cực đại của một mạng vận tải G với một luồng nguyên cho trước trên G như sau:



Gán nhãn lần 1, 2 và 3 ta thu được mạng vận tải cùng với các đường tăng luồng tương ứng sau:

- Đường tăng luồng lần 1: $P_1 = [s, b, t]$ với giá trị tăng là 1 dọc theo đường P_1
- Đường tăng luồng lần 2: $P_2 = [s, c, t]$ với giá trị tăng là 2 dọc theo đường P_2
- Đường tăng luồng lần 3: $P_3 = [s, c, b, t]$ với giá trị tăng là 1 dọc theo đường P_3



Ở lần gán nhãn thứ 4, chúng ta không thể gán nhãn được cho đỉnh thu t , trong khi đó tất cả các đỉnh có nhãn đều đã ở trạng thái đã có nhãn và xét xong. Thuật toán kết thúc, như vậy luồng hiện tại là luồng cực đại f cần tìm với giá trị của luồng là $value(f) = 9$.

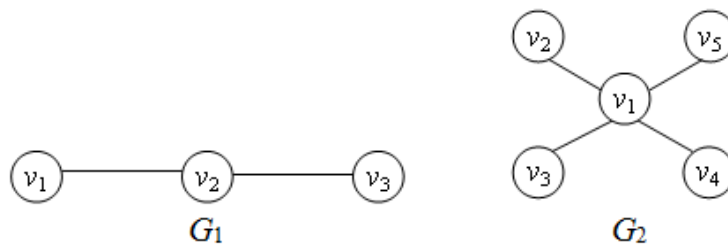
3.10. Cây

3.10.1. Định nghĩa

Cây (Tree) là một đơn đồ thị vô hướng, liên thông không chứa chu trình và có ít nhất một đỉnh. Trường hợp một đơn đồ thị vô hướng không chứa chu trình thì được gọi là một *rừng* (Forest).

Như vậy rừng là một đồ thị mà mỗi thành phần liên thông của nó là một cây.

Ví dụ 3.71. Mỗi đồ thị sau là một cây:



3.10.2. Một số tính chất cơ bản

Mệnh đề 3.49. *Một cây bất kỳ có ít nhất hai đỉnh thì có ít nhất hai đỉnh treo.*

Chứng minh. Xét một cây bất kỳ T có ít nhất hai đỉnh và (a, b) là một cạnh tùy ý của T . Giả sử u và v tương ứng là đỉnh đầu và đỉnh cuối của đường đi sơ cấp

dài nhất chứa cạnh (a, b) trong cây T . Vì T là một cây nên $u \neq v$. Hơn nữa u và v phải là hai đỉnh treo, vì nếu không chẳng hạn u không phải đỉnh treo thì u phải là đầu mút của một cạnh (u, x) với x là đỉnh không thuộc đường đi từ u đến v . Suy ra đường đi sơ cấp từ x đến v chứa cạnh (a, b) dài hơn đường đi từ u đến v . Điều này mâu thuẫn với giả sử trên. Vậy trong cây T phải có tối thiểu hai đỉnh treo. $\square$

Định lý 3.50. Cho T là một đơn đồ thị vô hướng có n đỉnh. Khi đó các tính chất sau là tương đương:

- (1) T là một cây.
- (2) T liên thông và có $n - 1$ cạnh.
- (3) T không chứa chu trình và có $n - 1$ cạnh.
- (4) T liên thông và mỗi cạnh là cầu.
- (5) Giữa hai đỉnh phân biệt bất kỳ của T luôn có duy nhất một đường đi sơ cấp.
- (6) T không chứa chu trình nhưng khi thêm một cạnh mới thì có được một chu trình duy nhất.

Chứng minh. Ta chứng minh theo lược đồ sau $((1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (4) \Rightarrow (5) \Rightarrow (6) \Rightarrow (1))$.

$(1) \Rightarrow (2)$ Chỉ cần chứng minh một cây có n đỉnh thì có $n - 1$ cạnh. Chứng minh bằng quy nạp. Điều này hiển nhiên khi $n = 2$. Giả sử cây T có k đỉnh thì có $k - 1$ cạnh. Ta sẽ chứng minh nếu cây T có $k + 1$ đỉnh thì có k cạnh. Thật vậy trong cây T gồm $k + 1$ đỉnh, nếu ta xoá một đỉnh treo và cạnh treo tương ứng thì cây nhận được là một cây có k đỉnh, và do đó suy ra cây này có $k - 1$ cạnh theo giả thiết quy nạp. Điều này có nghĩa cây T có k cạnh.

$(2) \Rightarrow (3)$ Chứng minh phản chứng. Giả sử nếu T có chu trình, thì rõ ràng bỏ đi một cạnh trong chu trình này T vẫn liên thông. Làm lại như thế cho đến khi trong T không còn chu trình nào mà vẫn liên thông. Lúc này ta sẽ thu được một cây có n đỉnh nhưng có ít hơn $n - 1$ cạnh. Điều này trái với giả thiết của (2).

$(3) \Rightarrow (4)$ Theo (3) việc loại bỏ bất kỳ cạnh nào trong T sẽ dẫn đến T có n đỉnh và $n - 1$ cạnh là đồ thị không liên thông. Điều này có nghĩa mọi cạnh trong T là cầu.

$(4) \Rightarrow (5)$ Vì T liên thông nên giữa hai đỉnh phân biệt bất kỳ của T luôn có một đường đi sơ cấp. Giả sử nếu có cặp đỉnh nào đó của T có hai đường đi sơ cấp khác nhau nối chúng thì suy ra T có chứa chu trình. Do đó các cạnh trên chu trình này không phải là cầu. Trái với giả thiết của (4).

$(5) \Rightarrow (6)$ T không chứa bất kỳ một chu trình nào, vì nếu không hai đỉnh phân biệt bất kỳ trên chu trình sẽ được nối bởi hai đường đi sơ cấp. Trái với giả thiết của

(5). Ngoài ra nếu thêm một cạnh mới e nối hai đỉnh u và v nào đó của T thì cạnh này sẽ tạo nên với đường đi sơ cấp duy nhất nối u và v thành một chu trình. Rõ ràng chu trình thu được này là duy nhất, bởi nếu có nhiều hơn một chu trình thì suy ra trong T trước đó phải đã có một chu trình.

(6) $\Rightarrow$ (1) Giả sử T không liên thông. Suy ra T có ít nhất hai thành phần liên thông. Khi đó nếu thêm vào T một cạnh nối hai đỉnh ở hai thành phần liên thông này, ta không thu được một chu trình nào. Mâu thuẫn với giả thiết của (6). Vậy T liên thông, và do đó nó là một cây. $\square$

3.10.3. Cây khung

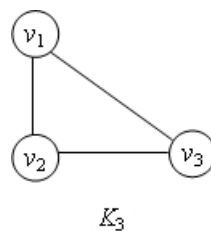
3.10.3.1. Định nghĩa

Cho đồ thị vô hướng liên thông $G = (V, E)$. *Cây khung* (hay *cây bao trùm-spanning tree*) của G là một đồ thị con T của G thỏa các điều kiện sau:

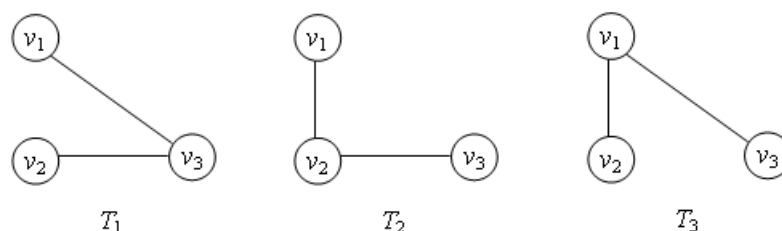
- (i) T là một cây,
- (ii) T chứa tất cả các đỉnh của G .

Như vậy, cây khung của một đồ thị chính là đồ thị con bao trùm gồm tất cả các đỉnh của đồ thị đó. Rõ ràng, mỗi đồ thị vô hướng liên thông G luôn có ít nhất một cây khung. Thật vậy, trong G nếu ta loại bỏ một cạnh nằm trên chu trình nào đó thì ta vẫn nhận được một đồ thị liên thông. Nếu cứ tiếp tục loại bỏ các cạnh ở các chu trình khác cho đến khi nào G không còn chu trình nhưng vẫn liên thông thì ta sẽ thu được một cây nối các đỉnh của G . Cây đó chính là cây khung của G .

Ví dụ 3.72. Xét đồ thị đầy đủ K_3 sau:



Các cây khung của K_3 là:



Kết quả sau cho thấy số lượng của cây khung của một đồ thị là rất lớn.

Định lý 3.51 (Cayley). *Số cây khung của đồ thị đầy đủ K_n là n^{n-2} .*

3.10.3.2. Cây khung nhỏ nhất

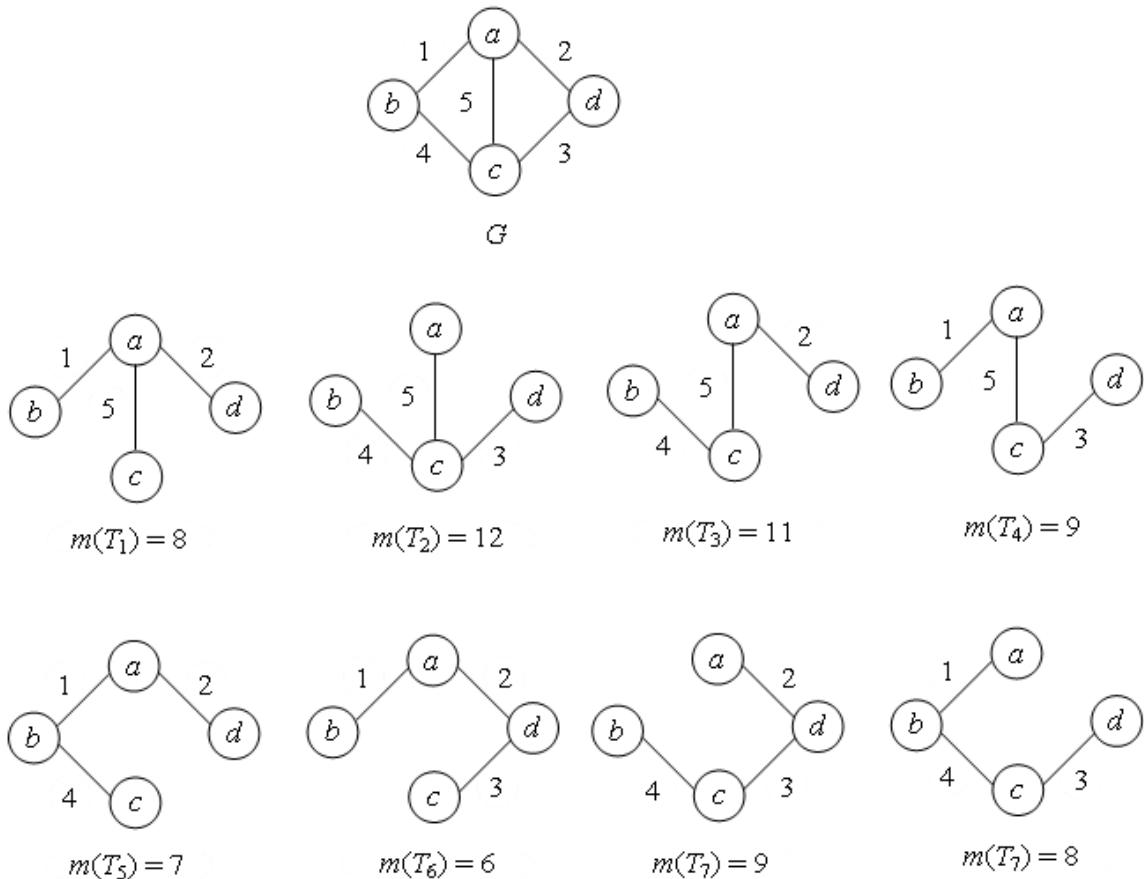
Cho đồ thị vô hướng liên thông có trọng số không âm $G = (V, E, c)$. Giả sử $T = (V, E_T, c|_{E_T})$ là cây khung có trọng số của đồ thị G . Độ dài của cây khung T , ký hiệu $c(T)$, là tổng trọng số của tất cả các cạnh trong cây khung T .

Một cách hình thức:

$$c(T) = \sum_{e \in E_T} c(e).$$

Cây khung có độ dài nhỏ nhất trong tất cả các cây khung của G được gọi là *cây khung nhỏ nhất* (minimum spanning tree-MST) của G . Có thể thấy cây khung nhỏ nhất của một đồ thị không phải duy nhất. Dễ thấy, nếu đồ thị có các cạnh với trọng số đôi một khác nhau thì cây khung nhỏ nhất là duy nhất.

Ví dụ 3.73. Sau đây là đồ thị vô hướng, liên thông có trọng số G và các cây khung của nó:



Như vậy, cây khung nhỏ nhất của G là T_6 với độ dài 6.

3.10.3.3. Bài toán cây khung nhỏ nhất

Bài toán cây khung nhỏ nhất được phát biểu như sau: “Cho đồ thị vô hướng liên thông có trọng số $G = (V, E, c)$. Hãy tìm cây khung nhỏ nhất của G ”.

Bài toán này là một trong những bài toán tối ưu trên đồ thị, có nhiều ứng dụng trong nhiều lĩnh vực khác nhau của đời sống. Dưới đây là hai mô hình thực tế tiêu biểu về ứng dụng của bài toán cây khung nhỏ nhất.

Bài toán 3.2 (Bài toán xây dựng hệ thống đường sắt). Hãy xây dựng một hệ thống đường sắt nối n thành phố sao cho hành khách có thể đi từ một thành phố bất kỳ đến tùy ý một trong số các thành phố còn lại, với chi phí về xây dựng là nhỏ nhất.

Bài toán này có thể mô hình hóa bằng đồ thị đầy đủ có trọng số không âm gồm n đỉnh, trong đó mỗi đỉnh là một thành phố, mỗi cạnh là một tuyến đường sắt nối giữa hai thành phố và chi phí xây dựng của mỗi tuyến đường sắt là trọng số trên mỗi cạnh tương ứng. Khi đó, phương án xây dựng hệ thống đường sắt tối ưu chính là xây dựng cây khung nhỏ nhất của đồ thị đầy đủ này.

Bài toán 3.3 (Bài toán nối mạng máy tính). Nối mạng một hệ thống gồm n máy tính đánh số từ 1 đến n , với chi phí nối máy i với máy j ($i, j = 1, 2, \dots, n$) là độ dài dây cáp nối tương ứng. Hãy nối mạng hệ thống máy tính trên sao cho tổng chi phí là nhỏ nhất.

Bài toán này cũng có thể mô hình hóa bằng đồ thị đầy đủ có trọng số không âm gồm n đỉnh, trong đó mỗi đỉnh là một máy tính, mỗi cạnh là một dây cáp nối giữa hai máy và chi phí của mỗi dây cáp là trọng số trên mỗi cạnh tương ứng. Rõ ràng, phương án xây dựng hệ thống máy tính tối ưu chính là xây dựng cây khung nhỏ nhất của đồ thị đầy đủ này.

3.10.3.4. Thuật toán tìm cây khung nhỏ nhất

Để giải bài toán cây khung nhỏ nhất chúng ta có thể liệt kê tất cả các cây khung của đồ thị và chọn trong số chúng cây khung nhỏ nhất. Phương pháp như vậy trong trường hợp đồ thị là đầy đủ sẽ có độ phức tạp thời gian $O(n^{n-2})$. Do đó, ngay cả đồ thị với số đỉnh khoảng hàng chục thì phương pháp này đã khó thực hiện được. Tuy nhiên, rất may với bài toán này chúng ta khá nhiều thuật toán hiệu quả để giải chúng. Ở đây, chúng ta sẽ tìm hiểu hai trong số các thuật toán phổ biến và hiệu quả là thuật toán Kruskal và thuật toán Prim.

(1) Thuật toán Kruskal

Thuật toán Kruskal là một thuật toán tham lam được đề xuất bởi Joseph Kruskal¹² năm 1956. Thuật toán này còn được hiểu là “thuật toán dựa trên cạnh”.

Thuật toán 3.11 (*Thuật toán Kruskal*)

Vào: Đồ thị vô hướng liên thông, có trọng số $G = (V, E, c)$ với n đỉnh.

Ra: Cây khung nhỏ nhất T của G .

Phương pháp:

Bước 1. (*Khởi tạo*) Cây T gồm n đỉnh nhưng không có cạnh.

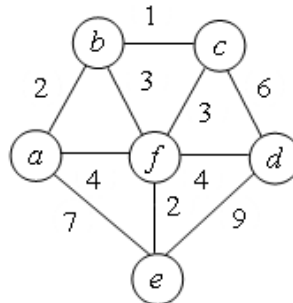
Bước 2. (*Sắp xếp*) Sắp xếp các cạnh của G theo thứ tự không giảm của trọng số.

Bước 3. (*Lặp: thêm cạnh*) Lần lượt lấy từng cạnh trong danh sách cạnh đã sắp ở Bước 2 thêm dần vào T theo nguyên tắc các cạnh thêm vào không được tạo thành chu trình trong T . Lặp lại quá trình này cho đến khi số cạnh trong T bằng $n - 1$.

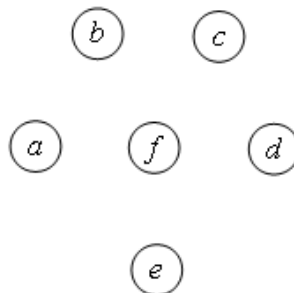
Bước 4. (*Kết luận*) T là cây khung nhỏ nhất cần tìm.

Rõ ràng, Bước 2 là bước tính toán nhiều nhất của Thuật toán Kruskal. Do đó, độ phức tạp thời gian của Thuật toán Kruskal là $O(m \log m)$, với m là số cạnh của đồ thị G . Như vậy, khi đồ thị có số cạnh nhỏ, chúng ta nên sử dụng Thuật toán Kruskal.

Ví dụ 3.74. Xét đồ thị vô hướng liên thông, có trọng số G gồm 6 đỉnh như sau:



Áp dụng Thuật toán Kruskal, khởi đầu cây T là:

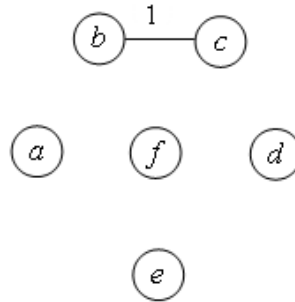


¹²Joseph Bernard Kruskal (1928): nhà toán học, thống kê và khoa học máy tính. Nhiều kết quả được mang tên ông như TT Kruskal, định lý cây Kruskal và định lý Kruskal-Katona. TT Kruskal lần đầu xuất hiện trong công trình *Proceedings of the American Mathematical Society* (1956).

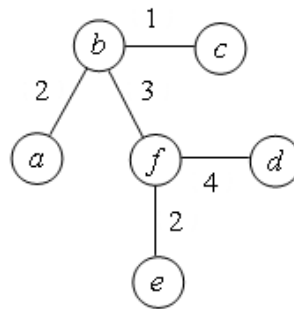
và các cạnh sau khi sắp thứ tự:

$$(b, c), (a, b), (e, f), (b, f), (c, f), (a, f), (d, f), (c, d), (a, e), (d, e)$$

Đầu tiên thêm cạnh (b, c) vào cây T , thỏa điều kiện của thuật toán:



Lặp lại quá trình thêm cạnh như trên vào cây T sao cho thỏa điều kiện của thuật toán, ta sẽ thu được cây khung nhỏ nhất T với độ dài 12 là:



Định lý 3.52. Thuật toán Kruskal cho ra cây khung nhỏ nhất T của đồ thị vô hướng liên thông có trọng số gồm n đỉnh G .

(2) Thuật toán Prim

Tương tự Thuật toán Kruskal, Thuật toán Prim cũng là một thuật toán tham lam. Thuật toán này lần đầu được tìm ra bởi Vojtech Jarnik¹³ năm 1930, sau đó độc lập bởi Robert C. Prim¹⁴ năm 1957 và cuối cùng là Edsger Dijkstra năm 1959. Vì vậy nó còn được gọi là Thuật toán DJP hay Thuật toán Jarnik. Thuật toán này còn được hiểu “thuật toán dựa trên đỉnh-cạnh”.

¹³Vojtech Jarnik (1897-1970): nhà toán học Séc. Lĩnh vực chính của ông là lý thuyết số và giải tích toán học. Ông cũng phát triển lý thuyết đồ thị với TT Prim đã biết.

¹⁴Robert Clay Prim (1921): nhà toán học và khoa học máy tính Mỹ.

Thuật toán 3.12 *Thuật toán Prim*

Vào: Đồ thị vô hướng liên thông có trọng số $G = (V, E, c)$ với n đỉnh.

Ra: Cây khung nhỏ nhất T của G .

Phương pháp:

Bước 1. (*Khởi tạo*) Cây T gồm một đỉnh bất kỳ.

Bước 2. (*Lặp: thêm đỉnh-cạnh*) Lần lượt tìm cạnh $e = (u, v) \in E$ có trọng số nhỏ nhất sao cho u là một đỉnh trong T và v là một đỉnh không ở trong T . Khi đó, bổ sung v và e vào T . Lặp lại quá trình này cho đến khi số đỉnh trong T bằng n .

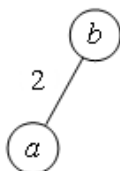
Bước 3. (*Kết luận*) T là cây khung nhỏ nhất cần tìm.

Rõ ràng, độ phức tạp thời gian của Thuật toán Prim là $O(m \log n)$, với m là số cạnh của đồ thị G . So với Thuật toán Kruskal làm việc kém hiệu quả đối với những đồ thị dày (số cạnh $m \approx n(n-1)/2$), Thuật toán Prim hiệu quả hơn nhiều.

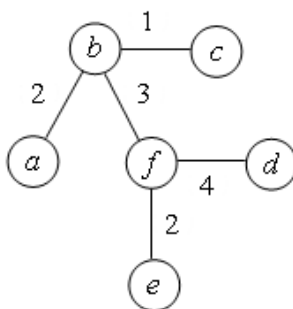
Ví dụ 3.75. Xét đồ thị tương tự như trong Ví dụ 3.74. Áp dụng Thuật toán Prim, khởi đầu cây T chẳng hạn gồm một đỉnh a là:



Đầu tiên, bổ sung đỉnh b và cạnh (a, b) với trọng số 2 vào cây T :



Tương tự như trên, lặp lại quá trình bổ sung đỉnh-cạnh thỏa Thuật toán Prim cho đến khi số đỉnh trong T bằng 6. Khi đó, cây khung nhỏ nhất T thu được với độ dài 12 là:



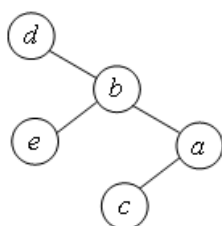
Định lý 3.53. *Thuật toán Prim cho ra cây khung nhỏ nhất T của đồ thị vô hướng liên thông có trọng số gồm n đỉnh G .*

3.10.4. Cây phân cấp

3.10.4.1. Định nghĩa

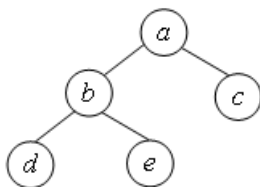
Cây phân cấp là một cây, trong đó có một đỉnh đặc biệt gọi là *gốc*, giữa các đỉnh có một mối quan hệ phân cấp còn gọi là *quan hệ cha-con*.

Chẳng hạn, xét cây T với gốc a . Khi đó nếu có hai đỉnh b, c kề với a thì b, c được gọi là con của a (hoặc gọi a là cha của b, c). Tương tự, nếu có d, e kề với b thì b là cha của d, e (hoặc d, e là con của b). Xem hình vẽ sau:



Hình 3.7: *Cây phân cấp*

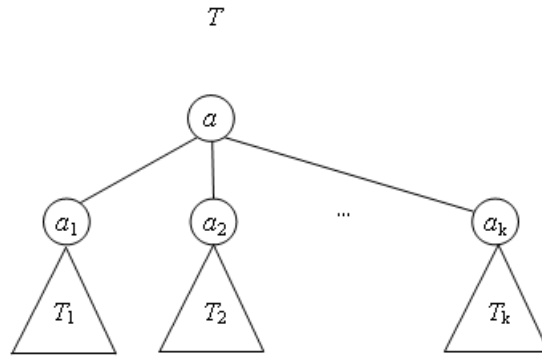
Người ta còn gọi cây phân cấp là *cây có gốc*. Một cây có gốc thường được vẽ với gốc ở trên cùng và cây phát triển từ trên xuống. Chẳng hạn, cây có gốc a ở hình vẽ trên còn được vẽ lại:



Hình 3.8: *Cây có gốc a*

Cây phân cấp cũng còn có thể được định nghĩa đệ quy như sau:

- (i) Tập rỗng là một cây phân cấp, còn được gọi là *cây rỗng*.
- (ii) Một đỉnh là một cây phân cấp.
- (iii) Giả sử a là một đỉnh và $T_1, T_2, \dots, T_k$ là k cây phân cấp với các gốc tương ứng là $a_1, a_2, \dots, a_k$. Cây T được xây dựng bằng cách cho đỉnh a làm “cha” của các đỉnh $a_1, a_2, \dots, a_k$ sẽ là một cây phân cấp. Trong cây T này, đỉnh a là gốc và $T_1, T_2, \dots, T_k$ là các cây con của gốc a .



Hình 3.9: Cây phân cấp tổng quát

Định nghĩa đệ quy của cây trong nhiều trường hợp thuận tiện trong việc mô tả chặt chẽ các khái niệm khác.

Trong một cây các đỉnh cha được gọi là *đỉnh trong*, các đỉnh còn lại không phải là đỉnh trong được gọi là *đỉnh lá*. Như vậy, đỉnh lá là đỉnh không có con và gốc là đỉnh duy nhất không có cha.

Mức của đỉnh trong cây được định nghĩa như sau:

- (i) Mức của gốc bằng 0.
- (ii) Nếu mức của đỉnh cha là i thì mức của các đỉnh con là $i + 1$.

Mức lớn nhất của một đỉnh bất kỳ trong cây được gọi là *chiều cao của cây*.

Chẳng hạn, cây có gốc a ở Hình 3.8 trên có chiều cao là 2. Cây phân cấp được sử dụng nhiều trong thực tế chẳng hạn như:

- Mục lục của một cuốn sách để đọc giả tiện tra cứu.
- Sơ đồ tổ chức của một cơ quan để khách tiện liên hệ.
- Cấu trúc thư mục trên một ổ đĩa của máy tính để quản lý các file, ...

Nếu mỗi đỉnh trong một cây phân cấp T có không quá m con thì T được gọi là cây m -phân. Cây m -phân T được gọi là *cây đầy đủ* nếu mỗi đỉnh trong của T đều có đúng m con.

3.10.4.2. Cây nhị phân

Cây nhị phân là một cây m -phân với $m = 2$. Tức là cây sao cho mỗi đỉnh có không quá hai con.

Trong cây nhị phân, mỗi đỉnh con thường được chỉ rõ là con bên trái hay con bên phải, con bên trái (tương ứng phải) được vẽ phía dưới và bên trái (tương ứng phải) của đỉnh cha. Cây nhị phân là một cây rất quan trọng và có nhiều ứng dụng rộng rãi.

Định lý 3.54. Số các cây nhị phân n đỉnh là

$$c_n = \frac{1}{n+1} \binom{2n}{n}.$$

Dãy số c_n ($n = 0, 1, 2, \dots$) do Catalan tìm ra, và thường được gọi là *dãy số Catalan*. Trong nhiều ứng dụng, người ta cần phải “viếng thăm” các đỉnh của cây nhị phân một cách có hệ thống để truy vấn dữ liệu. Quá trình thăm tất cả các đỉnh của cây nhị phân sao cho mỗi đỉnh thăm một lần được gọi là *duyệt cây nhị phân*. Như vậy, duyệt cây là đưa ra danh sách tuyến tính các đỉnh của cây, trong đó mỗi đỉnh chính xác một lần. Có ba phương pháp duyệt cây nhị phân: *duyệt theo thứ tự trước* (hay *tiền thứ tự*), *duyệt theo thứ tự giữa* (hay *trung thứ tự*) và *duyệt theo thứ tự sau* (hay *hậu thứ tự*). Các phương pháp duyệt này khác nhau chủ yếu ở thứ tự thăm các đỉnh, được định nghĩa đệ quy như sau.

- (1) Nếu cây nhị phân T rỗng thì cả ba cách duyệt đều cho ra danh sách đỉnh rỗng.
- (2) Nếu cây nhị phân T chỉ có một đỉnh thì cả ba cách duyệt đều cho ra danh sách gồm một đỉnh này.

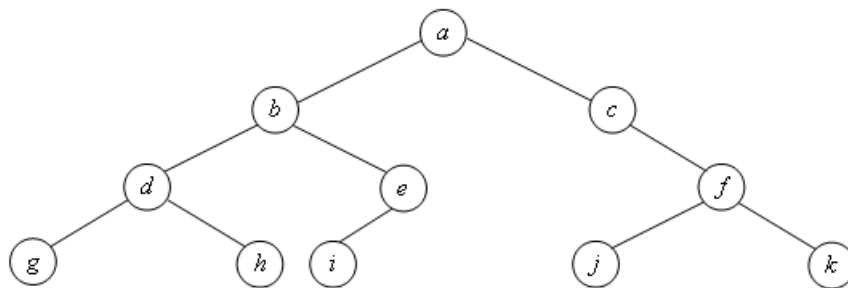
(3) Nếu cây nhị phân T có gốc a và các cây con T_1, T_2 thì:

- Duyệt theo thứ tự trước của T là danh sách gồm gốc a , sau đó là các đỉnh của cây con T_1 được duyệt theo thứ tự trước và cuối cùng các đỉnh của cây con T_2 được duyệt theo thứ tự trước.

- Duyệt theo thứ tự giữa của T là danh sách gồm các đỉnh của cây con T_1 được duyệt theo thứ tự giữa, sau đó là gốc a và cuối cùng đến các đỉnh của cây con T_2 được duyệt theo thứ tự giữa.

- Duyệt theo thứ tự sau của T là danh sách gồm các đỉnh của cây con T_1 được duyệt theo thứ tự sau, rồi đến các đỉnh của cây con T_2 được duyệt theo thứ tự sau và cuối cùng là gốc a .

Ví dụ 3.76. Xét cây nhị phân T như sau:



Hình 3.10: Cây nhị phân

Các cách duyệt cây nhị phân T là:

Duyệt theo thứ tự trước: $a, b, d, g, h, e, i, c, f, j, k$

Duyệt theo thứ tự giữa: $g, d, h, b, i, e, c, j, f, k$

Duyệt theo thứ tự sau: $g, h, d, i, e, b, j, k, f, c, a$

3.10.4.3. Cây biểu thức

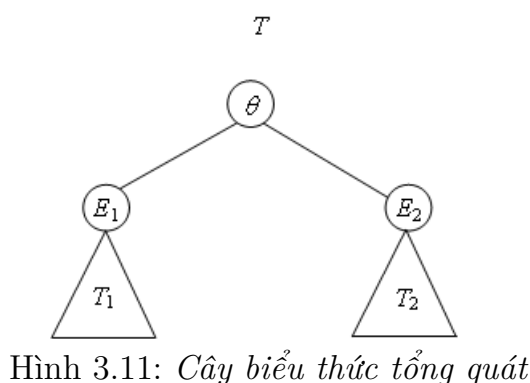
Chúng ta có thể dùng cây nhị phân để biểu diễn các biểu thức số học (cũng như các mệnh đề phức hợp hay tổ hợp các tập hợp). Cây như vậy được gọi là cây biểu thức. Một cách hình thức cây biểu thức được định nghĩa như sau.

Cây biểu thức là một cây nhị phân sao cho mỗi đỉnh của nó được gán nhãn theo quy tắc:

(i) Các lá được gán bằng các đại lượng.

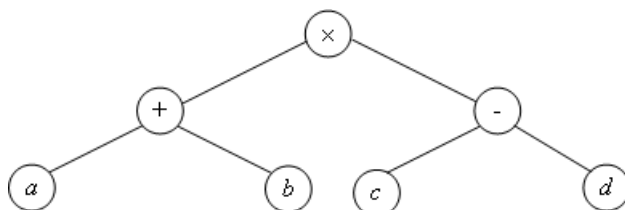
(ii) Các đỉnh trong được gán bằng các dấu phép toán của một biểu thức nào đó.

Chẳng hạn, biểu thức $E = (E_1)\theta(E_2)$ và các cây biểu thức T_1, T_2 biểu diễn các biểu thức con E_1, E_2 thì cây biểu thức T biểu diễn E được xây dựng như sau:



Ví dụ 3.77. Xét biểu thức đại số $E = (a + b) \times (c - d)$.

Khi đó, cây biểu thức tương ứng là:



Duyệt cây nhị phân trên theo giữa ta được biểu thức *dạng trung tố* $a + b \times c - d$. Rõ ràng, đây là biểu thức đại số E đã bỏ đi các dấu ngoặc. Chúng ta biết rằng các dấu ngoặc trong E là rất quan trọng, bỏ nó đi sẽ dẫn đến có thể hiểu E theo

hiều cách. Vì vậy, người ta nói cách viết (ký pháp) quen thuộc trong đại số học là *ký pháp trung tự* kèm theo các dấu ngoặc.

Trường hợp duyệt cây nhị phân trên theo thứ tự trước (tương ứng sau) ta được biểu thức *dạng tiền tố* $\times + ab - cd$ (tương ứng biểu thức *dạng hậu tố* $ab + cd - \times$). Có thể thấy các biểu thức ở dạng tiền tố và hậu tố là rõ ràng, tức là không cần đến dấu ngoặc trong các biểu thức này. Các biểu thức dạng tiền tố và hậu tố này tương ứng còn được gọi là *ký pháp Ba lan* và *ký pháp Ba lan đảo*. Cách gọi như thế là để ghi nhớ đóng góp của nhà toán học và logic học Ba lan Lukasiewicz (1878-1956) trong vấn đề này. Việc chuyển một biểu thức viết theo ký pháp quen thuộc (có dấu ngoặc) sang dạng ký pháp Ba lan hay ký pháp Ba lan đảo có thể thực hiện bằng cách vẽ cây biểu thức tương ứng, sau đó duyệt cây theo thứ tự trước và thứ tự sau. Tuy nhiên, trường hợp ngược lại thay vì vẽ cây chúng ta có thể xem xét để xác định dần các công thức bộ phận của công thức đã cho.

Ví dụ 3.78. Xét biểu thức viết theo ký pháp Ba lan sau:

$$\times + ab / - cd 2$$

Hãy chuyển biểu thức trên sang ký pháp quen thuộc. Ta có:

$$\begin{aligned} & \times \underbrace{+ab}_{a+b} / \underbrace{-cd}_{c-d} 2 \\ & \times (a+b) / \underbrace{(c-d)2}_{\frac{c-d}{2}} \\ & \times (a+b) \left(\frac{c-d}{2} \right) \\ & (a+b) \times \left(\frac{c-d}{2} \right) \end{aligned}$$

Như vậy, ký pháp quen thuộc của biểu thức viết theo ký pháp Ba lan trên là:

$$(a+b) \times \left(\frac{c-d}{2} \right).$$

Ví dụ 3.79. Xét biểu thức viết theo ký pháp Ba lan đảo là:

$$ab + cd - 2 / \times$$

Khi đó:

$$\begin{aligned} & \underbrace{ab+cd-2}_{a+b \quad c-d} / \times \\ & (a+b) \underbrace{(c-d)2}_{c-d} / \times \\ & \quad \quad \quad \frac{c-d}{2} \\ & (a+b) \left(\frac{c-d}{2} \right) \times \\ & (a+b) \times \left(\frac{c-d}{2} \right) \end{aligned}$$

Do đó, ký pháp quen thuộc của biểu thức viết theo ký pháp Ba lan đảo trên là:

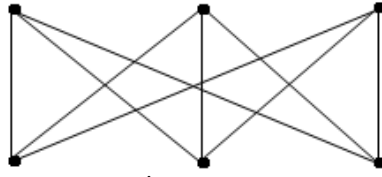
$$(a+b) \times \left(\frac{c-d}{2} \right).$$

3.11. Đồ thị phẳng

Một trong những vấn đề quan trọng của lý thuyết đồ thị là nghiên cứu câu hỏi làm thế nào để biểu diễn được đồ thị trên mặt phẳng sao cho không có hai đường biểu diễn cạnh nào của chúng là cắt nhau. Không phải đồ thị nào cũng có thể biểu diễn được như vậy và những đồ thị có thể biểu diễn thỏa mãn điều kiện đó được gọi là đồ thị phẳng (planar graph). Bài toán “ba nhà, ba giếng” từ thời xa xưa sau chính là một trong các bài toán dẫn đến khái niệm đồ thị phẳng.

Bài toán 3.4 (Ba nhà, ba giếng). Có ba nhà ở gần ba cái giếng, các nhà không có đường nối thẳng đến với nhau và các giếng cũng như vậy không có đường nối thẳng đến với nhau. Có một lần bất hòa, họ tìm cách làm các con đường khác nhau dẫn đến các giếng sao cho các đường này đôi một không cắt nhau. Hỏi họ có thể thực hiện được điều này không?

Bài toán này có thể mô hình hóa bằng đồ thị như sau: mỗi nhà, mỗi giếng biểu diễn bằng một đỉnh và con đường từ mỗi nhà đến mỗi giếng biểu diễn bằng một cạnh. Khi đó, đồ thị tương ứng cho bài toán này chính là đồ thị hai phía đầy đủ $K_{3,3}$:



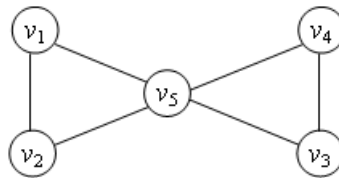
Hình 3.12: Đồ thị ba nhà, ba giếng

Câu trả lời cho bài toán này là có ngay trong mục sau.

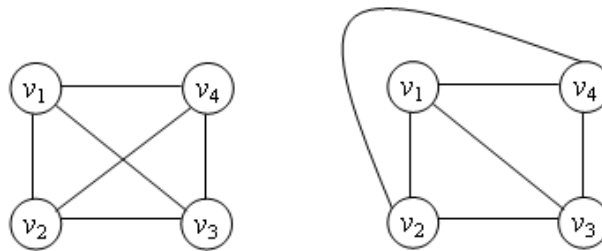
3.11.1. Định nghĩa

Đồ thị G được gọi là *phẳng* nếu ta có thể vẽ được nó trên mặt phẳng sao cho không có hai cạnh bất kỳ nào cắt nhau ở ngoài đỉnh. Cách vẽ như vậy được gọi là *biểu diễn phẳng* của đồ thị.

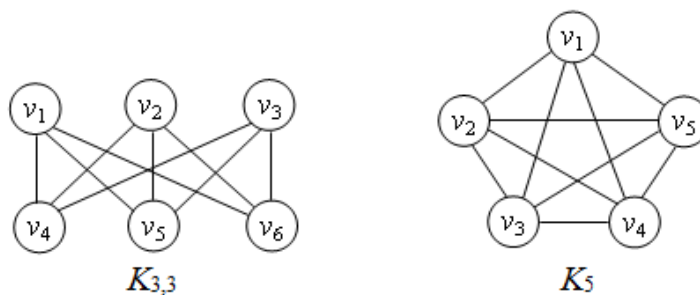
Ví dụ 3.80. Đồ thị cánh bướm sau là đồ thị phẳng:



Đồ thị K_4 sau là đồ thị phẳng bởi vì có thể vẽ lại nó như hình bên không có hai đường nào cắt nhau ở ngoài đỉnh:



Ví dụ 3.81. Các đồ thị $K_{3,3}$ và K_5 sau là các đồ thị không phẳng:



Chúng ta không thể biểu diễn phẳng được cho các đồ thị này (theo định nghĩa). Như vậy, không thể thực hiện được các con đường như trong Bài toán 3.4.

Các đồ thị không phẳng K_5 và $K_{3,3}$ có một số tính chất cơ bản sau:

- Nếu xóa một cạnh hoặc một đỉnh của đồ thị thì đồ thị thu được sẽ là một đồ thị phẳng.
- K_5 là đồ thị không phẳng có số đỉnh ít nhất và $K_{3,3}$ là đồ thị không phẳng có số cạnh ít nhất.

Ví dụ 3.82 (Biểu diễn phẳng của bản đồ). Từ một bản đồ địa lý thế giới, chúng ta xây dựng một đồ thị với mỗi nước là một đỉnh và hai đỉnh được nối với nhau bằng một cạnh nếu hai nước tương ứng có chung đường biên giới. Rõ ràng, chúng ta có thể thấy ngay đồ thị nhận được là một đồ thị phẳng.

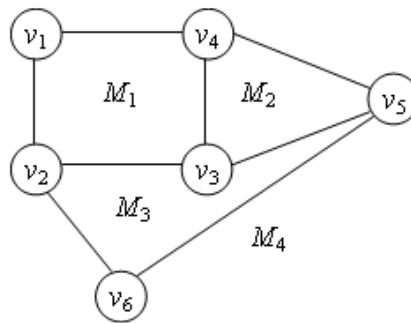
Những đồ thị mà có thể biến đổi cho trùng nhau bằng phép biến dạng co dãn liên tục trên mặt phẳng thì không được coi là những đồ thị phẳng khác nhau.

3.11.2. Một số tính chất cơ bản

Cho đồ thị phẳng G . Mỗi phần mặt phẳng trong G giới hạn bởi một chu trình đơn không chứa bên trong nó một chu trình đơn khác được gọi là một *miền* (hay *diện*) (*hữu hạn*) của G . Chu trình giới hạn một miền được gọi là *biên* của miền. Phần mặt phẳng bên ngoài tất cả các miền hữu hạn (tức phần bù của hợp tất cả các miền hữu hạn) của đồ thị phẳng liên thông được gọi là *miền vô hạn*. Số cạnh ít nhất tạo thành một biên (tức độ dài của chu trình ngắn nhất) của G hoặc số cạnh của G nếu G không có chu trình được gọi là *đai* của G , ký hiệu δ .

Có thể thấy miền vô hạn là duy nhất.

Ví dụ 3.83. Xét đồ thị phẳng G có 6 đỉnh và 8 cạnh như sau:



Đồ thị G có 4 miền, trong đó M_4 là miền vô hạn, M_1 là miền có biên $v_1v_2v_3v_4v_1$, M_2 là miền có biên $v_3v_4v_5v_3$ và M_3 là miền có biên $v_2v_3v_6v_2$. Chu trình đơn

$v_1v_2v_3v_5v_4v_1$ không giới hạn một miền vì chứa bên trong nó có một chu trình đơn khác chẳng hạn là $v_3v_4v_5v_3$.

Euler đã chứng minh được các cách biểu diễn phẳng khác nhau của một đồ thị phẳng luôn chia mặt phẳng ra thành cùng một số miền, đó là Euler đã tìm ra mối liên hệ giữa số miền, số đỉnh và số cạnh của đồ thị phẳng.

Định lý 3.55 (Euler, 1752). *Nếu đồ thị phẳng là liên thông có n đỉnh, m cạnh và d miền thì*

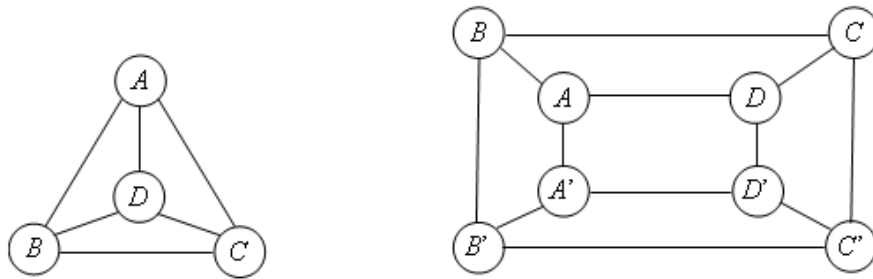
$$n - m + d = 2.$$

Chứng minh. Xét đồ thị phẳng G thỏa các điều kiện của định lý. Bây giờ, ta thực hiện loại bỏ một số cạnh của G để thu được một cây khung của G . Dễ thấy, mỗi lần bỏ một cạnh thì số miền của G cũng giảm một, tuy nhiên số đỉnh của G không thay đổi. Điều này có nghĩa giá trị của biểu thức $n - m + d$ không thay đổi trong suốt quá trình loại bớt cạnh của G để thu được một cây khung. Như vậy, cây khung thu được này có n đỉnh, và do đó có $n - 1$ cạnh cùng với một miền duy nhất. Do đó

$$n - m + d = n - (n - 1) + 1 = 2.$$

□

Hệ thức $n - m + d = 2$ còn được gọi là “hệ thức Euler cho hình đa diện”, vì được Euler chứng minh đầu tiên cho hình đa diện có n đỉnh, m cạnh và d mặt. Mỗi hình đa diện có thể coi là một đồ thị phẳng. Chẳng hạn, hình tứ diện $ABCD$ và hình hộp $ABCD A'B'C'D'$ có thể biểu diễn bằng các đồ thị phẳng dưới đây.



Hình 3.13: Biểu diễn phẳng của hình tứ diện và hình hộp

Vậy trong một đồ thị phẳng và liên thông thì số miền của đồ thị là $d = m - n + 2$. Nếu ký hiệu h là số miền hữu hạn, tức $h = d - 1$, thì số miền hữu hạn sẽ là $h = m - n + 1$.

Hệ quả 3.11. *Trong một đồ thị phẳng liên thông tùy ý, luôn tồn tại ít nhất một đỉnh có bậc không vượt quá 5.*

Chứng minh. Xét đồ thị phẳng liên thông tùy ý G , và giả sử mọi đỉnh của G đều có bậc không nhỏ hơn 6. Suy ra mỗi đỉnh của G phải là đầu mút của ít nhất 6 cạnh. Hơn nữa mỗi cạnh lại có hai đầu mút, nên từ đây ta có bất đẳng thức $6n \leq 2m$ hay $3n \leq m$. Mặt khác, mỗi miền trong đồ thị phẳng được bao ít nhất bằng 3 cạnh và mỗi cạnh có thể nằm trên biên của tối đa hai miền. Do đó, ta có bất đẳng thức $3d \leq 2m$. Cộng vế theo vế hai bất đẳng thức $3n \leq m$, $3d \leq 2m$ và rút gọn ta thu được bất đẳng thức $d + n \leq m$. Bất đẳng thức này mâu thuẫn với hệ thức Euler ở trên. $\square$

Định lý Euler có thể được phát biểu tổng quát lại như sau.

Định lý 3.56 (Euler tổng quát). *Nếu đồ thị phẳng G có n đỉnh, m cạnh và d miền thì*

$$n - m + d = \omega(G) + 1$$

hay còn nói trong G số miền hữu hạn bằng chu số:

$$h = m - n + \omega(G) = v(G).$$

Hệ thức Euler còn có thể dùng để thiết lập một số bất đẳng thức cho các đồ thị phẳng. Các bất đẳng thức này chính là điều kiện cần của hệ thức Euler.

Hệ quả 3.12. *Nếu đơn đồ thị phẳng là liên thông có $n \geq 3$ đỉnh và m cạnh thì $m \leq 3n - 6$.*

Chẳng hạn, xét đồ thị đầy đủ K_5 . Vì K_5 có 5 đỉnh và 10 cạnh, nên không thỏa mãn bất đẳng thức trong Hệ quả 3.12. Suy ra, K_5 không phải đồ thị phẳng.

Hệ quả 3.13. *Nếu đơn đồ thị phẳng là liên thông có $n \geq 3$ đỉnh, m cạnh và không có chu trình độ dài 3 thì $m \leq 2n - 4$.*

Chẳng hạn, xét đồ thị hai phía đầy đủ $K_{3,3}$. Vì $K_{3,3}$ có 6 đỉnh, 9 cạnh và không có chu trình độ dài 3 (đồ thị hai phía). Do đó, bất đẳng thức trong Hệ quả 3.13 không thỏa mãn, suy ra $K_{3,3}$ không phải đồ thị phẳng. Đây cũng chính là câu trả lời cho bài toán ba nhà ba giếng.

Hệ quả 3.14. *Nếu đồ thị phẳng G có n đỉnh, m cạnh và đại của nó $\delta \geq 3$ thì*

$$m \leq \frac{\delta}{\delta - 2}(n - 2).$$

Chứng minh. Gọi d là số miền của G . Ta có $d\delta \leq 2m$. Theo công thức Euler suy ra

$$(m - n + 2)\delta \leq 2m$$

hay

$$m \leq \frac{\delta}{\delta - 2}(n - 2).$$

□

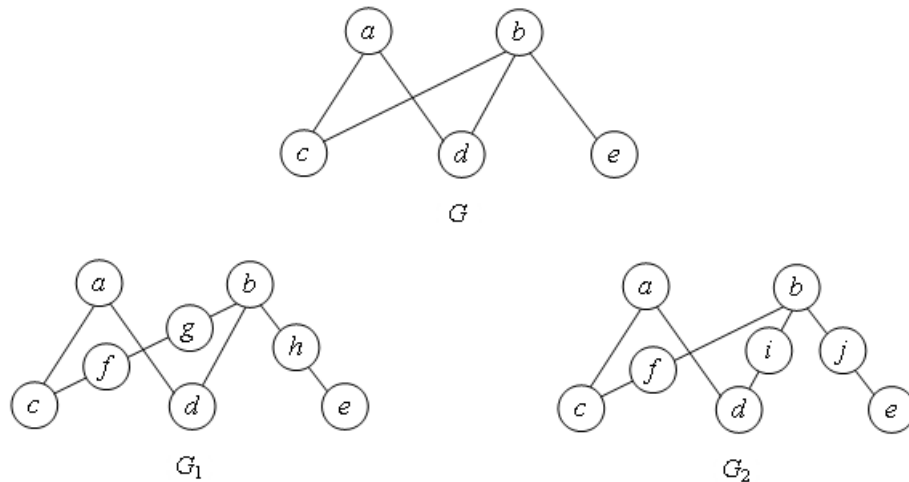
Vận dụng Hệ quả 3.14 chúng ta cũng có thể khẳng định được các đồ thị $K_{3,3}$ và K_5 là không phẳng.

Hệ quả 3.15.

- (1) Đồ thị đầy đủ K_n là không phẳng khi $n \geq 5$ và phẳng khi $n \leq 4$.
- (2) Đồ thị hai phía đầy đủ $K_{m,n}$ là phẳng khi và chỉ khi $m \leq 2$ hoặc $n \leq 2$.

Từ các khẳng định trên chúng ta thấy hai đồ thị $K_{3,3}$ và K_5 là không phẳng. Do đó, có thể thấy một đồ thị là không phẳng nếu nó chứa một trong hai đồ thị này như là đồ thị con. Hơn nữa tất cả các đồ thị không phẳng cần phải chứa đồ thị con nhận được từ $K_{3,3}$ hoặc K_5 bằng một số phép toán biến đổi nào đó. Phép toán loại bỏ một cạnh bất kỳ (u, v) trong đồ thị G sau đó thêm vào đỉnh mới ω cùng hai cạnh mới (u, ω) và (ω, v) được gọi là *phép phân chia sơ cấp* trong G . Đồ thị H thu được từ đồ thị G bằng một dãy các phép phân chia sơ cấp được gọi là *đồ thị đồng phôi* với G (hay G là một *cấu hình* của H).

Ví dụ 3.84. Xét các đồ thị sau:



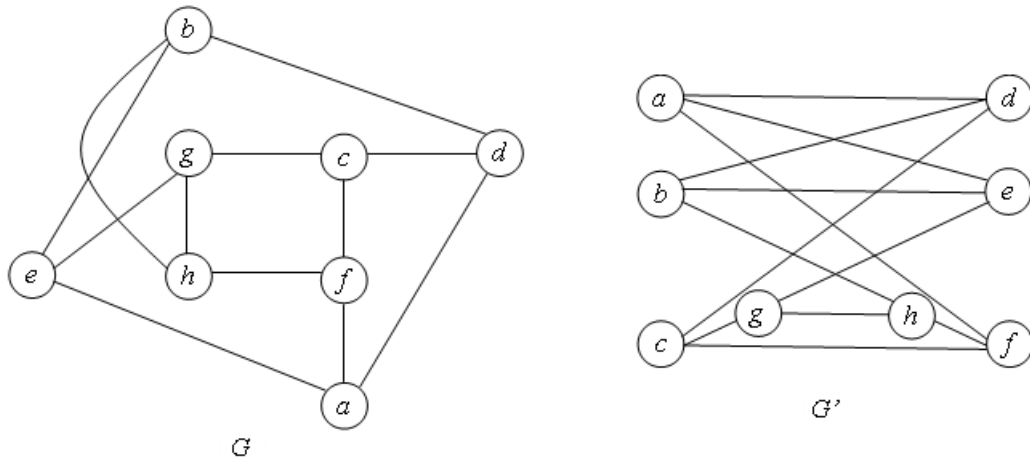
Ta có G_1 và G_2 là các đồ thị đồng phôi với đồ thị G .

Với khái niệm đồ thị con đồng phôi này, năm 1930 nhà toán học Ba Lan Kuratowski đã thiết lập một định lý biểu thị đặc điểm của các đồ thị phẳng. Qua định lý này Kuratowski đã phân lớp được các đồ thị phẳng.

Định lý 3.57 (Kuratowski, 1930). *Một đồ thị là phẳng khi và chỉ khi nó không chứa một đồ thị con đồng phôi với $K_{3,3}$ và K_5 .*

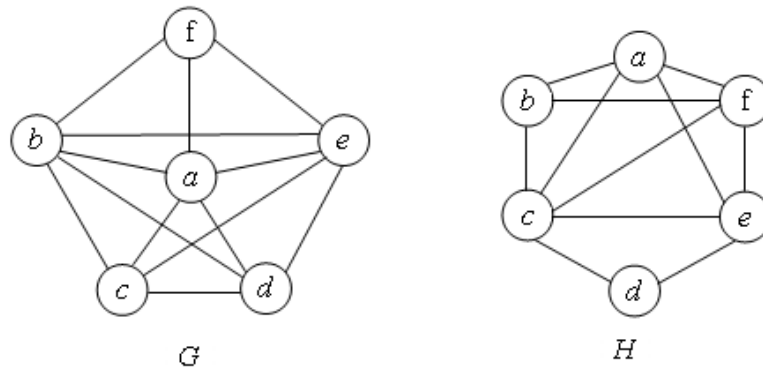
Chứng minh. Điều kiện cần là rõ ràng, bởi mọi đồ thị chứa đồ thị con đồng phôi với $K_{3,3}$ hoặc K_5 là không phẳng. Chứng minh điều kiện đủ là rất phức tạp, ở đây chúng ta không trình bày chi tiết. $\square$

Ví dụ 3.85. Xét đồ thị G sau:



Ta có đồ thị G' là vẽ lại của đồ thị G . Đồ thị G' là đồng phôi với đồ thị $K_{3,3}$. Do đó, theo Định lý Kuratowski G là đồ thị không phẳng.

Ví dụ 3.86. Xét các đồ thị sau đây:



Theo Định lý Kuratowski ta có đồ thị G là không phẳng, vì nếu xoá đỉnh f cùng các cạnh $(f, b), (f, a), (f, e)$ thì ta nhận được đồ thị con K_5 .

Đồ thị H là phẳng, vì H có 6 đỉnh nhưng không thể chứa đồ thị con $K_{3,3}$ bởi có đỉnh bậc 2, trong khi các đỉnh của $K_{3,3}$ đều có bậc 3. Ngoài ra H cũng không thể chứa đồ thị con K_5 bởi có hai đỉnh bậc nhỏ hơn 4, trong khi tất cả các đỉnh của K_5 đều có bậc 4.

Ví dụ 3.87. Đồ thị Peterson cũng là một đồ thị không phẳng, bởi nó chứa một đồ thị con đồng phôi với đồ thị hai phía đầy đủ $K_{3,3}$ [5].

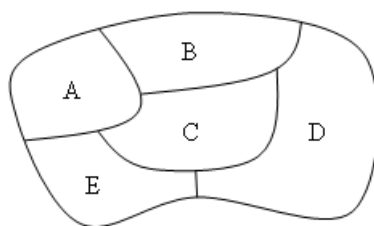
3.12. Tô màu đồ thị

3.12.1. Tô màu bản đồ

Những bài toán liên quan đến tô màu bản đồ đã dẫn đến rất nhiều kết quả trong lý thuyết đồ thị. Trong một bản đồ, hai miền có chung nhau một đường biên được xem là *hai miền kề nhau* (hai miền chỉ có chung nhau một điểm biên không được coi là kề nhau). Một bản đồ thường được *tô màu* sao cho hai miền kề nhau được tô hai màu khác nhau.

Như vậy, để đảm bảo chắc chắn hai miền kề nhau không bao giờ có màu trùng nhau, chúng ta tô mỗi miền bằng một màu khác nhau. Tuy nhiên rõ ràng việc làm đó nói chung là không hợp lý, vì chúng ta sẽ phải dùng đến rất nhiều màu nếu bản đồ có nhiều miền, cũng như rất khó phân biệt những màu gần giống nhau. Do vậy, người ta nghĩ đến chuyện chỉ dùng một số màu cần thiết để tô bản đồ. Từ đây, một bài toán được đặt ra là “xác định số màu tối thiểu cần có để tô màu đúng một bản đồ”.

Ví dụ 3.88. Xét bản đồ sau với 5 miền (hữu hạn) tất cả. Có thể thấy số màu tối thiểu để tô bản đồ là 3, chẳng hạn tô miền C: đỏ, miền A và D: xanh, miền B và E: vàng.

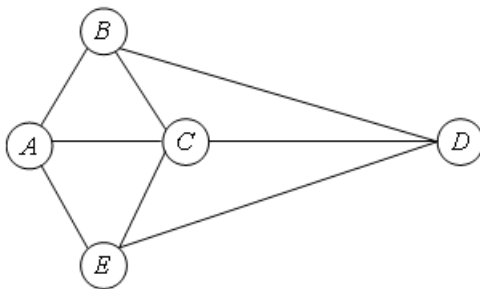


Rõ ràng, nếu chỉ dùng hai màu để tô bản đồ trên là không đủ.

3.12.2. Tô màu đồ thị

Mỗi bản đồ trên mặt phẳng có thể biểu diễn bằng một đồ thị, trong đó mỗi miền của bản đồ được biểu diễn bằng một đỉnh và các cạnh nối hai đỉnh nếu các miền được biểu diễn bằng hai đỉnh này là kề nhau. Đồ thị nhận được bằng cách này gọi là *đồ thị đối ngẫu* của bản đồ đang xét.

Ví dụ 3.89. Đồ thị đối ngẫu của bản đồ ở Ví dụ 3.88 là:



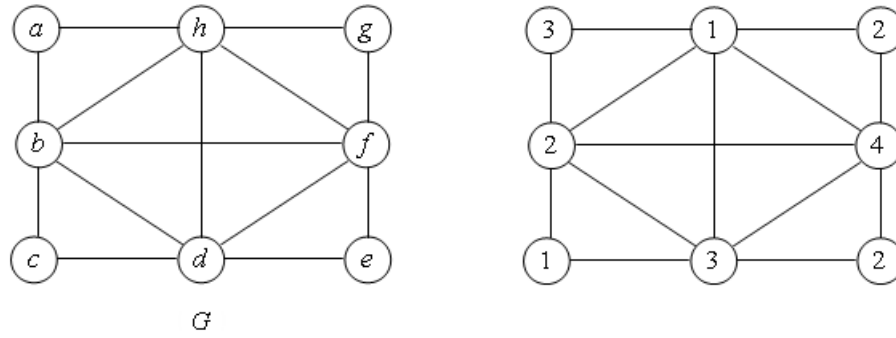
Rõ ràng, mọi bản đồ trên mặt phẳng đều có đồ thị đối ngẫu phẳng. Khi đó bài toán tô màu các miền của bản đồ tương đương với bài toán tô màu các đỉnh của đồ thị đối ngẫu sao cho không có hai đỉnh liền kề có cùng một màu. Một cách hình thức ta có định nghĩa sau.

Xét đồ thị $G = (V, E)$ và một số nguyên dương k . Ánh xạ $c : V \rightarrow \{1, 2, \dots, k\}$ được gọi là k -tô màu của G nếu với mỗi cặp đỉnh kề nhau u, v trong G ta luôn có $c(u) \neq c(v)$. Số k nhỏ nhất để G có k -tô màu được gọi là *sắc số* (chromatic number) của G và ký hiệu $\chi(G)$. Khi $\chi(G) = k$ thì G được gọi là k -sắc.

Như vậy sắc số của một đồ thị là số màu ít nhất cần dùng để tô các đỉnh của đồ thị, mỗi đỉnh một màu sao cho hai đỉnh kề nhau phải có màu khác nhau. Dễ thấy, một đồ thị tô màu được, tức là tồn tại k -tô màu, khi và chỉ khi nó không có khuyên. Ngoài ra sắc số của một đồ thị sẽ không vượt quá số đỉnh của nó, nghĩa là $1 \leq \chi(G) \leq n$ với $n = |V|$. Có thể thấy ngay đồ thị đầy đủ K_n có sắc số là n , đồ thị hai phía đầy đủ $K_{m,n}$ có sắc số là 2 và như vậy đồ thị hai phía bất kỳ có sắc số không vượt quá 2.

Ví dụ 3.90. Đồ thị G trong Ví dụ 3.89 có $\chi(G) = 3$.

Ví dụ 3.91. Xét đồ thị G sau:



Đồ thị G có 4 đỉnh b, d, f, h đôi một kề nhau, nên phải tô bằng 4 màu khác nhau. Do đó $\chi(G) \geq 4$. Ngoài ra có thể dùng 4 màu này để tô các đỉnh còn lại của G như hình bên. Như vậy tóm lại $\chi(G) = 4$.

Chúng ta có thể thấy mỗi cách tô màu cho đồ thị $G = (V, E)$ sẽ ứng với một cách phân hoạch tập đỉnh V thành các tập ổn định trong không giao nhau và mỗi tập ứng với một màu. Ngược lại mỗi cách phân hoạch tập đỉnh V thành các tập ổn định trong không giao nhau sẽ cho ta một cách tô màu.

Bổ đề 3.5. Mọi chu trình độ dài lẻ luôn có sắc số bằng 3.

Chứng minh. Giả sử chu trình có độ dài là $2n + 1$. Ta sẽ chứng minh bổ đề bằng quy nạp theo n . Trường hợp $n = 1$, chu trình sẽ gồm 3 đỉnh sao cho hai đỉnh bất kỳ đều kề nhau và do đó trong trường hợp này rõ ràng phải dùng đúng 3 màu để tô các đỉnh của chu trình.

Giả sử bổ đề đúng với trường hợp chu trình có độ dài $2k + 1$ ($2 \leq k \in \mathbb{N}$). Xét chu trình α có độ dài $2(k + 1) + 1 = 2k + 3$ với dãy các đỉnh $v_1, v_2, \dots, v_{2k+1}, v_{2k+2}, v_{2k+3}$. Khi đó nối v_1 với v_{2k+1} để được một chu trình β có độ dài $2k + 1$. Theo giả thiết quy nạp, chu trình β có sắc số bằng 3. Lúc này ta lấy màu của v_1 tô cho v_{2k+2} và màu của v_{2k+1} tô cho v_{2k+3} . Suy ra chu trình α đã được tô màu mà không thêm màu mới. Điều này có nghĩa sắc số của α bằng 3. $\square$

Mệnh đề 3.58. Nếu đồ thị G chứa một đồ thị con đồng phôi với đồ thị đầy đủ K_n thì $\chi(G) \geq n$.

Chứng minh. Gọi H là đồ thị con của G đồng phôi với K_n . Vì $\chi(K_n) = n$ nên suy ra $\chi(H) \geq n$. Do đó $\chi(G) \geq n$. $\square$

Sau đây là một tiêu chuẩn đơn giản kiểm tra xem một đơn đồ thị có sắc số bằng 2 hay không.

Định lý 3.59 (Konig). *Giả sử đơn đồ thị G có ít nhất một cạnh. Khi đó điều kiện cần và đủ để G có 2-sắc là G không chứa chu trình độ dài lẻ.*

Chứng minh. Giả sử G là đồ thị thỏa điều kiện bài toán và có $\chi(G) = 2$. Khi đó theo Bổ đề 3.5, G sẽ không chứa chu trình độ dài lẻ.

Ngược lại, không mất tính chất tổng quát có thể giả sử đồ thị G là liên thông và không chứa chu trình độ dài lẻ. Cố định một đỉnh bất kỳ a của G và tô nó bằng màu 1 trong hai màu 1 và 2. Khi đó với mỗi đỉnh v của G , tồn tại một đường đi từ a đến v . Nếu đường đi này có độ dài chẵn thì tô màu 1 cho v , còn nếu đường đi này có độ dài lẻ thì tô màu 2 cho v . Trường hợp nếu có hai đường đi mang tính chẵn lẻ khác nhau cùng nối a với v thì rõ ràng G phải chứa ít nhất một chu trình độ dài lẻ. Điều mâu thuẫn này cho thấy chỉ cần dùng đúng hai màu 1 và 2 để tô đồ thị G . $\square$

Hệ quả 3.16. *Mọi chu trình độ dài chẵn đều có sắc số bằng 2.*

Định lý 3.60. *Đồ thị vô hướng $G = (V, E)$ là k -sắc khi và chỉ khi G có hàm Grundy g thỏa*

$$\max_{v \in V} g(v) \leq k - 1.$$

Chứng minh. Nếu đồ thị G có hàm Grundy g thỏa điều kiện $\max_{v \in V} g(v) \leq k - 1$, thì chọn ngay g làm hàm k -tô màu cho G . Chứng minh ngược lại được dành làm bài tập. $\square$

Hệ quả 3.17. *Mọi đồ thị vô hướng không có khuyên đều có hàm Grundy và giá trị cực đại của các hàm này phải bằng nhau và bằng sắc số của đồ thị trừ đi 1.*

Mệnh đề 3.61. *Với mỗi số nguyên dương n , tồn tại một đồ thị không chứa K_3 và có sắc số bằng n .*

Sau đây là thuật toán hiệu quả tô màu một đơn đồ thị.

Thuật toán 3.13 (*Thuật toán Welch-Powell*)

Vào: Đơn đồ thị không có khuyên $G = (V, E)$ với n đỉnh.

Ra: Đồ thị G đã được tô màu.

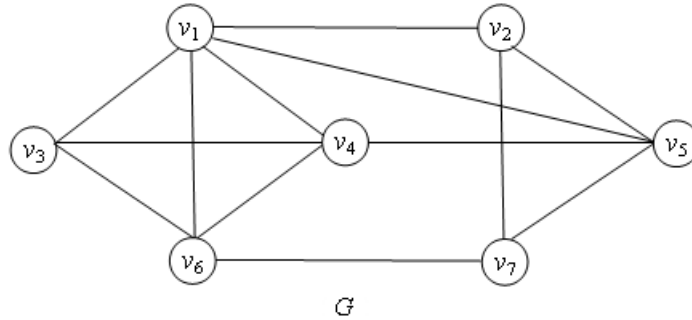
Phương pháp:

Bước 1. (*Khởi tạo*) Sắp thứ tự các đỉnh của V theo thứ tự không tăng của bậc, giả sử $v_1, v_2, \dots, v_n$.

Bước 2. (*Lập*) Gán màu 1 cho v_1 và các đỉnh khác (nếu tồn tại) trong danh sách trên sao cho nó không liên kề với v_1 . Nếu danh sách còn đỉnh chưa tô màu, thì ta lập lại với đỉnh đầu tiên của danh sách các đỉnh chưa tô màu này với màu tô là 2 giống như trên. Quá trình dừng khi tất cả các đỉnh đều đã được tô màu.

Thuật toán Welch-Powell là một cách hữu hiệu để tô một đồ thị G . Tuy nhiên, thuật toán này chỉ cung cấp một giới hạn trên cho $\chi(G)$, tức là thuật toán không phải luôn cung cấp một số màu tối thiểu để tô G . Việc tìm $\chi(G)$ nói chung là cực kỳ khó.

Ví dụ 3.92. Xét đồ thị G sau đây:



Danh sách các đỉnh của G theo thứ tự không tăng của bậc là $v_1, v_4, v_5, v_6, v_2, v_3, v_7$. Quá trình tô màu cho G theo Thuật toán Welch-Powell như sau:

- Gán màu 1 cho đỉnh thứ nhất v_1 . Đỉnh kế tiếp trong danh sách không nằm kề v_1 là v_7 , gán màu 1 cho v_7 .
- Đỉnh thứ nhất trong danh sách còn lại chưa được tô là v_4 , gán màu 2. Đỉnh kế tiếp trong danh sách chưa được tô này không kề với v_4 là v_2 , tô màu 2.
- Lặp lại quá trình này cho đến khi tất cả các đỉnh đều được tô.

Danh sách các đỉnh của G được gán màu là:

$$"v_1(1), v_4(2), v_5(3), v_6(3), v_2(2), v_3(4), v_7(1)".$$

Số ở trong dấu ngoặc tròn của danh sách trên là màu gán cho đỉnh đứng ngay trước. Ngoài ra, vì các đỉnh v_1, v_3, v_4, v_6 đôi một kề nhau, nên phải tô bằng các màu khác nhau. Do đó, có ít nhất 4 màu để tô G . Thuật toán Welch-Powell ở trên cũng chỉ sử dụng 4 màu để tô G . Do vậy, suy ra $\chi(G) = 4$.

Mệnh đề 3.62. Nếu đồ thị $G = (V, E)$ có n đỉnh và sắc số k thì $\alpha(G) \geq \frac{n}{k}$.

Chứng minh. Lập các tập đỉnh cùng màu:

$$I_j = \{v \in V : v \text{ tô màu } j\}, j = 0, 1, 2, \dots, k-1$$

là các tập ổn định trong của G . Ta có $|I_j| \leq \alpha(G)$. Suy ra

$$n = \sum_{j=0}^{k-1} |I_j| \leq k \cdot \alpha(G)$$

hay

$$\alpha(G) \geq \frac{n}{k}.$$

□

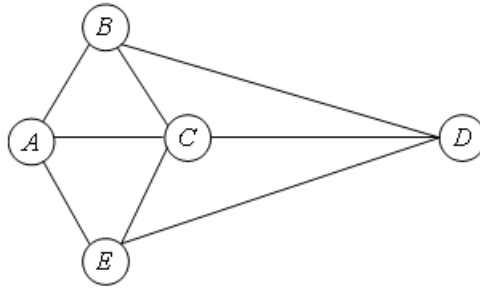
Mệnh đề 3.63. Nếu bậc lớn nhất của các đỉnh trong đồ thị G là d thì $\chi(G) \leq d+1$.

Chứng minh. Chứng minh bằng quy nạp theo số đỉnh n của G . Phần chứng minh này được dành làm bài tập. □

Bây giờ chúng ta tìm hiểu về vấn đề tô màu các cạnh của một đồ thị. Ánh xạ $c' : E \rightarrow \{1, 2, \dots, k\}$ được gọi là k -tô màu cạnh của G nếu với mỗi cặp cạnh liên thuộc với nhau e, f trong G ta luôn có $c'(e) \neq c'(f)$. Số k nhỏ nhất để G có k -tô màu cạnh được gọi là *sắc số cạnh* (edge chromatic number hay chromatic index) của G và ký hiệu $\chi'(G)$. Khi $\chi'(G) = k$ thì G được gọi là k -sắc cạnh.

Như vậy sắc số cạnh của một đồ thị là số màu ít nhất cần dùng để tô các cạnh của đồ thị, mỗi cạnh một màu sao cho hai cạnh liên thuộc cùng một đỉnh phải có màu khác nhau.

Ví dụ 3.93. Xét đồ thị G trong Ví dụ 3.89:



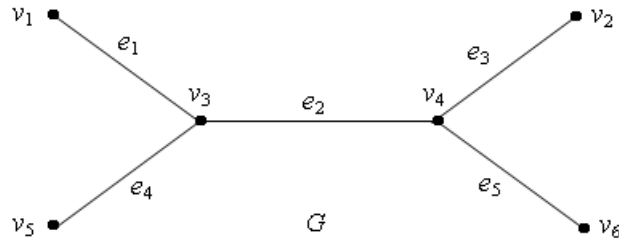
Chúng ta sẽ tô màu cạnh cho đồ thị này như sau: vì $\deg(C) = 4$, nên có bốn cạnh liên thuộc với đỉnh chung C . Suy ra phải dùng 4 màu để tô các cạnh này, chẳng hạn như tô màu 1 cho cạnh (A, C) , tô màu 2 cho cạnh (B, C) , tô màu 3 cho cạnh (D, C) và tô màu 4 cho cạnh (E, C) . Ngoài ra có thể dùng các màu này để tô các cạnh còn lại như sau: cạnh (A, B) tô màu 4, cạnh (B, D) tô màu 1, cạnh (D, E) tô màu 2 và cạnh (A, E) tô màu 3. Do đó $\chi'(G) = 4$. Như vậy số màu tối thiểu để tô các cạnh của G là 4.

Bài toán sắc số cạnh cũng có thể chuyển về bài toán sắc số như sau: với mỗi đồ thị $G = (V, E)$, xây dựng đồ thị $G' = (E, E')$ trong đó tập cạnh E' xác định bởi:

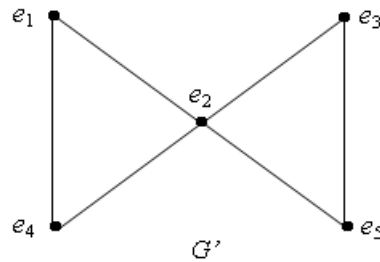
$$E' = \{(e, f) : e, f \in E \text{ và là hai cạnh liên thuộc với nhau}\}.$$

Khi đó dễ thấy sắc số cạnh của G bằng sắc số của G' .

Ví dụ 3.94. Xét đồ thị $G = (V, E)$ như sau:



Ta có đồ thị $G' = (E, E')$ là:

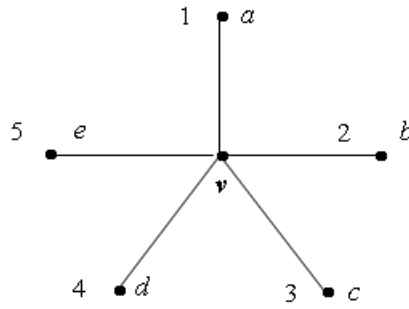


Rõ ràng $\chi(G') = \chi'(G) = 3$.

3.12.3. Tô màu đồ thị phẳng

Định lý 3.64 (Định lý 5 màu, Kempe & Heawood, 1890). *Mọi đồ thị phẳng không có khuyên có sắc số không lớn hơn 5.*

Chứng minh. Xét G là đồ thị thỏa điều kiện của định lý. Ta chứng minh định lý bằng quy nạp theo số đỉnh n của G . Rõ ràng các trường hợp $n = 1, 2, 3, 4, 5$ là hiển nhiên đúng. Giả sử định lý đúng với $n = k$ ($k \geq 6$). Ta chứng minh định lý cũng đúng với $n = k + 1$. Thật vậy theo Hệ quả 3.11, G có ít nhất một đỉnh v với bậc không lớn hơn 5. Xây dựng đồ thị G' từ G bằng cách bỏ đỉnh v . Theo giả thiết quy nạp ta có $\chi(G') \leq 5$. Lấy một cách tô màu nào đó của G' , nếu các đỉnh kề với v được tô bằng ít hơn 5 màu thì vẫn còn thừa màu để tô cho v , và khi đó $\chi(G) = \chi(G') \leq 5$. Do đó chỉ cần xét trường hợp đỉnh v kề với 5 đỉnh và các đỉnh kề với v được đánh số thứ tự theo chiều kim đồng hồ và tô bằng 5 màu như hình vẽ sau:



Hình 3.14: 5 đỉnh kề với 5 màu

Lúc này chúng ta sẽ đổi màu lại các đỉnh a, b, c, d, e và để dành ra một màu cho đỉnh v . Đầu tiên xét tất cả các đường đi trong G bắt đầu từ đỉnh a sao cho chỉ gồm các đỉnh tô màu 1 và màu 3. Trong các đường này nếu không có đường nào đi qua đỉnh c thì trao đổi màu 1 với màu 3 cho tất cả các đỉnh trên các đường đi đó. Sau đó lấy màu 1 tô cho đỉnh v . Ngược lại nếu có một đường đi từ đỉnh a đến đỉnh c gồm toàn các đỉnh được tô bằng hai màu 1 và 3 thì đường này cùng với hai cạnh (a, v) và (v, c) tạo thành một chu trình trong G . Do tính chất phẳng của G nên hai đỉnh b và d không thể cùng nằm bên trong hoặc cùng nằm bên ngoài chu trình này được. Suy ra không có đường đi nào nối b với d gồm các đỉnh chỉ tô bằng màu 2 và màu 4. Do đó trao đổi màu 2 và màu 4 cho tất cả các đỉnh trên các đường đi qua đỉnh b . Khi đó hai đỉnh b và d có cùng màu 4. Lấy màu 2 tô cho đỉnh v . Tóm lại chúng ta luôn lấy được 1 màu trong 5 màu của các đỉnh a, b, c, d, e để tô cho đỉnh v . $\square$

Sau đây là một trong các định lý nổi tiếng nhất trong toán học. Định lý bốn màu, định lý này đầu tiên được đưa ra như một phỏng đoán vào năm 1850 bởi một sinh viên người Anh tên là F. Guthrie và cuối cùng đã được hai nhà toán học Mỹ là Kenneth Appel và Wolfgang Haken chứng minh vào năm 1976. Trước năm 1976 cũng đã có nhiều chứng minh sai, mà thông thường rất khó tìm thấy chỗ sai, đã được công bố. Hơn thế nữa đã có nhiều cố gắng một cách vô ích để tìm phản thí dụ bằng cách cố vẽ bản đồ cần hơn bốn màu để tô nó. Có lẽ một trong những chứng minh sai nổi tiếng nhất trong toán học là chứng minh sai “bài toán bốn màu” được công bố năm 1879 bởi luật sư, nhà toán học nghiệp dư Luân Đôn tên là Alfred Kempe. Nhờ công bố lời giải của “bài toán bốn màu”, Kempe được công nhận là hội viên Hội Khoa học Hoàng gia Anh. Các nhà toán học chấp nhận cách chứng minh của ông ta cho tới 1890, khi Percy Heawood phát hiện ra sai lầm trong chứng minh của Kempe. Tuy nhiên dùng phương pháp của Kempe, Heawood đã chứng minh được “bài toán năm màu” (tức là mọi bản đồ có thể tô không quá năm màu). Như vậy Heawood mới giải được “bài toán năm màu”, còn “bài toán

bốn màu” vẫn còn đó và là một thách đố đối với các nhà toán học trong suốt gần một thế kỷ. Việc tìm lời giải của “bài toán bốn màu” đã ảnh hưởng đến sự phát triển theo chiều hướng khác nhau của lý thuyết đồ thị. Mãi đến năm 1976, khai thác phương pháp của Kempe và nhờ công cụ máy tính điện tử, Appel và Haken đã tìm ra lời giải của “bài toán bốn màu”. Chứng minh của họ dựa trên sự phân tích từng trường hợp một cách cẩn thận nhờ máy tính. Họ đã chỉ ra rằng nếu “bài toán bốn màu” là sai thì sẽ có một phản thí dụ thuộc một trong gần 2000 loại khác nhau và đã chỉ ra không có loại nào dẫn tới phản thí dụ cả. Trong chứng minh của mình họ đã dùng hơn 1000 giờ máy tính. Cách chứng minh này đã gây ra nhiều cuộc tranh cãi vì máy tính đã đóng vai trò quá quan trọng. Chẳng hạn liệu có thể có sai lầm trong chương trình và điều đó dẫn tới kết quả sai không? Lý luận của họ có thực sự là một chứng minh hay không, nếu nó phụ thuộc vào thông tin ra từ một máy tính không đáng tin cậy?

Định lý 3.65 (Định lý 4 màu, Appel & Haken, 1976). *Mọi đồ thị phẳng không có khuyên có sắc số không lớn hơn 4.*

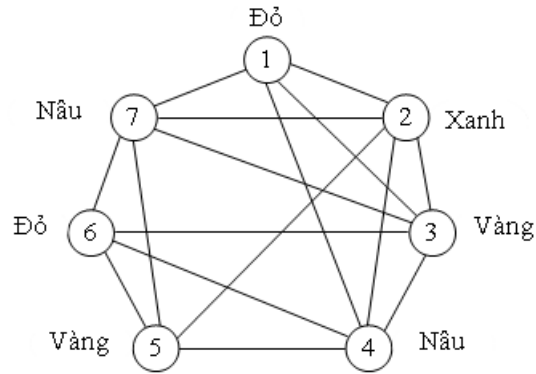
Biết rằng đồ thị đầy đủ K_n ($n \geq 5$) có sắc số lớn hơn 4, do đó bằng Định lý 4 màu chúng ta cũng suy ra được K_n , với $n \geq 5$, là đồ thị không phẳng.

3.12.4. Một số ứng dụng

Sau đây là một số ứng dụng của tô màu đồ thị.

Bài toán 3.5 (Lập lịch thi). Hãy lập lịch thi trong trường đại học sao cho không có sinh viên nào có hai môn thi cùng một lúc.

Bài toán này có thể giải bằng mô hình đồ thị, với các đỉnh là các môn thi, và có một cạnh nối hai đỉnh nếu có sinh viên phải thi cả hai môn được biểu diễn bằng hai đỉnh này. Thời gian thi của mỗi môn được biểu thị bằng các màu khác nhau. Như vậy việc lập lịch thi sẽ tương ứng với việc tô màu đồ thị này. Chẳng hạn, có 7 môn thi cần xếp lịch. Giả sử các môn học được đánh số từ 1 tới 7 và các cặp môn thi sau có chung sinh viên: 1 và 2, 1 và 3, 1 và 4, 1 và 7, 2 và 3, 2 và 4, 2 và 5, 2 và 7, 3 và 4, 3 và 6, 3 và 7, 4 và 5, 4 và 6, 5 và 6, 5 và 7, 6 và 7. Hình dưới đây biểu diễn đồ thị tương ứng. Khi đó việc lập lịch thi chính là việc tô màu đồ thị này. Vì số màu của đồ thị này là 4 nên cần có 4 đợt thi.



Hình 3.15: Đồ thị cho bài toán lập lịch

Bài toán 3.6 (Phân chia tần số). Các kênh truyền hình từ số 1 tới số 12 được phân chia cho các đài truyền hình sao cho không có đài phát nào cách nhau không quá 240 km lại dùng cùng một kênh. Có thể chia kênh truyền hình như thế nào bằng mô hình tô màu đồ thị?

Ta xây dựng đồ thị bằng cách coi mỗi đài phát là một đỉnh. Hai đỉnh được nối với nhau bằng một cạnh nếu chúng ở cách nhau không quá 240 km. Việc phân chia kênh tương ứng với việc tô màu đồ thị, trong đó mỗi màu biểu thị một kênh.

Chương 4. Đại số Boole

Đại số Boole được đề xuất bởi nhà toán học Claude Shannon¹ năm 1938 khi ông chứng tỏ rằng có thể dùng các qui tắc cơ bản của logic do George Boole đưa ra năm 1854 để thiết kế các mạch điện tử. Các qui tắc này tạo nên cơ sở của đại số Boole.

4.1. Khái niệm đại số Boole

4.1.1. Định nghĩa

*Đại số Boole*² là một tập hợp khác rỗng $\mathcal{A}$ cùng với ba phép toán ký hiệu nhân ($\cdot$), cộng ($+$) và lấy phần bù ($-$) thỏa các tiên đề sau với mọi $a, b, c \in \mathcal{A}$:

(1) *Tính giao hoán*:

(i) $a \cdot b = b \cdot a$

(ii) $a + b = b + a$

(2) *Tính kết hợp*:

(i) $(a \cdot b) \cdot c = a \cdot (b \cdot c)$

(ii) $(a + b) + c = a + (b + c)$

(3) *Tính phân phối*:

(i) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

(ii) $a + (b \cdot c) = (a + b) \cdot (a + c)$

(4) *Tính đồng nhất*: tồn tại hai phần tử khác nhau của $\mathcal{A}$, ký hiệu là 1 và 0, sao cho:

(i) $a \cdot 1 = 1 \cdot a = a$

(ii) $a + 0 = 0 + a = a$

1 và 0 tương ứng được gọi là *phần tử trung hoà* của phép $\cdot$ và $+$

(5) *Tính bù*: với mọi $a \in \mathcal{A}$, tồn tại duy nhất phần tử $\bar{a} \in \mathcal{A}$ sao cho:

(i) $a \cdot \bar{a} = \bar{a} \cdot a = 0$

(ii) $a + \bar{a} = \bar{a} + a = 1$

$\bar{a}$ được gọi là *phần tử bù* của a .

¹Claude Elwood Shannon (1916-2001) là nhà toán học, kỹ sư điện tử và mật mã học người Mỹ, được biết đến như là cha đẻ của lý thuyết thông tin.

²Còn đọc là Bun.

Ví dụ 4.1. *Đại số logic* là một đại số Boole, trong đó $\mathcal{A}$ là tập hợp các mệnh đề; các phép toán $\wedge$ (hội), $\vee$ (tuyển), $\neg$ (phủ định) tương ứng với $\cdot, +, -$; các hằng Đ (đúng), S (sai) tương ứng với các phần tử trung hoà 1, 0.

Ví dụ 4.2. *Đại số tập hợp* là một đại số Boole, trong đó $\mathcal{A}$ là tập hợp $\mathcal{P}(X)$ gồm các tập con của tập khác rỗng X ; các phép toán $\cap$ (giao), $\cup$ (hợp), $-$ (bù) tương ứng với $\cdot, +, -$; các tập $X, \emptyset$ tương ứng với các phần tử trung hoà 1, 0.

Ví dụ 4.3. Cho $\mathcal{B} = \{0, 1\}$ và các phép toán $\cdot, +, -$ trên $\mathcal{B}$ được định nghĩa như sau:

- (i) $1.1 = 1, 1.0 = 0, 0.1 = 0, 0.0 = 0$
- (ii) $1 + 1 = 1, 1 + 0 = 1, 0 + 1 = 1, 0 + 0 = 0$
- (iii) $\bar{1} = 0, \bar{0} = 1$.

Rõ ràng, $\mathcal{B}$ là một đại số Boole. Đây cũng chính là đại số logic, trong đó 1 và 0 tương ứng với Đ (đúng) và S (sai). Mỗi phần tử 0 và 1 của $\mathcal{B}$ còn được gọi là một bit.

Ví dụ 4.4. Cho $\mathcal{M}$ là tập hợp các số thực có cận trên a , cận dưới b và tâm đối xứng O . Các phép toán $\cdot, +, -$ trên $\mathcal{M}$ được định nghĩa như sau:

- (i) $a.b = \min(a, b)$
- (ii) $a + b = \max(a, b)$
- (iii) $\bar{a}$ là điểm đối xứng của a qua O .

Khi đó $\mathcal{M}$ là một đại số Boole, trong đó a và b tương ứng với các phần tử trung hoà 1 và 0.

Ở định nghĩa đại số Boole trên, các tiên đề của đại số Boole được xếp theo từng cặp (i) và (ii). Do đó, ta thấy từ mỗi tiên đề (i), nếu thay phép $\cdot$ bởi phép $+$, thay phép $+$ bởi phép $\cdot$, thay 1 bởi 0 và thay 0 bởi 1, thì ta thu được tiên đề (ii) tương ứng. Do đó, cặp tiên đề (i) và (ii) được gọi là *đối ngẫu* của nhau. Điều này có nghĩa, nếu ta chứng minh được một định lý trong đại số Boole thì ta có ngay một định lý khác, đối ngẫu của nó, bằng cách thay phép $\cdot$ và 1 tương ứng bởi phép $+$ và 0 (và ngược lại). Từ đây, chúng ta có nguyên lý đối ngẫu như sau.

Nguyên lý đối ngẫu: *đối ngẫu của một định lý là một định lý.*

4.1.2. Các hằng đẳng thức của đại số Boole

Từ định nghĩa của đại số Boole, chúng ta dễ dàng suy ra thêm được các tính chất cơ bản khác của các phép toán Boole.

Định lý 4.1.

(1) *Tính nuốt:* $a.0 = 0, a + 1 = 1$

(2) *Tính lũy đẳng:* $a.a = a, a + a = a$

(3) *Tính hút:* $a.(a + b) = a, a + (a.b) = a$

(4) *Tính bù kép:* $\overline{\overline{a}} = a$

(5) *Luật De Morgan:* $\overline{a.b} = \overline{a} + \overline{b}, \overline{a + b} = \overline{a}. \overline{b}$

(6) $\overline{1} = 0, \overline{0} = 1$.

Chứng minh.

(1) Theo các tiên đề (3)-(5) của định nghĩa đại số Boole, ta có:

$$\begin{aligned}
0 &= a.\overline{a} \\
&= a.(\overline{a} + 0) \\
&= a.\overline{a} + a.0 \\
&= 0 + a.0 \\
&= a.0
\end{aligned}$$

(2) Theo các tiên đề (3)-(5) của định nghĩa đại số Boole, ta có:

$$\begin{aligned}
a &= a.1 \\
&= a.(a + \overline{a}) \\
&= a.a + a.\overline{a} \\
&= a.a + 0 \\
&= a.a
\end{aligned}$$

(3) Theo các tiên đề (3), (4) của định nghĩa đại số Boole và tính nuốt ở trên, ta có:

$$\begin{aligned}
a.(a + b) &= (a + 0).(a + b) \\
&= a + (0.b) \\
&= a + 0 \\
&= a
\end{aligned}$$

(4) Suy ra ngay từ tiên đề (5) của định nghĩa đại số Boole.

(5) Theo tiên đề (1), (3)-(5) của định nghĩa đại số Boole và tính nuốt ở trên, ta có:

$$\begin{aligned}
 (a.b).(\bar{a} + \bar{b}) &= (a.b.\bar{a}) + (a.b.\bar{b}) \\
 &= (a.\bar{a}.b) + (a.b.\bar{b}) \\
 &= (0.b) + (a.0) \\
 &= 0 + 0 \\
 &= 0
 \end{aligned}$$

và

$$\begin{aligned}
 (a.b) + (\bar{a} + \bar{b}) &= (\bar{a} + \bar{b}) + (a.b) \\
 &= (\bar{a} + \bar{b} + a).(\bar{a} + \bar{b} + b) \\
 &= (\bar{a} + a + \bar{b}).(\bar{a} + \bar{b} + b) \\
 &= (1 + \bar{b}).(\bar{a} + 1) \\
 &= 1.1 \\
 &= 1
 \end{aligned}$$

Vì $a.b$ chỉ có một phần tử bù duy nhất, nên theo tiên đề (5) của định nghĩa đại số Boole ta có $\overline{a.b} = \bar{a} + \bar{b}$.

(6) Theo tính nuốt $1.0 = 0$ và $1 + 0 = 1$. Từ đây theo tiên đề (5) của định nghĩa đại số Boole ta có $\bar{1} = 0, \bar{0} = 1$. $\square$

Thật ra, bởi sự tương ứng của các phép toán $., +, \bar{}$ của đại số Boole với các phép toán logic, thì chúng ta có thể chuyển ngay các tính chất của các phép toán logic thành các tính chất của các phép toán Boole cũng như ngược lại.

Lưu ý 4.1. (a) Hệ tiên đề của đại số Boole nêu ở trên không phải là một hệ tối thiểu.

(b) Tương tự trong đại số logic, trong đại số Boole người ta cũng xét các công thức được thành lập từ các biến $a, b, c, \dots$ nhờ các phép toán $., +, \bar{}$. Trong các công thức này người ta quy ước thứ tự thực hiện các phép toán là $\bar{}, ., +$. Ngoài ra, $a.b$ còn được viết gọn lại là ab và đọc là *tích* của a và b , còn $a + b$ đọc là *tổng* của a và b . Chúng ta cũng có thể biến đổi công thức và rút gọn công thức tương tự trong đại số logic. Chúng ta cũng xét các tích sơ cấp và tổng sơ cấp tương tự hội sơ cấp và tuyển sơ cấp trong đại số logic. Do đó, mọi công thức đều có thể đưa về dạng tích chuẩn tắc hoàn toàn hoặc về dạng tổng chuẩn tắc hoàn toàn tương tự dạng hội và tuyển chuẩn tắc hoàn toàn trong đại số logic.

4.2. Hàm Boole

Cho $\mathcal{B} = \{0, 1\}$ và xét $\mathcal{B}^n = \{(x_1, x_2, \dots, x_n) | x_i \in \mathcal{B}, i = 1, 2, \dots, n\}$. Một biến x được gọi là *biến Boole* nếu nó chỉ nhận giá trị trong tập $\mathcal{B}$. Một hàm n biến Boole $F : \mathcal{B}^n \longrightarrow \mathcal{B}$ được gọi là một *hàm Boole* (hay *hàm đại số logic*, *hàm nhị phân*) *bậc n* và ký hiệu $F(x_1, x_2, \dots, x_n)$.

Các giá trị của hàm Boole cũng thường được cho qua bảng chân trị, là bảng chỉ chứa các giá trị 0 và 1, tương tự như bảng logic toán. Chẳng hạn, hàm Boole bậc 2 $F(x, y)$ với giá trị bằng 1 khi $x = y = 1$ và bằng 0 trong các trường hợp khác có thể được biểu diễn qua bảng như sau:

x	y	$F(x, y)$
1	1	1
1	0	0
0	1	0
0	0	0

Các hàm Boole cũng có thể được biểu diễn bằng cách dùng các biểu thức được tạo bởi các biến Boole và các phép toán Boole, các biểu thức như vậy được gọi là biểu thức Boole. Một cách hình thức *biểu thức Boole* với n biến $x_1, x_2, \dots, x_n$ được định nghĩa một cách đệ quy như sau:

(i) $0, 1, x_1, x_2, \dots, x_n$ là các biểu thức Boole.

(ii) Nếu A và B là các biểu thức Boole thì $\overline{A}, A.B, A + B$ cũng là các biểu thức Boole.

Người ta chứng minh được với mỗi hàm Boole bất kỳ, luôn tồn tại một biểu thức Boole tương ứng và ngược lại. Khi đó, giá trị của hàm Boole có thể nhận được bằng cách thay 0 và 1 cho các biến trong biểu thức Boole đó. Như vậy, các biểu thức Boole biểu diễn được hết các hàm Boole.

Ví dụ 4.5. Xét hàm Boole bậc 3 F cho dưới dạng biểu thức Boole như sau:

$$F(x, y, z) = xz + \overline{y}.$$

Giá trị của hàm Boole F có thể được tính qua bảng chân trị là

x	y	z	xz	$\overline{y}$	$F(x, y, z)$
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	1	0	1

Hai hàm Boole bậc n F và G được gọi là *bằng nhau* nếu $F(x_1, x_2, \dots, x_n) = G(x_1, x_2, \dots, x_n)$ với mọi $x_1, x_2, \dots, x_n \in \mathcal{B}$. *Phần bù* của hàm Boole F , ký hiệu $\overline{F}$, là hàm Boole bậc n xác định bởi:

$$\overline{F}(x_1, x_2, \dots, x_n) = \overline{F(x_1, x_2, \dots, x_n)}$$

Tổng Boole và *tích Boole* của hai hàm F và G , tương ứng ký hiệu $F + G$ và FG , là các hàm Boole bậc n xác định bởi:

$$(1) (F + G)(x_1, x_2, \dots, x_n) = F(x_1, x_2, \dots, x_n) + G(x_1, x_2, \dots, x_n)$$

$$(2) (FG)(x_1, x_2, \dots, x_n) = F(x_1, x_2, \dots, x_n) \cdot G(x_1, x_2, \dots, x_n).$$

Hai biểu thức Boole khác nhau biểu diễn cùng một hàm Boole thì được gọi là *tương đương* với nhau. Xét một biến Boole x và một giá trị $\sigma \in \{0, 1\}$. Ta định nghĩa x mũ σ , ký hiệu x^σ , là một biến Boole được xác định như sau:

$$x^\sigma = \begin{cases} x & \text{nếu } \sigma = 1 \\ \overline{x} & \text{nếu } \sigma = 0. \end{cases}$$

Từ định nghĩa dễ thấy $x^\sigma = 1 \Leftrightarrow x = \sigma$.

Tập đặc trưng của hàm Boole F , ký hiệu T_F , là tập các bộ n biến Boole $(x_1, x_2, \dots, x_n)$ sao cho $F(x_1, x_2, \dots, x_n) = 1$. Một cách hình thức:

$$T_F = \{(x_1, x_2, \dots, x_n) \in \mathcal{B}^n : F(x_1, x_2, \dots, x_n) = 1\}$$

Dễ kiểm chứng được tập đặc trưng T_F có các tính chất cơ bản sau:

Mệnh đề 4.2.

$$(1) T_{\overline{F}} = \overline{T_F}.$$

$$(2) T_{F+G} = T_F \cup T_G.$$

$$(3) T_{FG} = T_F \cap T_G.$$

Cho n biến Boole $x_1, x_2, \dots, x_n$. Một biểu thức có dạng $x_{i_1}^{\sigma_1} x_{i_2}^{\sigma_2} \dots x_{i_k}^{\sigma_k}$, trong đó $\sigma_1, \sigma_2, \dots, \sigma_k \in \mathcal{B}$ và $1 \leq i_1 < i_2 < \dots < i_k \leq n$, được gọi là một *tích sơ cấp* của n biến $x_1, x_2, \dots, x_n$. Số các biến xuất hiện trong một tích sơ cấp được gọi là *hạng* của tích sơ cấp đó. Nếu hàm Boole F bậc n được biểu diễn dưới dạng tổng của một số tích sơ cấp khác nhau của n biến, thì biểu diễn đó được gọi là *dạng tổng chuẩn tắc* của hàm F . Trường hợp nếu các tích sơ cấp trong dạng tổng chuẩn tắc của F đều có hạng n , thì dạng tổng chuẩn tắc này được gọi là *dạng tổng chuẩn tắc hoàn toàn* của F . Dạng tổng chuẩn tắc hoàn toàn của một hàm Boole là duy nhất.

Ví dụ 4.6. Xét hàm Boole $F(x, y)$ như sau:

$$F(x, y) = \begin{cases} 0 & \text{nếu } x = y \\ 1 & \text{ngược lại.} \end{cases}$$

Hàm Boole này thường được gọi là *hàm tuyến loại* và ký hiệu là $x \oplus y$. Dễ kiểm chứng được $x \oplus y = \bar{x}y + x\bar{y}$. Điều này có nghĩa tổng $\bar{x}y + x\bar{y}$ là một dạng tổng chuẩn tắc của hàm Boole F .

Mệnh đề 4.3. Mọi hàm Boole bậc n F đều có thể biểu diễn dưới dạng:

$$F(x_1, x_2, \dots, x_n) = \sum_{(\sigma_1, \dots, \sigma_i) \in \mathcal{B}^i} x_1^{\sigma_1} x_2^{\sigma_2} \dots x_i^{\sigma_i} F(\sigma_1, \dots, \sigma_i, x_{i+1}, \dots, x_n)$$

trong đó i là số tự nhiên bất kỳ, $1 \leq i \leq n$.

Cho $i = 1$ trong mệnh đề trên và nhận xét rằng vai trò của các biến x_i là như nhau, thì ta thu được hệ quả sau.

Hệ quả 4.1. Mọi hàm Boole bậc n F đều có thể được khai triển theo một biến x_i :

$$F(x_1, x_2, \dots, x_n) = \bar{x}_i F(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) + x_i F(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n).$$

Cho $i = n$ trong mệnh đề trên và bỏ đi các nhân tử bằng 1 trong tích, các số hạng bằng 0 trong tổng, ta cũng thu được hệ quả sau.

Hệ quả 4.2. Mọi hàm Boole bậc n F đều có thể được khai triển dưới dạng:

$$F(x_1, x_2, \dots, x_n) = \sum_{(\sigma_1, \sigma_2, \dots, \sigma_n) \in T_F} x_1^{\sigma_1} x_2^{\sigma_2} \dots x_n^{\sigma_n}$$

Hệ quả 4.2 cho thấy, mọi hàm Boole đều có thể biểu diễn dưới dạng tổng chuẩn tắc hoàn toàn. Như vậy, mọi hàm Boole đều có thể biểu diễn bằng một biểu thức Boole chỉ chứa ba phép toán tích, tổng và bù. Khi đó, ta nói hệ $\{tích, tổng, bù\}$ là đầy đủ.

Bằng đối ngẫu, chúng ta cũng có thể chứng minh một kết quả tương tự bằng cách thay tích bởi tổng và ngược lại, từ đó dẫn tới việc biểu diễn hàm Boole F qua một tích các tổng. Biểu diễn này được gọi là dạng *tích chuẩn tắc hoàn toàn* của F :

$$F(x_1, x_2, \dots, x_n) = \prod_{(\sigma_1, \sigma_2, \dots, \sigma_n) \in \overline{T_F}} (x_1^{\overline{\sigma_1}} + x_2^{\overline{\sigma_2}} + \dots + x_n^{\overline{\sigma_n}}).$$

Ví dụ 4.7. Xét hàm Boole bậc 3 $F(x, y, z)$ cho trong Ví dụ 4.5. Dạng tổng và tích chuẩn tắc hoàn toàn của F được tìm thấy như sau:

Theo bảng chân trị của F ta có:

$$F(0, 0, 0) = F(0, 0, 1) = F(1, 0, 0) = F(1, 0, 1) = F(1, 1, 1) = 1.$$

Do đó, dạng tổng chuẩn tắc hoàn toàn của F là:

$$F(x, y, z) = \overline{x} \overline{y} \overline{z} + \overline{x} \overline{y} z + x \overline{y} \overline{z} + x \overline{y} z + xyz.$$

Cũng từ bảng chân trị của F , ta có:

$$F(0, 1, 0) = F(0, 1, 1) = F(1, 1, 0) = 0.$$

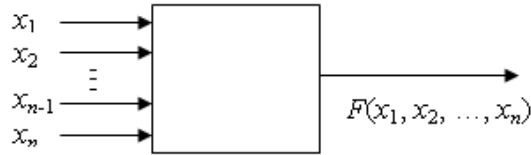
Suy ra, dạng tích chuẩn tắc hoàn toàn của F là:

$$F = (x + \overline{y} + z)(x + \overline{y} + \overline{z})(\overline{x} + \overline{y} + z).$$

4.3. Mạch logic

4.3.1. Cổng logic

Xét một thiết bị (Hình vẽ sau) có một số đường vào (dẫn tín hiệu vào) và chỉ có một đường ra (phát tín hiệu ra). Giả sử các tín hiệu vào $x_1, x_2, \dots, x_n$ (ta gọi là *đầu vào* hay *input*) và tín hiệu ra F (gọi là *đầu ra* hay *output*) đều chỉ có hai trạng thái khác nhau, tức là mang một bit thông tin, mà ta ký hiệu là 0 và 1.



Hình 4.1: Mạch logic một đường ra

Một thiết bị với các đầu vào và đầu ra mang giá trị 0, 1 như trên được gọi là một *mạch logic*. Đầu ra của một mạch logic là một hàm Boole F của các đầu vào $x_1, x_2, \dots, x_n$. Lúc đó, người ta nói mạch logic như vậy là *thực hiện* hàm F . Các mạch cơ sở tạo nên mạch logic được gọi là các *cổng logic*. Mỗi cổng logic này thực hiện một phép toán Boole. Chúng ta có 3 cổng logic cơ bản sau.

(1) Cổng NOT

Cổng NOT còn được gọi là *cổng KHÔNG* hay *bộ đảo*. Cổng này thực hiện hàm lấy phần bù, với đầu ra $F(x)$ là phần bù của đầu vào x .

$$F(x) = \bar{x} = \begin{cases} 0 & \text{nếu } x = 1 \\ 1 & \text{nếu } x = 0. \end{cases}$$



Hình 4.2: Cổng NOT

Chẳng hạn, nếu x là một chuỗi các bit 01011 thì cổng NOT cho ra chuỗi 10100.

(2) Cổng AND

Cổng AND còn được gọi là *cổng VÁ*. Cổng này thực hiện hàm tích, với đầu ra $F(x, y)$ là tích của các đầu vào x và y .

$$F(x, y) = xy = \begin{cases} 1 & \text{nếu } x = y = 1 \\ 0 & \text{ngược lại.} \end{cases}$$



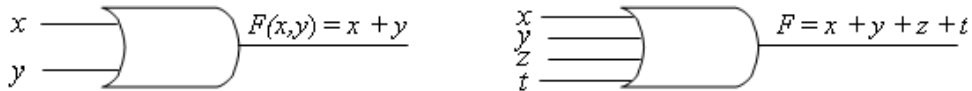
Hình 4.3: Cổng AND

Chẳng hạn, nếu x và y tương ứng là chuỗi các bit 01011 và 00110 thì cổng AND cho ra chuỗi 00010.

(3) Cổng OR

Cổng này còn được gọi là *cổng HOẶC*. Cổng này thực hiện hàm tổng, với đầu ra $F(x, y)$ là tổng của các đầu vào x và y .

$$F(x, y) = x + y = \begin{cases} 0 & \text{nếu } x = y = 0 \\ 1 & \text{ngược lại.} \end{cases}$$



Hình 4.4: Cổng OR

Chẳng hạn, nếu x và y tương ứng là xâu các bit 01011 và 00110 thì cổng OR cho ra xâu 01111.

4.3.2. Mạch logic

(1) Ở mục trên, chúng ta đã định nghĩa mạch logic (một đường ra). Mạch logic này là tổ hợp của các cổng logic, với chức năng thực hiện các hàm Boole phức tạp. Như chúng ta đã biết, một hàm Boole bất kỳ có thể biểu diễn bằng một biểu thức chỉ chứa các phép $-$, $.$, $+$. Từ đó suy ra có thể lắp ghép thích hợp các cổng NOT, AND, OR để tạo nên một mạch logic thực hiện một hàm Boole bất kỳ.

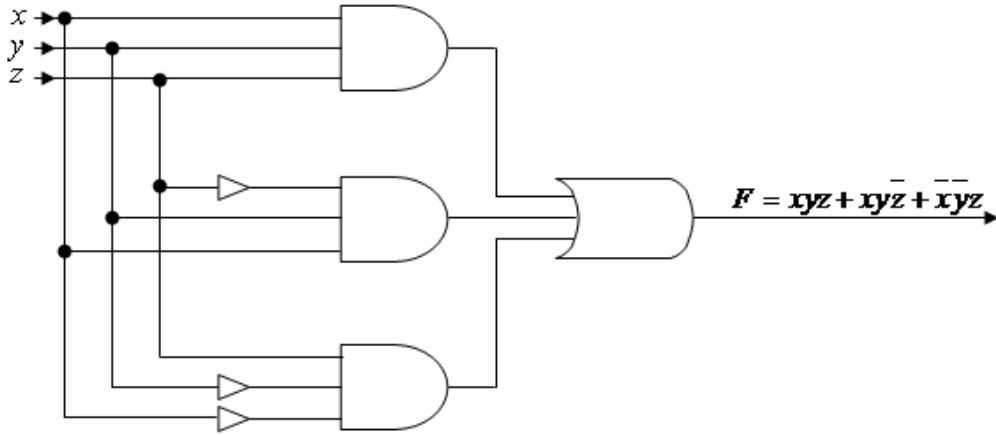
Ví dụ 4.8. Xây dựng một mạch logic thực hiện hàm Boole cho bởi bảng sau.

x	y	z	$F(x, y, z)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Từ bảng trên, ta thấy $F(1, 1, 1) = F(1, 1, 0) = F(0, 0, 1) = 1$ và $F(0, 0, 0) = F(0, 1, 0) = F(0, 1, 1) = F(1, 0, 0) = F(1, 0, 1) = 0$. Do đó, hàm F có dạng tổng chuẩn tắc hoàn toàn là:

$$F(x, y, z) = xyz + xy\bar{z} + \bar{x}\bar{y}z.$$

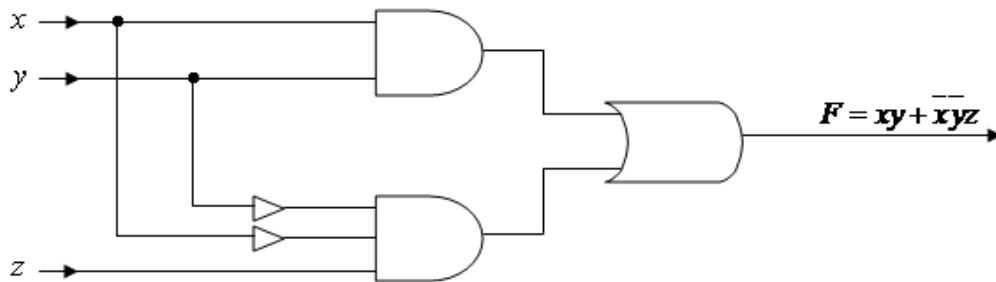
Suy ra, mạch logic thực hiện hàm F sẽ là:



Tuy nhiên, có thể thấy biểu thức của hàm $F(x, y, z)$ ở trên có thể rút gọn lại là:

$$\begin{aligned} F(x, y, z) &= xyz + xy\bar{z} + \bar{x}\bar{y}z \\ &= xy(z + \bar{z}) + \bar{x}\bar{y}z \\ &= xy + \bar{x}\bar{y}z. \end{aligned}$$

Khi đó, mạch logic thực hiện hàm $F(x, y, z) = xy + \bar{x}\bar{y}z$ sẽ là:



Như vậy, hai mạch logic ở trên cùng thực hiện cùng một hàm Boole, khi đó người ta nói hai mạch logic này là *tương đương*. Tuy nhiên, có thể thấy mạch logic thứ hai đơn giản hơn. Do đó, vấn đề đặt ra đi tìm mạch logic đơn giản thực hiện hàm F . Rõ ràng, vấn đề này là gắn liền với vấn đề tìm biểu thức đơn giản nhất biểu diễn hàm F đó. Đây là vấn đề có ý nghĩa và khó.

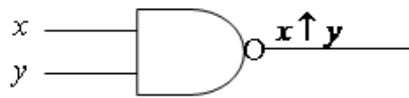
Lưu ý 4.2. Ở trên, chúng ta vừa xét việc thực hiện một hàm Boole bất kỳ bằng một mạch logic chỉ gồm các cổng NOT, AND và OR. Tuy nhiên, bằng luật De Morgan ta có $x + y = \overline{\bar{x}\bar{y}}$ và $xy = \overline{\bar{x} + \bar{y}}$, suy ra phép toán $+$ có thể biểu diễn qua

phép toán $\neg$, $\cdot$ và phép toán $+$ có thể biểu diễn qua phép toán $\neg$, $+$. Do đó hệ $\{\neg, \cdot\}$ hoặc hệ $\{\neg, +\}$ cũng là các *hệ đầy đủ* (để biểu diễn tất cả các biểu thức logic). Như vậy, chúng ta có thể thực hiện một hàm Boole bất kỳ bằng một mạch logic chỉ gồm có các cổng NOT, AND hoặc NOT, OR.

Ví dụ 4.9. Xét hàm *Sheffer*:

$$F(x, y) = x \uparrow y = \begin{cases} 0 & \text{nếu } x = y = 1 \\ 1 & \text{ngược lại.} \end{cases}$$

Khi đó, mạch logic thực hiện hàm $\uparrow$ được gọi là cổng NAND và được cho bởi hình dưới đây:



Hình 4.5: Cổng NAND

Vì $x \uparrow y = \overline{xy}$, nên có thể xem cổng NAND như là nối tiếp của hai cổng AND và NOT như hình vẽ dưới đây:



Hình 4.6: Cổng NAND biểu diễn qua hai cổng AND và NOT

Hơn nữa, ta có

$$\bar{x} = x \uparrow x$$

$$xy = \overline{x \uparrow y} = (x \uparrow y) \uparrow (x \uparrow y)$$

$$x + y = \overline{\bar{x} \bar{y}} = \bar{x} \uparrow \bar{y} = (x \uparrow x) \uparrow (y \uparrow y).$$

Do đó, hệ $\{\uparrow\}$ cũng là đầy đủ, suy ra bất kỳ một hàm Boole nào cũng có thể thực hiện được bằng một mạch logic chỉ gồm có cổng NAND.

Ví dụ 4.10. Xét hàm *Vebb* như sau:

$$F(x, y) = x \downarrow y = \begin{cases} 0 & \text{nếu } x = 1 \text{ hoặc } y = 1 \\ 1 & \text{ngược lại.} \end{cases}$$

Khi đó, mạch logic thực hiện hàm $\downarrow$ được gọi là cổng NOR, và được cho bởi hình dưới đây:



Hình 4.7: Cổng NOR

Vì $x \downarrow y = \overline{x + y}$, nên có thể xem cổng NOR như là nối tiếp của hai cổng OR và NOT như hình vẽ dưới đây:



Hình 4.8: Cổng NOR biểu diễn qua hai cổng OR và NOT

Lập luận tương tự như trên, ta có:

$$\bar{x} = x \downarrow x$$

$$x + y = \overline{x \downarrow y} = (x \downarrow y) \downarrow (x \downarrow y)$$

$$xy = \overline{\bar{x} + \bar{y}} = \bar{x} \downarrow \bar{y} = (x \downarrow x) \downarrow (y \downarrow y).$$

Suy ra, hệ $\{\downarrow\}$ cũng là đầy đủ. Do đó, bất kỳ một hàm Boole nào cũng có thể thực hiện được bằng một mạch logic chỉ gồm có cổng NOR.

Xét phép toán logic *tuyển loại* $\oplus$ (còn gọi là *tuyển chặt*). Theo định nghĩa của $\oplus$, ta có hàm

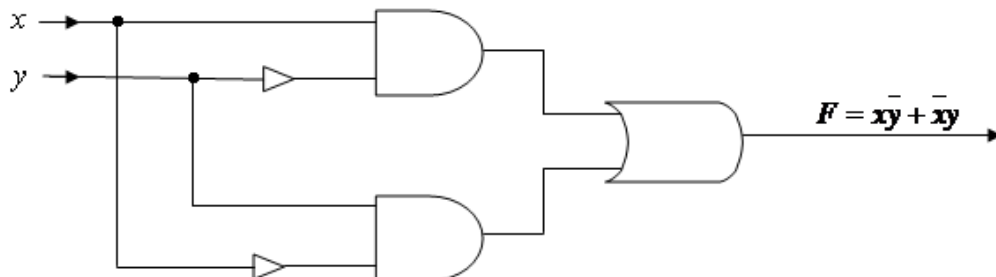
$$F(x, y) = x \oplus y = \begin{cases} 0 & \text{nếu } x = y \\ 1 & \text{ngược lại.} \end{cases}$$

Hàm này có mạch logic sau, thường được gọi là cổng XOR hay cổng OR-loại:



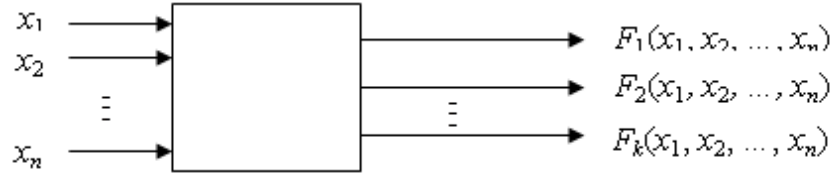
Hình 4.9: Cổng XOR

Vì $x \oplus y = x\bar{y} + \bar{x}y$, nên cổng này thường được vẽ lại là:



Hình 4.10: Cổng XOR vẽ lại

(2) Nhiều bài toán đòi hỏi phải xây dựng những *mạch logic có nhiều đường ra*, cho các đầu ra $F_1, F_2, \dots, F_k$ là các hàm Boole của các đầu vào $x_1, x_2, \dots, x_n$.



Hình 4.11: *Mạch logic có nhiều đường ra*

Chẳng hạn, ta xét phép cộng hai số tự nhiên được viết trong hệ nhị phân. Ở đây một số có n chữ số viết trong hệ nhị phân $\overline{a_n a_{n-1} \dots a_1}$ được gọi là *số n -bit*. Đầu tiên xét phép cộng hai số 1-bit x, y . Ta có bảng cộng hai số này như sau:

$$\begin{aligned} 1 + 1 &= 10 \\ 1 + 0 &= 01 \\ 0 + 1 &= 01 \\ 0 + 0 &= 00 \end{aligned}$$

Bảng 4.1.: *Bảng cộng hai số 1-bit*

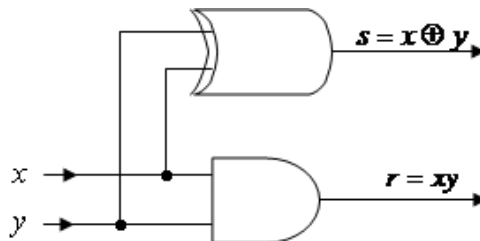
Tổng $x + y$ là một số 2-bit $\overline{rs}$: $x + y = \overline{rs}$, với s và r lần lượt là *bit tổng* và *bit nhớ* của $x + y$. Khi đó, các hàm Boole s và r của x, y được xác định từ bảng trên là:

x	y	r	s
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	0

Bảng 4.2.: *Bảng các hàm Boole s, r của x, y*

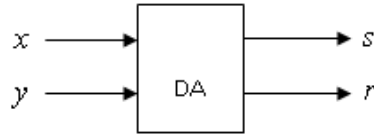
Từ bảng ngay trên, rõ ràng ta thấy: $s = x \oplus y$ và $r = x.y$

Khi đó, mạch logic thực hiện hai hàm $s = x \oplus y$ và $r = x.y$ là:



Hình 4.12: *Mạch logic thực hiện hai hàm $s = x \oplus y$ và $r = x.y$*

Mạch này được gọi là *mạch cộng hai số 1-bit* hay *mạch cộng bán phần*, ký hiệu DA, được vẽ như sau:



Hình 4.13: *Mạch cộng hai số 1-bit*

Bây giờ, ta xét phép cộng hai số 2-bit $\overline{x_2x_1}$ và $\overline{y_2y_1}$:

$$\begin{array}{r} x_2 \quad x_1 \\ + \\ y_2 \quad y_1 \end{array}$$

Phép cộng thực hiện theo từng cột, ở cột thứ nhất (từ phải sang trái) ta tính $x_1 + y_1$ được bit tổng s_1 và bit nhớ c_1 , ở cột thứ hai ta tính $x_2 + y_2 + c_1$, tức là phải cộng ba số 1-bit.

...

4.4. Cực tiểu hóa các mạch logic

Hiệu quả của một mạch logic phụ thuộc vào số các cổng và sự bố trí các cổng đó. Quá trình thiết kế một mạch logic được bắt đầu bằng một bảng chỉ rõ các giá trị đầu ra đối với mỗi một tổ hợp các giá trị đầu vào. Chúng ta luôn có thể sử dụng khai triển tổng các tích của mạch để tìm tập các cổng logic thực hiện mạch đó. Tuy nhiên, khai triển tổng các tích có thể chứa các số hạng nhiều hơn mức cần thiết. Các số hạng trong khai triển tổng các tích chỉ khác nhau ở một biến sao cho trong số hạng này xuất hiện biến đó và trong số hạng kia xuất hiện phần bù của nó, đều có thể được tổ hợp lại. Chẳng hạn, xét mạch có đầu ra bằng 1 khi và chỉ khi $x = y = z = 1$, hoặc $x = z = 1$ và $y = 0$. Khi đó, khai triển tổng các tích của mạch này là $xyz + x\bar{y}z$. Rõ ràng, hai tích trong khai triển này chỉ khác nhau ở một biến y . Suy ra, chúng có thể tổ hợp lại như sau:

$$xyz + x\bar{y}z = (y + \bar{y})xz = 1xz = xz.$$

Do đó, xz là biểu thức với ít phép toán hơn biểu diễn mạch đã cho. Mạch thứ hai chỉ dùng một cổng, trong khi mạch thứ nhất phải dùng ba cổng và một bộ đảo.

4.4.1. Phương pháp bản đồ Karnaugh

Phương pháp này được đề xuất bởi Maurice Karnaugh vào những năm 1953, dựa vào ý tưởng tìm các số hạng trong một biểu thức Boole để tổ hợp lại, nhằm rút gọn các khai triển tổng các tích. Đây là phương pháp dựa trên việc rà soát trực quan để nhận dạng các số hạng cần được nhóm lại, do đó chỉ thích hợp với các hàm Boole có số biến tương đối nhỏ, không quá bốn. Cũng vì cách làm là trực quan nên phương pháp này thực sự khó cài đặt. Bây giờ, chúng ta sẽ đi vào xét các hàm Boole với số biến lần lượt là 2, 3 và 4.

(1) Đầu tiên, xét hàm Boole 2 biến $F(x, y)$. Rõ ràng, có tất cả bốn tích sơ cấp khác nhau trong khai triển tổng các tích của hàm $F(x, y)$. Khi đó, một bảng chia làm bốn ô biểu diễn cho các tích sơ cấp trong khai triển tổng các tích của hàm $F(x, y)$ sao cho các ô biểu diễn tích sơ cấp có mặt trong khai triển ghi số 1 được gọi là *bản đồ Karnaugh đối với hàm $F(x, y)$* hay *bản đồ Karnaugh hai biến*. Một trong các cách để lập bản đồ Karnaugh hai biến của hàm $F(x, y)$ được cho trong hình dưới đây:

	y	$\bar{y}$
x	xy	$x\bar{y}$
$\bar{x}$	$\bar{x}y$	$\bar{x}\bar{y}$

Hình 4.14: *Bản đồ Karnaugh hai biến*

Hai ô trong bản đồ Karnaugh của hàm Boole F (hai biến, ba biến, ...) được gọi là *kề nhau* nếu các tích sơ cấp mà chúng biểu diễn chỉ khác nhau một biến. Để rút gọn khai triển tổng các tích hai biến bằng phương pháp bản đồ Karnaugh, chúng ta cần nhận dạng các tích sơ cấp có thể tổ hợp lại. Trường hợp thứ nhất, đó là các khối gồm 2 ô kề nhau biểu diễn các tích sơ cấp có thể được tổ hợp lại thành một biến. Trường hợp còn lại là các khối 2×2 biểu diễn các tích sơ cấp có thể tổ hợp lại thành một tích không có một biến nào, cụ thể là biểu thức 1.

Ví dụ 4.11. Cực tiểu hóa các hàm Boole 2 biến $F(x, y)$ sau đây bằng phương pháp bản đồ Karnaugh:

(a) $F(x, y) = xy + \bar{x}y.$

(b) $F(x, y) = x\bar{y} + \bar{x}y.$

(c) $F(x, y) = x\bar{y} + \bar{x}y + \bar{x}\bar{y}.$

Ta có, ba bản đồ Karnaugh hai biến tương ứng cho ba hàm Boole ở trên là:

	y	$\bar{y}$		y	$\bar{y}$		y	$\bar{y}$
x	1		x		1	x		1
$\bar{x}$	1		$\bar{x}$	1		$\bar{x}$	1	1
	(a)			(b)			(c)	

Hình 4.15: Ba bản đồ Karnaugh hai biến

Nhóm các tích sơ cấp được chỉ ra trong hình sau:

	y	$\bar{y}$		y	$\bar{y}$		y	$\bar{y}$
x	1		x		1	x		1
$\bar{x}$	1		$\bar{x}$	1		$\bar{x}$	1	1
	(a)			(b)			(c)	

Hình 4.16: Nhóm các tích sơ cấp của các hàm biểu diễn trong bản đồ Karnaugh

Vậy, khai triển cực tiểu của các hàm Boole trên tương ứng là:

- (a) $F(x, y) = y.$
- (b) $F(x, y) = x\bar{y} + \bar{x}y.$
- (c) $F(x, y) = \bar{x} + \bar{y}.$

(2) Tiếp theo, xét hàm Boole 3 biến $F(x, y, z)$. Có 8 tích sơ cấp khác nhau trong khai triển tổng các tích của hàm $F(x, y, z)$. Do đó, bản đồ Karnaugh ba biến cho hàm $F(x, y, z)$ là một bảng gồm 8 ô, mỗi ô biểu diễn một tích sơ cấp có thể có của hàm $F(x, y, z)$. Một trong các cách để lập bản đồ Karnaugh ba biến của hàm $F(x, y, z)$ được cho trong hình dưới đây:

	yz	$y\bar{z}$	$\bar{y}z$	$\bar{y}\bar{z}$
x	xyz	$xy\bar{z}$	$x\bar{y}z$	$x\bar{y}\bar{z}$
$\bar{x}$	$\bar{x}yz$	$\bar{x}y\bar{z}$	$\bar{x}\bar{y}z$	$\bar{x}\bar{y}\bar{z}$

Hình 4.17: Bản đồ Karnaugh ba biến

Để rút gọn khai triển tổng các tích ba biến, ta sẽ dùng bản đồ Karnaugh ba biến trên để nhận dạng các tích sơ cấp có thể tổ hợp lại. Trường hợp thứ nhất, đó là các khối gồm 2 ô kề nhau biểu diễn các tích sơ cấp có thể được tổ hợp lại thành một tích của hai biến. Trường hợp thứ hai là các khối 2×2 và 4×1 biểu diễn các tích sơ cấp có thể tổ hợp lại thành một biến duy nhất. Cuối cùng là một khối gồm tất cả 8 ô biểu diễn các tích có thể tổ hợp lại thành một tích không có một biến nào, cụ thể là biểu thức 1.

Ví dụ 4.12. Cực tiểu hóa các hàm Boole 3 biến $F(x, y, z)$ sau đây bằng phương pháp bản đồ Karnaugh:

(a) $F(x, y, z) = xy\bar{z} + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}\bar{z}.$

(b) $F(x, y, z) = x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}.$

(c) $F(x, y, z) = xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z} + \bar{x}yz + \bar{x}\bar{y}z + \bar{x}\bar{y}\bar{z}.$

Ta có, bản đồ Karnaugh ba biến cho các hàm Boole trên tương ứng là:

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$		yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$		yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x		1	1		x			1	1	x	1	1	1	1
$\bar{x}$	1		1		$\bar{x}$	1		1	1	$\bar{x}$	1		1	1

Nhóm các tích sơ cấp được chỉ ra trong hình sau:

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$		yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$		yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
x		1	1		x			1	1	x	1	1	1	1
$\bar{x}$	1		1		$\bar{x}$	1		1	1	$\bar{x}$	1		1	1

Do đó, khai triển cực tiểu của các hàm Boole trên lần lượt là:

(a) $F(x, y, z) = x\bar{z} + \bar{y}\bar{z} + \bar{x}yz.$

(b) $F(x, y, z) = \bar{y} + \bar{x}z.$

(c) $F(x, y, z) = x + \bar{y} + z.$

(3) Cuối cùng, xét hàm Boole 4 biến $F(w, x, y, z)$. Có tất cả 16 tích sơ cấp khác nhau trong khai triển tổng các tích của hàm $F(w, x, y, z)$. Do đó, *bản đồ Karnaugh bốn biến* cho hàm $F(w, x, y, z)$ là một bảng gồm 16 ô, mỗi ô biểu diễn một tích sơ cấp có thể có của hàm $F(w, x, y, z)$. Một trong những cách lập bản đồ Karnaugh 4 biến của hàm $F(w, x, y, z)$ được cho trong hình dưới đây:

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	$wxyz$	$wxy\bar{z}$	$wx\bar{y}\bar{z}$	$wx\bar{y}z$
$w\bar{x}$	$w\bar{x}yz$	$w\bar{x}y\bar{z}$	$w\bar{x}\bar{y}\bar{z}$	$w\bar{x}\bar{y}z$
$\bar{w}x$	$\bar{w}xyz$	$\bar{w}xy\bar{z}$	$\bar{w}x\bar{y}\bar{z}$	$\bar{w}x\bar{y}z$
$\bar{w}\bar{x}$	$\bar{w}\bar{x}yz$	$\bar{w}\bar{x}y\bar{z}$	$\bar{w}\bar{x}\bar{y}\bar{z}$	$\bar{w}\bar{x}\bar{y}z$

Hình 4.18: *Bản đồ Karnaugh bốn biến*

Lưu ý rằng, trong bản đồ Karnaugh này mỗi một ô kề với bốn ô khác. Khi đó, sự rút gọn một khai triển tổng các tích bốn biến được thực hiện bằng cách nhận dạng các khối gồm 2, 4, 8 hoặc 16 ô biểu diễn các tích sơ cấp có thể tổ hợp lại được. Mỗi ô biểu diễn một tích sơ cấp hoặc được dùng để lập một tích có ít biến hơn hoặc được đưa vào trong khai triển. Cũng như trong trường hợp bản đồ Karnaugh hai và ba biến, mục tiêu là cần phải nhận dạng các khối lớn nhất có chứa các số 1 bằng cách dùng một số ít nhất các khối, mà trước hết là các khối lớn nhất.

Ví dụ 4.13. Cực tiểu hóa các hàm Boole 4 biến $F(w, x, y, z)$ sau đây bằng phương pháp bản đồ Karnaugh:

(a) $F(w, x, y, z) = wxyz + wxy\bar{z} + wx\bar{y}\bar{z} + w\bar{x}yz + w\bar{x}\bar{y}z + w\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}z + \bar{w}\bar{x}yz + \bar{w}\bar{x}\bar{y}\bar{z}.$

(b) $F(w, x, y, z) = wxy\bar{z} + w\bar{x}yz + w\bar{x}\bar{y}\bar{z} + w\bar{x}\bar{y}z + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}yz + \bar{w}\bar{x}\bar{y}\bar{z}.$

(c) $F(w, x, y, z) = wxy\bar{z} + wxy\bar{z} + w\bar{x}yz + w\bar{x}\bar{y}\bar{z} + w\bar{x}\bar{y}z + \bar{w}x\bar{y}z + \bar{w}x\bar{y}\bar{z} + \bar{w}\bar{x}\bar{y}\bar{z} + \bar{w}\bar{x}\bar{y}z + \bar{w}\bar{x}\bar{y}\bar{z}.$

Bản đồ Karnaugh bốn biến cho các hàm Boole trên tương ứng là:

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$		yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$		yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1	1	1		wx			1		wx		1	1	
$w\bar{x}$	1		1	1	$w\bar{x}$	1	1	1		$w\bar{x}$	1	1	1	
$\bar{w}\bar{x}$	1	1			$\bar{w}\bar{x}$		1	1		$\bar{w}\bar{x}$		1	1	
$\bar{w}x$				1	$\bar{w}x$			1		$\bar{w}x$	1	1	1	1

Nhóm các tích sơ cấp được chỉ ra trong hình sau:

	yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$		yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$		yz	$y\bar{z}$	$\bar{y}\bar{z}$	$\bar{y}z$
wx	1	1	1		wx			1		wx		1	1	
$w\bar{x}$	1		1	1	$w\bar{x}$	1	1	1		$w\bar{x}$	1	1	1	
$\bar{w}\bar{x}$	1	1			$\bar{w}\bar{x}$		1	1		$\bar{w}\bar{x}$		1	1	
$\bar{w}x$				1	$\bar{w}x$			1		$\bar{w}x$	1	1	1	1

Suy ra, khai triển cực tiểu của các hàm Boole trên lần lượt là:

(a) $F(w, x, y, z) = wyz + wx\bar{z} + w\bar{x}\bar{y} + \bar{w}\bar{y}z + \bar{w}x\bar{y}z.$

(b) $F(w, x, y, z) = \bar{y}\bar{z} + w\bar{x}y + \bar{x}y\bar{z}.$

(c) $F(w, x, y, z) = \bar{z} + \bar{w}x + w\bar{x}y.$

4.4.2. Phương pháp Quine-McCluskey

Phương pháp Quine-McCluskey được đề xuất vào những năm 1956 bởi W. V. Quine và Edward J. McCluskey. Phương pháp này bao gồm hai phần. Phần đầu tìm các số hạng là ứng viên để đưa vào khai triển cực tiểu như một tổng các tích Boole mà ta gọi là các nguyên nhân nguyên tố, phần thứ hai xác định xem trong số các ứng viên đó ứng viên nào là thực sự dùng được. So với phương pháp Karnaugh, phương pháp này hữu hiệu hơn nhiều khi giải quyết hàm Boole nhiều hơn bốn biến. Tuy nhiên, bài toán cực tiểu hàm Boole được biết là bài toán NP-khó, nên Thuật toán Quine-McCluskey vẫn có độ phức tạp là hàm mũ theo số biến.

Đầu tiên chúng ta cần một số khái niệm cơ bản sau. Cho hai hàm Boole F và G bậc n . Ta nói G là một *nguyên nhân* của F nếu $T_G \subseteq T_F$, nghĩa là $G \Rightarrow F$ là một

hằng đúng. Như vậy, rõ ràng mỗi tích sơ cấp trong một dạng tổng chuẩn tắc của F là một nguyên nhân của F . Khi đó, mỗi tích sơ cấp A của F được gọi là một *nguyên nhân nguyên tố* của F nếu trong A xoá đi một biến bất kỳ thì tích sơ cấp nhận được không còn là nguyên nhân của F .

Như vậy, nếu $F_1, F_2, \dots, F_k$ là các nguyên nhân của F thì $\sum_{i=1}^k F_i$ cũng là một nguyên nhân của F , vì $T_{F_i} \subseteq T_F$ với mọi $i = 1, 2, \dots, k$, và do đó

$$T_{\sum_{i=1}^k F_i} = \bigcup_{i=1}^k T_{F_i} \subseteq T_F.$$

Giả sử S là một tập các nguyên nhân của F . Ta nói S là *hệ đầy đủ* đối với F nếu $F = \sum_{G \in S} G$, nghĩa là $T_F = \bigcup_{G \in S} T_G$. Người ta chứng minh được kết quả quan trọng sau:

Mệnh đề 4.4. *Tập tất cả các nguyên nhân nguyên tố của hàm F là một hệ đầy đủ đối với F .*

Khi đó, với S là tập tất cả các nguyên nhân nguyên tố của hàm F thì dạng tổng chuẩn tắc $\sum_{G \in S} G$ của F được gọi là *dạng tổng chuẩn tắc thu gọn* của F . Thuật toán tìm dạng tổng chuẩn tắc thu gọn của hàm F được trình bày ngay trong mục sau.

4.4.2.1. Tìm dạng tổng chuẩn tắc thu gọn

Xét n biến Boole $x_1, x_2, \dots, x_n$. Ta sẽ biểu diễn mỗi tích sơ cấp của n biến này bằng một dãy n ký hiệu trong tập $\{0, 1, -\}$ theo quy ước: ký tự thứ i là 1 nếu x_i có mặt trong tích sơ cấp là bình thường, còn nếu x_i có mặt trong tích sơ cấp với dấu phủ định thì ký tự này là 0, cuối cùng nếu x_i không có mặt thì ký tự này sẽ là $-$. Chẳng hạn, với tích sơ cấp của 5 biến $x_1, x_2, \dots, x_5$ là $\overline{x_1}x_3x_4\overline{x_5}$, thì nó sẽ được biểu diễn bằng dãy 0 – 110. Với cách biểu diễn này, hai tích sơ cấp được gọi là *kề nhau* nếu qua các biểu diễn tương ứng chúng chỉ khác nhau ở một vị trí 0 hoặc 1. Khi đó, với các tích sơ cấp kề nhau này chúng ta có thể thay bằng một tích sơ cấp gọn hơn. Quá trình thực hiện như thế này được gọi là *phép dán* của các tích sơ cấp. Chẳng hạn, với hai tích sơ cấp kề nhau Ax và $A\overline{x}$, có thể được thay bằng tích A (dùng tiên đề bù) như sau:

$$Ax + A\overline{x} = A.$$

Trên cơ sở phép dán này người ta xây dựng thuật toán để tìm dạng tổng chuẩn

tắc thu gọn của một hàm Boole sau đây.

Thuật toán 4.1 (*Tìm dạng tổng chuẩn tắc thu gọn*)

Vào: Hàm Boole F bậc n dưới dạng tổng chuẩn tắc hoàn toàn.

Ra: Dạng tổng chuẩn tắc thu gọn của F .

Phương pháp:

Bước 1. (*Lập bảng*) Lập một bảng gồm nhiều cột để ghi các kết quả dán. Ở cột thứ nhất viết vào các biểu diễn của các nguyên nhân của F . Chia các biểu diễn này thành từng nhóm, trong đó mỗi nhóm có số các ký hiệu 1 bằng nhau và thứ tự của các nhóm là tăng dần theo số các ký hiệu 1.

Bước 2. (*Lập: dán các biểu diễn*) Lần lượt thực hiện tất cả các phép dán các biểu diễn trong nhóm i với các biểu diễn trong nhóm $i + 1$ ($i = 1, 2, \dots$). Biểu diễn nào tham gia ít nhất một phép dán sẽ được ghi nhận bằng một dấu * bên cạnh. Kết quả dán được ghi vào cột tiếp theo. Lập lại phép dán cho cột kế tiếp đến khi không thu thêm được cột nào mới.

Bước 3. (*Kết luận*) Tất cả các biểu diễn không có dấu * chính là hệ S gồm tất cả các nguyên nhân nguyên tố của F . Khi đó, tổng $\sum_{G \in S} G$ là dạng tổng chuẩn tắc thu gọn của hàm F cần tìm.

Ví dụ 4.14. Tìm tổng chuẩn tắc thu gọn của hàm Boole 4 biến $F(w, x, y, z)$ sau:

$$F(w, x, y, z) = \bar{w}\bar{x}\bar{y}z + \bar{w}x\bar{y}z + \bar{w}\bar{x}yz + w\bar{x}\bar{y}z + w\bar{x}yz + \bar{w}xyz + wxyz.$$

Lập bảng ta có:

0001*	0-01*	0- -1
0101*	00-1*	-0-1
0011*	-001*	- -11
1001*	-011*	
1011*	10-1*	
0111*	01-1*	
1111*	0-11*	
	1-11*	
	-111*	

Như vậy, dạng tổng chuẩn tắc thu gọn của hàm F là

$$F((w, x, y, z)) = \bar{w}z + \bar{x}z + yz.$$

Ví dụ 4.15. Tìm tổng chuẩn tắc thu gọn của hàm Boole 4 biến $F(w, x, y, z)$ sau:

$$F(w, x, y, z) = \bar{w}\bar{x}y\bar{z} + \bar{w}\bar{x}yz + w\bar{x}yz + w\bar{x}\bar{y}\bar{z} + w\bar{x}\bar{y}z + wxy\bar{z} + wxyz.$$

Lập bảng ta có:

0010*	001-	11- -
0011*	-011	
1100*	110-*	
1011*	11-0*	
1101*	1-11	
1110*	11-1*	
1111*	111-*	

Vậy, dạng tổng chuẩn tắc thu gọn của hàm F là:

$$F(w, x, y, z) = \bar{w}\bar{x}y + \bar{x}yz + wyz + wx.$$

4.4.2.2. Tìm dạng tổng chuẩn tắc cực tiểu

Sau khi tìm được dạng tổng chuẩn tắc thu gọn của hàm Boole bậc n F bằng Thuật toán 4.1, tức là tìm được tất cả các nguyên nhân nguyên tố của F , chúng ta tìm dạng tổng chuẩn tắc cực tiểu (tối thiểu) của F theo thuật toán sau. Để bắt đầu, chúng ta cần một số khái niệm sau đây. Mỗi tích sơ cấp trong dạng tổng chuẩn tắc hoàn toàn của hàm F còn được gọi là một *cấu tạo đơn vị* của F . Nếu một nguyên nhân nguyên tố A là một phần con của cấu tạo đơn vị B , thì ta nói A là *phủ* B . Gọi S là tập tất cả các nguyên nhân nguyên tố của F . Nếu mọi cấu tạo đơn vị của F đều được phủ ít nhất bởi một phần tử của S thì ta nói hệ S là *phủ* F . Dễ thấy, nếu S là phủ hàm F thì S là hệ đầy đủ đối với F . Khi đó, một nguyên nhân nguyên tố trong S được gọi là *cốt yếu* nếu bỏ nó đi khỏi S thì các nguyên nhân nguyên tố còn lại của S không thể phủ F .

Thuật toán 4.2 (*Tìm dạng tổng chuẩn tắc cực tiểu*)

Vào: Hàm Boole F bậc n và dạng chuẩn tắc thu gọn của F .

Ra: Dạng tổng chuẩn tắc cực tiểu của F .

Phương pháp:

Bước 1. (Lập bảng) Lập một bảng chữ nhật, trong đó mỗi cột ứng với một cấu tạo đơn vị của F và mỗi dòng ứng với một nguyên nhân nguyên tố của F . Tại ô (i, j) ta đánh dấu cộng (+) nếu nguyên nhân nguyên tố ở dòng i là phủ cấu tạo đơn vị ở cột j .

Bước 2. (Xác định các nguyên nhân nguyên tố cốt yếu) Tại những cột chỉ có duy nhất một dấu +, xem dấu + đó thuộc dòng nào thì dòng đó ứng với một nguyên nhân nguyên tố cốt yếu.

Bước 3. (Xóa cột và dòng) Xóa tất cả các cột được phủ bởi các nguyên nhân nguyên tố cốt yếu. Trong bảng còn lại, xóa tiếp những dòng không còn dấu +, và sau đó nếu có hai cột giống nhau thì xóa bớt một cột.

Bước 4. (Kết luận) Xác định một hệ S gồm các nguyên nhân nguyên tố với số biến ít nhất phủ các cột còn lại. Khi đó, tổng của các nguyên nhân nguyên tố cốt yếu và các nguyên nhân nguyên tố trong S chính là dạng tổng chuẩn tắc cực tiểu của hàm F đã cho.

Các Bước 2 và 3 trong Thuật toán 4.2 có tác dụng rút gọn bảng trước khi lựa chọn. Độ phức tạp của thuật toán chủ yếu nằm ở Bước 4. Tình huống tốt nhất là mọi nguyên nhân nguyên tố đều là cốt yếu. Trường hợp này không phải lựa chọn gì và hàm F có duy nhất một dạng tổng chuẩn tắc cực tiểu cũng chính là dạng tổng chuẩn tắc thu gọn. Trường hợp xấu nhất là không có nguyên nhân nguyên tố nào là cốt yếu. Trường hợp này ta phải lựa chọn toàn bộ bảng.

Ví dụ 4.16. Cực tiểu hàm Boole 4 biến $F(w, x, y, z)$ trong Ví dụ 4.14.

Từ tổng chuẩn tắc thu gọn tìm được ở trên, lập bảng ta có:

	$\overline{w}\overline{x}\overline{y}z$	$\overline{w}x\overline{y}z$	$\overline{w}x\overline{y}z$	$w\overline{x}\overline{y}z$	$w\overline{x}yz$	$\overline{w}xyz$	$wxyz$
$\overline{w}z$	+	+	+			+	
$\overline{x}z$	+		+	+	+		
yz			+		+	+	+

Rõ ràng, trong bảng trên các nguyên nhân nguyên tố đều là cốt yếu nên tổng chuẩn tắc cực tiểu của hàm F là:

$$F(w, x, y, z) = \overline{w}z + \overline{x}z + yz.$$

Ví dụ 4.17. Cực tiểu hàm Boole 4 biến $F(w, x, y, z)$ trong Ví dụ 4.15.

Từ tổng chuẩn tắc thu gọn tìm được ở trên, lập bảng ta được:

	$\overline{w} \overline{x} y \overline{z}$	$\overline{w} \overline{x} y z$	$w \overline{x} y z$	$w x \overline{y} \overline{z}$	$w x \overline{y} z$	$w x y \overline{z}$	$w x y z$
$w x$				+	+	+	+
$\overline{w} \overline{x} y$	+	+					
$\overline{x} y z$		+	+				
$w y z$			+				+

Rõ ràng, trong bảng trên các nguyên nhân nguyên tố cốt yếu nằm ở dòng 1 và 2, do đó ta xóa các cột 1, 2, 4, 5, 6 và 7. Hai dòng 1 và 2 không chứa dấu + cũng xóa đi. Bảng sau khi rút gọn là:

	$w \overline{x} y z$
$\overline{x} y z$	+
$w y z$	+

Chọn $S = \{\overline{x} y z\}$ hoặc $S = \{w y z\}$ để phủ cột còn lại $w \overline{x} y z$. Suy ra, có hai dạng tổng chuẩn tắc cực tiểu của hàm F là:

$$F_1(w, x, y, z) = w x + \overline{w} \overline{x} y + \overline{x} y z$$

hoặc

$$F_2(w, x, y, z) = w x + \overline{w} \overline{x} y + w y z.$$

Tài liệu tham khảo

- [1] Berge, C., *Graphs and hypergraphs*, North-Holland Pub. Co., 1973.
- [2] Bondy, J. A. and Murty, U. S. R., *Graph theory and applications*, North-Holland Pub. Co., 1982.
- [3] Gallai, T., “Über extreme Punkt- und Kantenmengen”, *Ann. Univ. Sci. Budapest. Eötvös Sect. Math.* 2, 133-138, 1959.
- [4] Hoàng Chung, *Đại cương về toán học hữu hạn*, Nhà xuất bản Giáo dục, 1998.
- [5] Nguyễn Gia Định, *Giáo trình toán rời rạc*, Đại học Khoa học Huế, 2003.
- [6] Vũ Đình Hòa, *Toán rời rạc*, Nhà xuất bản Đại học Sư phạm, 2010.
- [7] Kenneth H. Rosen, *Toán học rời rạc-ứng dụng trong tin học*, 1998 (sách biên dịch).
- [8] Nguyễn Đức Nghĩa và Nguyễn Tô Thành, *Toán rời rạc*, Nhà xuất bản Đại học Quốc gia Hà nội, 2003.
- [9] Đặng Huy Ruận, *Lý thuyết đồ thị và ứng dụng*, Nhà xuất bản Khoa học và Kỹ thuật, 2001.
- [10] Schulz, A. “Correctness-Proof of a greedy-algorithm for minimum vertex cover of a tree”, *CS Stack Exchange*, Retrieved 8 July 2014.
- [11] Nguyễn Hoàng Sơn, *Bài giảng Lý thuyết độ phức tạp tính toán*, 2008.
- [12] Ngô Đắc Tân, *Lý thuyết tổ hợp và đồ thị*, Nhà xuất bản Đại học Quốc gia Hà nội, 2004.
- [13] Hoàng Chí Thành, *Đồ thị và các thuật toán*, Nhà xuất bản Giáo dục, 2007.
- [14] Yannakakis, M. and Gavril, F., "Edge dominating sets in graphs", *SIAM Journal on Applied Mathematics* 38 (3): 364–372, 1980.