Nguyễn Hữu Điển

GIÁO TRÌNH
TOÁN RỜI RẠC

Nhà xuất bản Đại học Quốc gia Hà Nội

$\frac{51}{GD-02}$1536/889-02 Mã số: 8H663T1

Lời nói đầu

Toán rời rạc là một lĩnh vực của toán học nghiên cứu các đối tượng rời rạc. Chúng ta sẽ sử dụng công cụ của toán rời rạc khi phải đếm các đối tượng, khi nghiên cứu quan hệ giữa các tập rời rạc, khi phân tích các quá trình hữu hạn. Một trong những nguyên nhân chủ yếu làm nâng tầm quan trọng của toán rời rạc là việc cất giữ và xử lý thông tin trên máy tính bản chất là các quá trình rời rạc. Cuốn sách này nhằm giới thiệu các kiến thức cơ bản trong ba lĩnh vực có nhiều ứng dụng của toán rời rạc là: lý thuyết tổ hợp, lý thuyết đồ thị và lôgic toán học.

Đây là bài giảng tôi đã đọc tại ĐHKHTN Hà Nội, với một học kì trong vòng 45 đến 60 tiết. Tôi đã chọn lọc những chủ để khởi đầu về Toán rời rạc. Mỗi chủ đề đều đi từ đơn giản tới các kết quả có ý nghĩa và cuối cùng là các thuật toán. Mỗi chủ đề còn nhiều vấn đề mở rộng, vì thời gain có hạn nên tôi chỉ giới thiệu trên lớp và dành cho những người sau này chuyên nghiên cứu về các bài toán tối ưu.

Tôi cũng tuyển chọn những bài tập điển hình và cho sinh viên thực hành trên các bài tập đó. Nội dung được lấy từ các tài liệu [2],[9],[8],[4],[10].

Lần đầu tiên biên soạn không thể tránh khỏi sai sót và nhầm lẫn mong bạn đọc cho ý kiến. Mọi góp ý gửi về địa chỉ:
huudien@vnu.edu.vn.

Hà Nội, tháng 04 năm 2019

Tác giả

NỘI DUNG

DANH SÁCH CÁC HÌNH

Chương 1

LÔGIC MỆNH ĐỀ

Công trình viết về lô gic đầu tiên bởi nhà triết học Hy Lạp Aristotle. Công trình là tập hợp những luật suy diễn làm cơ sở cho việcùnghiên cứu của mọi ngành nghiên cứu trí thức. Đến thế kỷ thứ 17 nhà triết học, toán học Đức, Gottfried Leibniz đã phát triển ý tưởng dùng những ký hiệu để thực hiện những quy trình suy diễn trong ký hiệu đại số, số học và cácùngành liên quan. Những tư tưởng của Leibniz được thực hiện cụ thể thành ngành lô gic ký hiệu do hai tác giả George Boole và Augustus De Morgan.

Trong chương này ta đưa vào những khái niệm và ký hiệu lôgic làm cơ sở nghiên cứu sau này.

1.1 Mệnh đề và giá trị chân trị

Các đối tượng cơ bản mà chúng ta khảo sát ở đây là các phát biểu hay các mệnh đề. Tuy nhiên trong chương này ta chỉ xét đến các mệnh đề toán học, và chúng ta nói vắn tắt các mệnh đề toán học là các mệnh đề. Đó là những phát biểu để diễn đạt một ý tưởng trọn vẹn và ta có thể khẳng định một cách khách quan là nó đúng hoặc sai. Tính chất cốt yếu của một mệnh đề là nó đúng hoặc sai, và không thể vừa đúng vừa sai. Giá trị đúng hoặc sai của một mệnh đề được gọi là chân trị của mệnh đề.

Về mặt ký hiệu, ta thường dùng các mẫu tự (như $p, q, r, ...$) để ký hiệu cho các mệnh đề, và chúng cũng được dùng để ký hiệu cho các biến logic, tức là các biến lấy giá trị đúng hoặc sai. chân trị "đúng" thường được viết là 1, và chân trị "sai" được viết là 0.

Ví dụ 1.1. Các phát biểu sau đây là các mệnh đề (toán học).

1. 6 là một số nguyên tố.
2. 5 là một số nguyên tố.
3. $-3 < 2$.
4. Tam giác cân có hai góc bằng nhau.
5. H_2O là một axít.

Các mệnh đề 2, 3, và 4 trong ví dụ trên là những mệnh đề đúng. Nói cách khác chân trị của các mệnh đề này là đúng. Các mệnh đề 1, 5 là những mệnh đề chân sai.

Ví dụ 1.2. Các phát biểu sau đây không phải là các mệnh đề vì tính đúng sai của chúng không xác định.

1. Ai đang đọc sách? (một câu hỏi)
2. Hãy đóng cửa lại đi!
3. Anh ta rất thông minh.
4. Cho x là một số nguyên dương.
5. a là một số chính phương.
6. $x + y = z$.

Trong việc khảo sát các mệnh đề, người ta còn phân ra làm hai loại:

Định nghĩa 1.1. *Mệnh đề sơ cấp* (elementary), mệnh đề phức hợp (compound). Mệnh đề sơ cấp là các "nguyên tử" theo nghĩa là nó không thể được phân tích thành một hay nhiều (từ hai trở lên) mệnh đề thành phần đơn giản hơn.

Định nghĩa 1.2. *Mệnh đề phức hợp* là mệnh đề được tạo thành từ một hay nhiều mệnh đề khác 2 bằng cách sử dụng các liên kết logic như từ "không" dùng trong việc phủ định một mệnh đề, các từ nối: "và", "hay", "hoặc", "suy ra", v.v....

Còn mệnh đề phức hợp là mệnh đề được tạo thành từ một hay nhiều mệnh đề khác 2 bằng cách sử dụng các liên kết logic như từ "không" dùng trong việc phủ định một mệnh đề, các từ nối: "và", "hay", "hoặc", "suy ra", v.v....

Ví dụ 1.3. Xét các mệnh đề sau đây.
$p =$ "15 chia hết cho 3".
$q =$ "2 là một số nguyên tố và là một số lẻ".

Ta có p là một mệnh đề sơ cấp. Nhưng q là một mệnh đề phức hợp, vì mệnh đề q được tạo thành từ hai mệnh đề "2 là một số nguyên tố" và "2 là một số lẻ" nhờ vào liên kết logic "và".

1.2 Các phép toán trên mệnh đề

Điều mà chúng ta quan tâm ở đây không phải là xác định tính đúng hoặc sai của một mệnh đề sơ cấp. Bởi vì những mệnh đề này thường là những phát biểu nói lên một ý tưởng nào đó trong một phạm vi chuyên môn nhất định. Vấn đề mà ta quan tâm ở đây là làm thế nào để tính toán chân trị của các mệnh đề phức hợp theo các mệnh đề sơ cấp cấu thành mệnh đề phức hợp đó nhờ vào các phép toán logic. Các phép toán logic ở đây là các ký hiệu được dùng thay cho các từ liên kết logic như "không", "và", "hay", "hoặc", "suy ra" hay "nếu ... thì ...", "nếu và chỉ nếu".

Các phép toán logic được định nghĩa bằng bảng chân trị (truth table). Bảng chân trị chỉ ra rõ ràng chân trị của mệnh đề phức hợp theo từng trường hợp của các chân trị của các mệnh đề sơ cấp tạo thành mệnh đề phức hợp. Bảng chân trị của các phép toán logic tất nhiên là phản ánh ngữ nghĩa tự nhiên của các từ liên kết tương ứng. Về mặt tự nhiên của ngôn ngữ, trong nhiều trường hợp cùng một từ nhưng có thể có nghĩa khác nhau trong những ngữ cảnh khác nhau. Do đó, bảng chân trị không thể diễn đạt mọi nghĩa có thể có của từ tương ứng với ký hiệu phép toán. Điều này cho thấy rằng đại số logic là rõ ràng hoàn chỉnh theo nghĩa là nó cho ta một hệ thống logic đáng tin cậy. Đại số logic còn đặc biệt quan trọng trong việc thiết kế mạch cho máy tính.

Bảng chân trị không chỉ dùng để kê ra sự liên hệ chân trị giữa mệnh đề phức hợp với chân trị của các mệnh đề sơ cấp cấu thành nó, mà bảng chân trị còn được dùng với mục đích rộng hơn: liệt kê sự liên hệ chân trị giữa các mệnh với các mệnh đề đơn giản hơn cấu thành chúng.

1.2.1. Phép phủ định

Cho p là một mệnh đề, chúng ta dùng ký hiệu $\overline{p}$ để chỉ mệnh đề phủ định của mệnh đề p. "Sự phủ định" được định nghĩa bởi bảng chân trị sau đây:

p	$\overline{p}$
0	1
1	0

Ký hiệu $^{-}$ được đọc là "không" . Trong một số sách khác, người ta còn dùng các ký hiệu sau đây để chỉ mệnh đề phủ định của một mệnh đề p: $\sim p, \neg p$.

Trong cột thứ nhất của bảng chân trị , ta liệt kê đầy đủ các trường hợp chân trị có thể có của mệnh đề p. Ở cột thứ hai kê ra chân trị tương ứng của mệnh đề $\overline{p}$ theo từng trường hợp chân trị của mệnh đề p. Định nghĩa này phù hợp với ngữ nghĩa tự nhiên của sự phủ định : Mệnh đề phủ định $\overline{p}$ có chân trị là đúng (1) khi mệnh đề p có chân trị sai (0), ngược lại $\overline{p}$ có chân trị sai (0) khi p có chân trị đúng (1).

Ví dụ 1.4. Nếu ta ký hiệu p là mệnh đề "$5 < 3$" thì $\overline{p}$ là ký hiệu cho mệnh đề "$5 \geq 3$". Trong trường hợp này p sai, $\overline{p}$ đúng. Ta có thể viết $p = 0, \overline{p} = 1$.

Ví dụ 1.5. Chỉ ra rằng $\overline{\overline{p}}$ và p luôn có cùng chân trị .

Lời giải. Lập bảng chân trị của mệnh đề $\overline{\overline{p}}$:

p	$\overline{p}$	$\overline{\overline{p}}$
0	1	0
1	0	1

Trên mỗi dòng giá trị trong bảng chân trị ta có chân trị của p và $\overline{\overline{p}}$ đều bằng nhau (so sánh cột 1 và cột 3 trong bảng). Vậy $\overline{\overline{p}}$ và p có cùng chân trị . Ta cũng nói rằng $\overline{\overline{p}}$ tương đương logic với p. ☺

Mệnh đề $\overline{(\overline{p})}$ thường được viết là $\overline{\overline{p}}$, vì điều này không có gì gây ra sự nhầm lẫn.

1.2.2. Phép hội

Định nghĩa 1.3. Cho p và q là hai mệnh đề. Ta ký hiệu mệnh đề "p hội q" là $p \wedge q$. Phép "và", ký hiệu là $\wedge$, được định nghĩa bởi bảng chân trị sau đây:

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

Chân trị của mệnh đề $p \wedge q$ phụ thuộc vào các chân trị của 2 mệnh đề p, q. Ta có 4 trường hợp chân trị của $p \wedge q$ ứng với 4 trường hợp chân trị của cặp mệnh đề (p, q) là $(0,0), (0,1), (1,0), (1,1)$. Trong 4 trường hợp chỉ có một trường hợp mệnh đề $p \wedge q$ đúng, đó là trường hợp p đúng và q đúng.

Qua định nghĩa trên ta nhận thấy rằng các mệnh đề $p \wedge q$ và $q \wedge p$ luôn luôn có cùng chân trị , hay tương đương logic. Tuy nhiên, trong ngôn ngữ thông thường các mệnh đề "p và q" và "q và p" đôi khi có ý nghĩa khác nhau theo ngữ cảnh.

Ví dụ 1.6. Cho các mệnh đề

- $p =$ "$5 > -7$",
- $q =$ "2721 là một số nguyên tố",
- $r =$ "một tam giác đều cũng là một tam giác cân".

Khi đó ta có

- $p \wedge q = 0$ ($p \wedge q$ sai, tức là có chân trị bằng 0, vì$p = 1$ và $q = 0$),
- $p \wedge r = 1$ ($p \wedge r$ đúng, tức là có chân trị bằng 1, vì $p = 1$ và $r = 1$).

Nhận xét. Bằng cách lập bảng chân trị , ta có

1. Các mệnh đề p và $p \wedge p$ luôn có cùng chân trị .
2. Mệnh đề $p \wedge \overline{p}$ luôn có chân trị bằng 0 (tức là một mệnh đề luôn sai).

Một mệnh đề phức hợp luôn luôn có chân trị là sai trong mọi trường hợp chân trị của các mệnh đề sơ cấp tạo thành nó sẽ được gọi là một sự mâu thuẩn.

1.2.3. Phép tuyển

Định nghĩa 1.4. Cho p và q là hai mệnh đề. Ta ký hiệu mệnh đề "p hay q" là $p \vee q$. Phép "hay", ký hiệu là $\vee$, được định nghĩa bởi bảng chân trị sau đây:

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

Chân trị của mệnh đề $p \vee q$ phụ thuộc vào các chân trị của 2 mệnh đề p, q. Trong 4 trường hợp chỉ có một trường hợp mệnh đề $p \vee q$ sai, đó là trường hợp p sai và q sai. Qua định nghĩa trên ta nhận thấy rằng các mệnh đề $p \vee q$ và $q \vee p$ luôn luôn có cùng chân trị , hay tương đương logic.

Ví dụ 1.7. Cho các mệnh đề

- $p =$ "$5 > 7$",
- $q =$ "2721 là một số nguyên tố",
- $r =$ "một tam giác đều cũng là một tam giác cân".

Khi đó ta có

- $p \vee q = 0$,
- $p \vee r = 1$.

Nhận xét.

1. Cho p là một mệnh đề. Lập bảng chân trị của mệnh đề $p \vee \overline{p}$

p	$\overline{p}$	$p \vee \overline{p}$
0	1	1
1	0	1

   ta có mệnh đề $p \vee \overline{p}$ luôn luôn đúng.

2. Người ta còn sử dụng phép "hoặc" trong việc liên kết các mệnh đề. Cho p và q là hai mệnh đề. Ta ký hiệu mệnh đề "p hoặc q" là

$p \vee q$. Phép "hoặc", ký hiệu là $p \underline{\vee} q$, được định nghĩa bởi bảng chân trị sau đây

p	q	$p \underline{\vee} q$
0	0	0
0	1	1
1	0	1
1	1	0

Phép "hoặc" còn được gọi là "hay loại trừ". chân trị của mệnh đề $p \underline{\vee} q$ phụ thuộc vào các chân trị của 2 mệnh đề p, q : mệnh đề $p \underline{\vee} q$ đúng khi trong 2 mệnh đề p và q có một mệnh đề đúng, một mệnh đề sai.

1.2.4. Phép kéo theo

Phép kéo theo, ký hiệu bởi $\rightarrow$, được đưa ra để mô hình cho loại phát biểu điều kiện có dạng : "nếu . . . thì . . .". Cho p và q là 2 mệnh đề, ta sẽ viết $p \rightarrow q$ để diễn đạt phát biểu "nếu p thì q". Phép toán kéo theo $\rightarrow$ được định nghĩa bởi bảng chân trị sau đây

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Mệnh đề $p \rightarrow q$, được đọc là "nếu p thì q", còn được phát biểu dưới các dạng khác sau đây

- "q nếu p".
- "p chỉ nếu q".
- "p là điều kiện đủ cho q".
- "q là điều kiện cần cho p".

1.2.5. Phép tương đương

Phép kéo theo 2 chiều hay phép tương đương, ký hiệu bởi $\leftrightarrow$, được đưa ra để mô hình cho loại phát biểu điều kiện hai chiều có dạng : ". . . nếu và chỉ nếu . . .". Cho p và q là 2 mệnh đề, ta viết $p \leftrightarrow q$ để diễn đạt phát biểu "p nếu và chỉ nếu q". Phép toán tương đương $\leftrightarrow$ được định nghĩa bởi bảng chân trị sau đây

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

1.2.6. Độ ưu tiên của các toán tử logic

Tương tự như đối với các phép toán số học, để tránh phải dùng nhiều dấu ngoặc trong các biểu thức logic, ta đưa ra một thứ tự ưu tiên trong việc tính toán. Ở trên ta có 5 toán tử logic $\bar{}$ (không) , $\wedge$ (và), $\vee$ (hay), $\rightarrow$ (kéo theo), $\leftrightarrow$ (tương đương)

$\bar{}$	ưu tiên mức 1 (cao nhất)
$\wedge, \vee$	ưu tiên mức 2 (thấp hơn)
$\rightarrow$, $\leftrightarrow$	ưu tiên mức 3 (thấp nhất)

trong đó, các toán tử liệt kê trên cùng dòng có cùng độ ưu tiên.

Ví dụ 1.8. 1. $\overline{p} \vee q$ có nghĩa là $((\overline{p}) \vee q)$.

2. $\overline{p} \vee q \rightarrow r \vee s$ có nghĩa là $(((\overline{p}) \vee q) \rightarrow (r \wedge s))$.
3. $\overline{p} \vee q \wedge r$ là nhập nhằng. Cần phải dùng các dấu ngoặc để chỉ rõ nghĩa.

1.3 Biểu thức logic

1.3.1. Định nghĩa và bảng chân trị biểu thức logic

Trong đại số ta có các biểu thức đại số được xây dựng từ các hằng số, các biến và các phép toán. Khi thay thế các biến trong một biểu thức đại số bởi các hằng số thì kết quả thực hiện các phép toán trong biểu thức sẽ là một hằng số.

Định nghĩa 1.5. Trong phép tính mệnh đề ta cũng có các *biểu thức logic* được xây dựng từ

- Các mệnh đề hay các giá trị hằng.
- Các biến mệnh đề.
- Các phép toán logic, và cả các dấu ngoặc "( )" để chỉ rõ thứ tự thực hiện của các phép toán.

Giả sử E, F là 2 biểu thức logic, khi ấy $\overline{E}, E \wedge F, E \vee F, E \rightarrow F, E \leftrightarrow F$ cũng là các biểu thức logic.

Ví dụ 1.9. Biểu diễn $E(p, q, r) = (((\overline{p}) \vee q) \rightarrow (r \wedge s))$ là một biểu thức logic trong đó p, q, r là các biến mệnh đề.

Bảng chân trị của biểu thức logic

Bảng chân trị của một biểu thức logic là bảng liệt kê chân trị của biểu thức logic theo các trường hợp về chân trị của tất cả các biến mệnh đề trong biểu thức logic hay theo các bộ giá trị của bộ biến mệnh đề. Với một biến mệnh đề, ta có 2 trường hợp là 0 (sai) hoặc 1 (đúng).

Với 2 biến mệnh đề p, q ta 4 trường hợp chân trị của bộ biến (p, q) là các bộ giá trị $(0, 0)$, $(0, 1)$, $(1, 0)$, và $(1, 1)$. Trong trường hợp tổng quát, với n biến mệnh đề thì ta có $2n$ trường hợp chân trị cho bộ n biến đó.

Ví dụ 1.10. Bảng chân trị của các biểu thức logic $p \rightarrow q$ và $\overline{p} \vee q$ theo các biến mệnh đề p, q như sau

p	q	$p \rightarrow q$	$\overline{p}$	$\overline{p} \vee q$
0	0	1	1	1
0	1	1	1	1
1	0	0	0	0
1	1	1	0	1

Ví dụ 1.11. Bảng chân trị của các biểu thức logic $p \vee (q \wedge r)$ theo các biến mệnh đề p, q, r như sau

Thứ tự	p	q	r	$q \wedge r$	$p \vee (q \wedge r)$
1	0	0	0	0	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	1	1	1	1
5	1	0	0	0	1
6	1	0	1	0	1
7	1	1	0	0	1
8	1	1	1	1	1

1.3.2. Sự tương đương logic

Định nghĩa 1.6. Hai biểu thức logic E và F theo các biến mệnh đề nào đó được gọi là *tương đương logic* (*đồng nhất bằng nhau*) khi E và F luôn luôn có cùng chân trị trong mọi trường hợp chân trị của bộ biến mệnh đề.

Khi đó ta viết $E \equiv F$, đọc là "E tương đương với F".

Như vậy, theo định nghĩa ta có thể kiểm tra xem 2 biểu thức logic có tương đương hay không bằng cách lập bảng chân trị của các biểu thức logic.

Ví dụ 1.12. Từ bảng chân trị của các biểu thức logic $p \to q$ và $\overline{p} \vee q$ theo các biến mệnh đề p, q ta có $(p \to q) \equiv (\overline{p} \vee q)$.

1.3.3. Giá trị của biểu thức logic

Một biểu thức logic được tạo thành từ các biến logic kết hợp với phép toán logic, bởi vậy nên giá trị biểu thức logic cũng chỉ nhận 1 trong 2 giá trị là "đúng" ($true$ hoặc 1) hay "sai" ($false$ hoặc 0) tùy thuộc vào giá trị của các biến logic và quy luật của các phép toán.

Ví dụ 1.13. Xét biểu thức logic $(\overline{p} \vee q)$, nếu thay $p = 1$ và $q = 0$ ta có

$$\overline{1} \vee 0 = 0 \vee 0 = 0.$$

1.4 Các luật logic và sử dụng

Các luật logic là cơ sở để ta thực hiện các biến đổi trên một biểu thức logic để có được một biểu thức logic mới tương đương logic với biểu thức logic có trước. Mỗi biểu thức logic cho ta một sự khẳng định về sự tương đương của 2 biểu thức logic. Ta sẽ sử dụng các qui tắc thay thế và các luật logic đã biết để thực hiện các phép biến đổi tương đương trên các biêu thức logic.

Dưới đây, chúng ta sẽ liệt kê ra một số luật logic thường được sử dụng trong lập luận và chứng minh. Các luật này có thể được suy ra trực tiếp từ các bảng chân trị của các biểu thức logic.

1.4.1. Các luật logic

1. Các luật về phép phủ định
   - $\overline{\overline{p}} \equiv p$ (luật phủ định của phủ định);
   - $\overline{1} \equiv 0$;
   - $\overline{0} \equiv 1$.
2. Luật giao hoán
   - $p \wedge q \equiv q \wedge p$;
   - $p \vee q \equiv q \vee p$.
3. Luật kết hợp
   - $p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r$;
   - $p \vee (q \vee r) \equiv (p \vee q) \vee r$.
4. Luật phân bố
   - $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$;
   - $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$.
5. Luật De Morgan
   - $\overline{p \wedge q} \equiv \overline{p} \vee \overline{q}$;
   - $\overline{p \vee q} \equiv \overline{p} \wedge \overline{q}$.

6. Luật về phần tử bù
   - $p \vee \overline{p} \equiv 1$;
   - $p \wedge \overline{p} \equiv 0$.
7. Luật kéo theo
   - $p \to q \equiv \overline{p} \vee q$.
8. Luật tương đương
   - $p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$.
9. Các luật đơn giản của phép tuyển
   - $p \vee p \equiv p$ (tính lũy đẳng của phép tuyển);
   - $p \vee 1 \equiv 1$ (luật này còn được gọi là luật thống trị);
   - $p \vee 0 \equiv p$ (luật này còn được gọi là luật trung hòa);
   - $p \vee (p \wedge q) \equiv p$ (luật này còn được gọi là luật hấp thụ).
10. Các luật đơn giản của phép hội
    - $p \wedge p \equiv p$ (tính lũy đẳng của phép hội);
    - $p \wedge 1 \equiv p$ (luật này còn được gọi là luật trung hòa);
    - $p \wedge 0 \equiv 0$ (luật này còn được gọi là luật thống trị);
    - $p \wedge (p \vee q) \equiv p$ (luật này còn được gọi là luật hấp thụ).

Một số luật trong các luật trình bày ở trên có thể được suy ra từ các luật khác. Chúng ta có thể tìm ra được một tập hợp luật logic tối thiểu mà từ đó ta có thể suy ra tất cả các luật logic khác, nhưng điều này không quan trọng lắm đối với chúng ta.

Những luật trên được chọn lựa để làm cơ sở cho chúng ta thực hiện các biến đổi logic, suy luận và chứng minh. Tất nhiên là còn nhiều luật logic khác mà ta không liệt kê ra ở đây.

Các luật kết hợp trình bày ở trên còn được gọi là tính chất kết hợp của phép toán hội và phép toán tuyển . Do tính chất này, ta có thể viết các biểu thức logic hội và các biểu thức tuyển dưới các dạng sau

- $E_1 \wedge E_2 \wedge \ldots \wedge E_m$.

- $E_1 \vee E_2 \vee \cdots \vee E_m$.

và việc tính toán chân trị có thể được thực hiện dựa trên một sự phân bố các cặp dấu ngoặc vào biểu thức một cách tùy ý để xác định một trình tự thực hiện các phép toán.

Ví dụ 1.14. Biểu thức $E_1 \wedge E_2 \wedge E_3 \wedge E_4$ có thể được tính toán chân trị bởi biểu thức sau

$$(E_1 \wedge E_2) \wedge (E_3 \wedge E_4)$$

hay có thể tính toán theo biểu thức

$$E_1 \wedge ((E_2 \wedge E_3) \wedge E_4).$$

1.4.2. Các quy tắc thay thế

Dưới đây là các qui tắc để cho ta có thể suy ra những biểu thức logic mới hay tìm ra các biểu thức logic tương đương với một biểu thức logic cho trước.

- **Qui tắc 1.**
  Trong một biểu thức logic E, nếu ta thay thế một biểu thức con bởi một biểu thức logic tương đương với biểu thức con đó thì ta sẽ được một biểu thức mới E' tương đương với biểu thức E.

  Ví dụ 1.15. Cho biểu thức logic $E = q \wedge \overline{p}$. Thay thế q trong biểu thức E bởi biểu thức $\overline{\overline{q}}$ (tương đương với q) ta được một biểu thức mới $E' = \overline{\overline{q}} \vee \overline{p}$. Theo qui tắc thay thế 1 ta có

  $$q \vee \overline{p} \equiv \overline{\overline{q}} \vee \overline{p}.$$

- **Qui tắc 2.**
  Giả sử biểu thức logic E là một hằng đúng. Nếu ta thay thế một biến mệnh đề p bởi một biểu thức logic tuỳ ý thì ta sẽ được một biểu thức logic mới E' cũng là một hằng đúng.

Ví dụ 1.16. Ta có biểu thức $E(p,q) = (p \to q) \leftrightarrow (\overline{p} \vee q)$ là một hằng đúng. Thay thế biến q trong biểu thức E bởi biểu thức $q \wedge r$ ta được biểu thức logic mới

$$E'(p,q,r) = (p \to (q \wedge r)) \leftrightarrow (\overline{p} \vee (q \wedge r)).$$

Theo qui tắc thay thế 2 ta có biểu thức $E'(p,q,r)$ cũng là một hằng đúng.

1.4.3. Ví dụ áp dụng

Ví dụ 1.17. Chứng minh rằng $(p \to q) \equiv (\overline{q} \to \overline{p})$.

Lời giải.

$$\begin{aligned}(p \to q) &\equiv \overline{p} \vee q \text{ (luật kéo theo)}\\ &\equiv q \vee \overline{p} \text{ (luật giao hoán)}\\ &\equiv \overline{q} \to \overline{p} \text{ (luật kéo theo)}\end{aligned}$$

Ví dụ 1.18. Chứng minh rằng biểu thức $((p \to q) \wedge p) \to q$ là một hằng đúng.

Lời giải.

$$\begin{aligned}((p \to q) \wedge p) \to q &\equiv \overline{(p \to q) \wedge p} \vee q \text{ (luật kéo theo)}\\ &\equiv (\overline{p \to q} \vee \overline{p}) \vee q \text{ (luật De Morgan)}\\ &\equiv \overline{p \to q} \vee (\overline{p} \vee q) \text{ (luật kết hợp)}\\ &\equiv \overline{p \to q} \vee (p \to q) \text{ (luật kéo theo)}\\ &\equiv 1 \text{ (luật về phần tử bù)}\end{aligned}$$

Vậy biểu thức $((p \to q) \wedge p) \to q$ là hằng đúng.

Ví dụ 1.19. Chứng minh rằng biểu thức $p \wedge q \to p$ là một hằng đúng.

Lời giải.

$$\begin{aligned}p \wedge q \to p &\equiv \overline{p \wedge q} \vee p \text{ (luật kéo theo)}\\ &\equiv (\overline{p} \vee \overline{q}) \vee p \text{ (luật De Morgan)}\\ &\equiv (\overline{q} \vee \overline{p}) \vee p \text{ (luật giao hoán)}\\ &\equiv \overline{q} \vee (\overline{p} \vee p) \text{ (luật kết hợp)}\\ &\equiv \overline{q} \vee 1 \text{ (luật về phần tử bù)}\\ &\equiv 1 \text{ (luật đơn giản)}\end{aligned}$$

Vậy mệnh đề $p \wedge q \to p$ là hằng đúng.

Ví dụ 1.20. Chứng minh rằng biểu thức $p \to p \to q$ là một mệnh đề hằng đúng.

Lời giải.

$$\begin{aligned} p \to p \vee q &\equiv \overline{p} \vee (p \vee q) \text{ (luật kéo theo)} \\ &\equiv (\overline{p} \vee p) \vee q \text{ (luật kết hợp)} \\ &\equiv 1 \vee q \text{ (luật về phần tử bù)} \\ &\equiv 1 \text{ (luật đơn giản).} \end{aligned}$$

Vậy mệnh đề $p \to p \vee q$ là hằng đúng.

Nhận xét. Các ví dụ trên cho ta thấy một quan hệ khác giữa các mệnh đề phức hợp hay các mệnh đề với quan hệ "*suy ra*". Khi mệnh đề $p \to q$ là hằng đúng, ta nói rằng p suy ra q (về mặt logic). Chúng ta sẽ dùng ký hiệu $\Rightarrow$ để chỉ quan hệ "*suy ra*". Quan hệ suy ra này có tính truyền (hay bắc cầu), nhưng không có tính chất đối xứng.

1.5 Các dạng chuẩn tắc

Dạng chuẩn tắc (chính tắc) của 1 biểu thức là biểu diễn biểu thức về dạng đơn giản, chỉ bao gồm các phép toán phủ định, hội tuyển của các mệnh đề.

1.5.1. Các dạng hội sơ cấp và dạng tuyển sơ cấp

Định nghĩa 1.7. Giả sử $p_1, p_2, \ldots, p_n$ là các biến mệnh đề. Một biểu thức logic F theo các biến mệnh đề $p_1, p_2, \ldots, p_n$ được gọi là một *biểu thức hội cơ bản (hội sơ cấp)* nếu nó có dạng sau

$$F = q_1 \wedge q_2 \wedge \ldots \wedge q_n$$

với $q_j = p_j$ hoặc $q_j = \overline{p}_j (j = 1, \ldots, n)$.

Ví dụ 1.21. Biểu thức $x \wedge \overline{y} \wedge z$ là một biểu thức hội cơ bản theo 3 biến mệnh đề x, y, z.

Định nghĩa 1.8. Giả sử $p_1, p_2, \ldots, p_n$ là các biến mệnh đề. Một biểu thức logic F theo các biến mệnh đề $p_1, p_2, \ldots, p_n$ được gọi là *một biểu thức tuyển cơ bản* (tuyển sơ cấp) nếu nó có dạng sau

$$F = q_1 \vee q_2 \vee \ldots q_n$$

với $q_j = p_j$ hoặc $q_j = p_j (j = 1, \ldots, n)$.

Ví dụ 1.22. Biểu thức $x \vee \overline{y} \vee z$ là một biểu thức tuyển cơ bản theo 3 biến mệnh đề x, y, z.

Định lý 1.1. *1) Điều kiện cần và đủ để một Hội sơ cấp đồng nhất sai là trong hội đó có chứa một mệnh đề sơ cấp đồng thời với phủ định của nó.*
2) Điều kiện cần và đủ để một Tuyển sơ cấp đồng nhất đúng là trong tuyển đó có chứa một mệnh đề sơ cấp đồng thời với phủ định của nó.

Chứng minh. 1) Điều kiện cần:

1.5.2. Dạng chuẩn tắc tuyển và dạng chuẩn tắc hội

Định nghĩa 1.9. Biểu thức logic $E(p_1, p_2, \ldots, p_n)$ theo các biến mệnh đề $p_1, p_2, \ldots, p_n$ được nói là có *dạng chính tắc tuyển* khi E có dạng

$$E = E_1 \vee E_2 \vee \cdots \vee E_m.$$

Trong đó mỗi biểu thức con E_i đều có dạng biểu thức hội cơ bản theo các biến $p_1, p_2, \ldots, p_n$.

Ví dụ 1.23. Các biểu thức sau đây có dạng chính tắc tuyển

- $E(x, y, z) = (x \wedge y \wedge z) \vee (\overline{x} \wedge y \wedge z) \vee (x \wedge y \wedge z)$
- $F(p_1, p_2, p_3, p_4) = (p_1 \wedge p_2 \wedge p_3 \wedge p_4) \vee (p_1 \wedge \overline{p}_2 \wedge p_3 \wedge \overline{p}_4)$.

Định lý 1.2. *Mọi biểu thức logic $E(p_1, p_2, \ldots, p_n)$ đều có thể viết dưới dạng chính tắc tuyển (chuẩn tắc tuyển) duy nhất, không kể sự sai khác về thứ tự trước sau của các biểu thức hội cơ bản trong phép tuyển). Nói một cách khác, ta có duy nhất một tập hợp các biểu thức hội cơ bản $\{E_1, E_2, \ldots, E_m\}$ sao cho biểu thức $E(p_1, p_2, \ldots, p_n)$ tương đương logic với biểu thức*

$$E = E_1 \vee E_2 \vee \cdots \vee E_m.$$

Định nghĩa 1.10. Biểu thức logic $E(p_1, p_2, \ldots, p_n)$ theo các biến mệnh đề $p_1, p_2, \ldots, p_n$ được nói là có *dạng chính tắc hội* khi E có dạng

$$E = E_1 \wedge E_2 \wedge \ldots \wedge E_m$$

Trong đó mỗi biểu thức con E_i đều có dạng chính tắc tuyển theo các biến $p_1, p_2, \ldots, p_n$

Ví dụ 1.24. Các biểu thức sau đây có dạng chuẩn tắc hội (chính tắc hội)

- $E(x, y, z) = (x \vee \overline{y} \vee z) \wedge (\overline{x} \vee y \vee z) \wedge (x \wedge y \wedge z)$.
- $F(p_1, p_2, p_3, p_4) = (p_1 \vee p_2 \vee p_3 \vee p_4) \wedge (p_1 \vee \overline{p}_2 \vee p_3 \vee \overline{p}_4)$.

Định lý 1.3. *Mọi biểu thức logic $E(p_1, p_2, \ldots, p_n)$ đều có thể viết dưới dạng chính tắc tuyển và dạng chính tắc hội duy nhất (Không kể sự sai khác về thứ tự trước sau của các biểu thức tuyển cơ bản trong phép hội).*

Nói một cách khác,

Ta có duy nhất một tập hợp các biểu thức hội cơ bản $\{E_1, E_2, \ldots, E_m\}$ sao cho biểu thức $E(p_1, p_2, \ldots, p_n)$ tương đương logic với biểu thức

$$E = E_1 \vee E_2 \vee \cdots \vee E_m.$$

Ta có duy nhất một tập hợp các biểu thức tuyển cơ bản $\{F_1, F_2, \ldots, F_m\}$ sao cho biểu thức $E(p_1, p_2, \ldots, p_n)$ tương đương logic với biêu thức

$$E = F_1 \wedge F_2 \wedge \ldots \wedge F_m.$$

Chứng minh. Giả sử E là một biểu thức lôgic

2) Chứng minh tương tự. ☺

Ví dụ 1.25. Cho $E = p_3 \to (p_2 \to p_1)$ tìm dạng chuẩn tắc hội và dạng chuẩn tắc tuyển.

Lời giải. Ta có $E = p_3 \to (p_2 \to p_1) \equiv \overline{p}_3 \vee \overline{p}_2 \vee p_1$ là dạng chuẩn tắc hội. Nhưng nó cũng có dạng chuẩn tắc tuyển với các hội cơ sở là $\overline{p}_3, \overline{p}_2, p_1$.

Định lý 1.4. *1) Điều kiện cần và đủ để biểu thức lôgic E đồng nhất đúng là trong dạng chuẩn tắc hội của E mỗi tuyển sơ cấp chứa một*

mệnh đề sơ cấp đồng thời với phủ định của nó.
2) Điều kiện cần và đủ để biểu thức lôgic E đồng nhất sai là trong dạng chuẩn tắc tuyển của E mỗi hội sơ cấp chứa một mệnh đề sơ cấp đồng thời với phủ định của nó.

Chứng minh. 1) Điều kiện cần. Giả sử E là công thức đồng nhất đúng. Theo Định lý 1.3 thì E có dạng chuẩn tắc hội

$$E = F_1 \wedge F_2 \wedge \ldots \wedge F_m.$$

Vì E là đồng nhất đúng nên F_i, $i = 1, \ldots, n$, là đồng nhất đúng. Theo Định lý 1.1 thì trong mỗi F_i, $(i = 1, \ldots, n)$ có chứa một mệnh đề cơ cấp đồng thời với phủ định của nó.

Điều kiện đủ. Giả sử F có dạng chuẩn tắc hội

$$E = F_1 \wedge F_2 \wedge \ldots \wedge F_m$$

trong đó mỗi $F_i, (i = 1, \ldots, n)$ có chứa một mệnh đề cơ cấp đồng thời với phủ định của nó. Theo Định lý 1.1 thì mỗi $F_i, (i = 1, \ldots, n)$ là đồng nhất đúng và do đó E là công thức đồng nhất đúng.

2) Chứng minh tương tự. ☺

Từ những định lý trên ta thiết lập một thuật toán nhận biết biểu thức lôgic là hằng đúng, hằng sai và thực hiện được.

1.5.3. Thuật toán nhận biết hằng đúng, hằng sai

Thuật toán.
Đầu vào: E là biểu thức lôgic bất kỳ.
Đầu ra: Dạng chuẩn tắc hội, dạng chuẩn tắc tuyển của E; E là Hằng đúng; Hằng sai.

Bước 1. Khử phép toán kéo theo $(\rightarrow)$ trong E bằng cách sử dụng $p \rightarrow q \equiv \overline{p} \vee q$ ta được công thức tương đương $E_1 \equiv E$.

Bước 2. Đưa phép toán phủ định $(-)$ trong E_1 về trực tiếp với các mệnh đề sơ cấp có mặt trong E_1 bằng áp dụng (luật De Norgan), $\overline{p \vee q} \equiv \overline{p} \wedge \overline{q}$ và $\overline{p \wedge q} \equiv \overline{p} \vee \overline{q}$ ta được biểu thức lôgic mới $A_2 \equiv E_1 \equiv E$

Bước 3. Đưa E_2 về dạng chuẩn tắc hội bằng cách dùng luật phân phối

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

ta được biểu thức lôgic $E_3 \equiv E_2$, ở đây E_3 bao gồm hội những hội sơ cấp.

- Nếu trong mỗi tuyển sơ cấp có chứa một biến mệnh đề sơ cấp đồng thời với phủ định của nó thì E_3 hằng đúng, nghĩa là E đúng.
- Ngược lại, Nếu một tuyển sơ cấp không chứa một biến mệnh đề sơ cấp đồng thời với phủ định của nó thì E_3 hằng không đúng, nghĩa là E không đúng.

Bước 4. Đưa E_2 về dạng chuẩn tắc tuyển bằng cách dùng luật phân phối

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

ta được biểu thức lôgic $E_3' \equiv E_2$, ở đây E_3' bao gồm hội những tuyển sơ cấp.

- Nếu trong mỗi hội sơ cấp có chứa một biến mệnh đề sơ cấp đồng thời với phủ định của nó thì E_3' hằng sai, nghĩa là E sai.
- Ngược lại, Nếu một hội sơ cấp không chứa một biến mệnh đề sơ cấp đồng thời với phủ định của nó thì E_3' hằng không sai, nghĩa là E biểu thức thực hiện được.

Chú ý. Biểu thức E thực hiện được khi và chỉ khi trong E có tồn tại một bộ giá trị đúng, sai của mệnh đề sơ cấp trong E sao cho với bộ giá trị đúng, sai đó thì E nhận giá trị đúng.

1.6 Quy tắc suy diễn

1.6.1. Đại cương về quy tắc suy diễn

Một hệ thống toán học bao gồm các tiên đề, các định nghĩa, và những khái niệm không được định nghĩa. Các tiên đề được giả định là đúng. Các định nghĩa được sử dụng để xây dựng hay đưa ra những khái niệm mới trên cơ sở những khái niệm đã có. Một số thuật ngữ, khái niệm sẽ không được định nghĩa rõ ràng nhưng được ngầm định nghĩa bởi các tiên đề. Trong một hệ toán học chúng ta có thể suy ra được các định lý. Một định lý là một khẳng định được chứng minh là đúng. Một số loại định lý được xem là các bổ đề, các hệ quả.

Một lập luận (hay lý luận) chỉ ra được tính đúng đắn của mệnh đề phát biểu trong định lý được gọi là chứng minh. Logic là một công cụ cho việc phân tích các chứng minh. Trong phần này chúng ta sẽ đề cập đến việc xây dựng một chứng minh toán học. Để thực hiện được một lập luận hay một chứng minh chúng ta cần hiểu các kỹ thuật và các công cụ được sử dụng để xây dựng một chứng minh. Thông thường một chứng minh sẽ bao gồm nhiều bước suy luận mà ở mỗi bước ta đi đến (hay suy ra) một sự khẳng định mới từ những khẳng định đã biết.

Ví dụ 1.26. Một bước suy diễn đúng và sai

(A) Nếu một danh sách L là khác rỗng thì ta có thể lấy ra phần tử đầu trong danh sách.Vì danh sách L là rỗng nên theo sự khẳng định trên ta không thể lấy ra phần tử đầu trong danh sách.

(B) Nếu một danh sách L là khác rỗng thì ta có thể lấy ra phần tử đầu trong danh sách.Vì ta không thể lấy ra phần tử đầu trong danh sách L nên danh sách L là danh sách rỗng.

Trong 2 suy diễn ở ví dụ trên thì suy diễn (B) là một suy luận đúng, nhưng suy diễn (A) là không đúng. Vậy làm thế nào để biết được một suy diễn là đúng hay sai ? Một bước suy luận như thế phải dựa trên một qui tắc suy diễn hợp lý nào đó để nó được xem là một suy luận đúng. Các qui tắc suy diễn là cơ sở để tay biết được một lập luận hay một chứng minh là đúng hay sai. Trong các mục tiếp theo chúng ta sẽ xem xét chi tiết hơn về các qui tắc suy diễn và giới thiệu một số qui tắc suy dễn cơ bản thường được dùng trong việc suy luận và chứng minh.

Định nghĩa qui tắc suy diễn

Tuy có nhiều kỹ thuật, nhiều phương pháp chứng minh khác nhau, nhưng trong chứng minh trong toán học ta thường thấy những lý luận dẫn xuất có dạng

$$\text{Nếu } P_1 \text{ và } P_2 \text{ và } \ldots \text{ và } P_n$$

$$\text{thì } Q.$$

Dạng lý luận này được xem là hợp lý (được chấp nhận là đúng) khi ta có biểu thức $(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \rightarrow q$ là hằng đúng.

Ta gọi dạng lý luận trên là một luật suy diễn và người ta cũng thường viết luật suy diễn trên theo các cách sau đây

1. *Cách* 1. Biểu thức hằng đúng
$$(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \rightarrow q \equiv 1.$$
2. *Cách* 2. Dòng suy diễn
$$(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \equiv Q.$$
3. *Cách* 3. Mô hình suy diễn
$$\begin{array}{c} P_1 \\ \ldots \\ P_n \\ \hline \therefore Q \end{array}$$

Các biểu thức logic $P_1, P_2, \ldots, P_n$ trong luật suy diễn trên được gọi là giả thiết (hay tiền đề), và biểu thức Q được gọi là kết luận. Ở đây chúng ta cũng cần lưu ý rằng lý luận trên đúng không có nghĩa là ta có Q đúng và cũng không khẳng định rằng $P_1, P_2, \ldots, P_n$ đều đúng. Lý luận chỉ muốn khẳng định rằng nếu như ta có $P_1, P_2, \ldots, P_n$ là đúng thì ta sẽ có Q cũng phải đúng.

Ví dụ 1.27. Giả sử p và q là các biến logic. Xác định xem mô hình sau đây có phải là một luật suy diễn hay không?

$$\begin{array}{c} p \rightarrow q \\ p \\ \hline \therefore q \end{array}$$

Lời giải. Lập bảng chân trị ta có

p	q	$p \rightarrow q$	$(p \rightarrow q) \wedge p$	$((p \rightarrow q) \wedge p) \rightarrow q$
0	0	1	0	1
0	1	1	1	1
1	0	0	0	1
1	1	1	1	1

Bảng chân trị cho thấy biểu thức $((p \rightarrow q) \wedge p) \rightarrow q$ là hằng đúng. Do đó, mô hình suy luận trên đúng là một luật suy diễn. Thật ra, ta chỉ cần nhìn vào các cột chân trị của p, q, và $p \rightarrow q$ trong bảng chân trị là ta có thể kết luận được rồi, vì từ bảng chân trị trên ta thấy rằng nếu các giả thiết $p \rightarrow q$ và p đúng (có giá trị bằng 1) thì kết luận q cũng đúng.

Ta có thể khẳng định được mô hình suy luận trên là một luật suy diễn mà không cần lập bảng chân trị . Giả sử $p \rightarrow q$ và p đúng. Khi đó q phải đúng, bởi vì nếu ngược lại (q sai) thì p cũng phải sai (sẽ mâu thuẫn với giả thiết).

1.6.2. Kiểm tra một quy tắc suy diễn

Để kiểm tra một suy luận cụ thể là đúng hay không, tức là có "hợp logic" hay không, ta có thể căn cứ vào các qui tắc suy diễn (luật suy diễn). Phép suy luận cụ thể có thể được xem như sự suy diễn trên các mệnh đề phức hợp. Các mệnh đề sơ cấp cụ thể (mà chân trị có thể đúng hoặc sai) trong phép suy luận sẽ được trừu tượng hóa (thay thế) bởi các biến logic. Như thế phép suy luận được trừu tượng hóa thành một qui tắc suy diễn trên các biểu thức logic mà ta có thể kiểm tra xem qui tắc suy diễn là đúng hay không. Đây chính là biện pháp để ta biết được một suy luận cụ thể là đúng hay sai.

Ví dụ 1.28. Xét sự suy luận sau đây

Nếu một danh sách L là khác rỗng thì ta có thể lấy ra phần tử đầu trong danh sách. Vì ta không thể lấy ra phần tử đầu trong danh sách L nên danh sách L là danh sách rỗng.

Lời giải. Trong phép suy luận, ta có các mệnh đề sơ cấp $p =$"danh sách L là khác rỗng", $q =$"ta có thể lấy ra phần tử đầu (từ danh sách L)". Thay thế các mệnh đề sơ cấp này bởi các biến logic p, q tương ứng thì phép suy luận cụ thể trên sẽ được trừu tượng hóa thành một suy diễn trên các biểu thức logic như sau

$$\begin{array}{l} p \to q \\ \overline{p} \\ \hline \therefore \overline{p} \end{array}$$

Mô hình suy diễn này chính là qui tắc suy diễn Modus Tollens, đã được biết là đúng. Vậy phép suy luận trên là suy luận đúng.

Ví dụ 1.29. Xét xem suy luận sau đây có đúng hay không?

Nếu $\sqrt{2}$ là số hữu tĩ thì phương trình $m^2 = 2n^2$ có nghiệm nguyên dương m, n. Nếu phương trình $m^2 = 2n^2$ có nghiệm nguyên dương m và n thì ta có mâu thuẫn. Vậy $\sqrt{2}$ là số vô tĩ.

Lời giải. Trừu tượng hóa các mệnh đề sơ cấp $p =$"$\sqrt{2}$ là số hữu tĩ", $q =$" phương trình $m^2 = 2n^2$ có nghiệm nguyên dương m, n " thành các biến logic p, q tương ứng thì phép suy luận trên có dạng mộ hình suy diễn

$$\begin{array}{l} p \to q \\ q \to 0 \\ \hline \therefore \overline{p} \end{array}$$

Kiểm tra mô hình suy diễn này ta sẽ thấy là đúng. Như thế phép suy luận trên là đúng.

1.6.3. Các quy tắc suy diễn cơ bản

Trong mục này chúng ta nêu lên một số qui tắc suy diễn (đúng) thường được sử dụng mà ta có thể kiểm tra chúng bằng các phương pháp đã được trình bày trong mục trước.

1. **Qui tắc rút gọn:** Cơ sở của qui tắc là hằng đúng

$$(P \wedge Q) \to P$$

P	Q	$P \wedge Q$	$(P \wedge Q) \to P$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	1

hoặc là viết dưới dạng mô hình suy diễn

$$\begin{array}{l} P \\ Q \\ \hline \therefore P \end{array}$$

2. **Qui tắc suy diễn cộng:** Cơ sở của qui tắc là hằng đúng

$$P \to (P \vee Q)$$

P	Q	$P \vee Q$	$P \to (P \vee Q)$
0	0	0	1
0	1	1	1
1	0	1	1
1	1	1	1

hoặc là viết dưới dạng mô hình suy diễn

$$\begin{array}{l} P \\ \hline \therefore P \vee Q \end{array}$$

3. **Qui tắc khẳng định**
Cơ sở của qui tắc khẳng định là hằng đúng

$$(P \to Q) \wedge P \to Q$$

$$\begin{array}{l} P \to Q \\ P \\ \hline \therefore Q \end{array}$$

Ví dụ 1.30. Chứng minh công thức sau là hằng đúng

$$(p_1 \vee p_2) \wedge ((p_1 \vee p_2) \to \overline{p_3 \wedge p_4}) \to \overline{p_3 \wedge p_4}$$

Lời giải. Mô hình suy diễn của công thức trên

$$\frac{\begin{array}{l} p_1 \vee p_2 \\ (p_1 \vee p_2) \to \overline{p_3 \wedge p_4} \end{array}}{\therefore\ p_3 \vee p_4} \text{ (Kđ)} \equiv \frac{\overline{p_3 \wedge p_4}}{\therefore\ p_3 \wedge p_4} \equiv \mathbf{1}$$

Vậy, biểu thức trên là hằng đúng.

Ví dụ 1.31. Chỉ ra biểu thức sau là hằng đúng

$$(p \wedge q) \to (p \vee q)$$

Lời giải. Ta có mô hình suy diễn

$$\frac{p \wedge q}{\therefore\ p \vee q} \equiv \frac{\begin{array}{l} p \wedge q \\ p \wedge q \to p \end{array}}{\therefore\ p \vee q} \text{ (Kđ)} \equiv \frac{p}{\therefore\ p \vee q} \equiv \mathbf{1}$$

Vậy, biểu thức trên là hằng đúng.

4. **Qui tắc phủ định** Cơ sở của qui tắc phủ định là hằng đúng

$$(P \to Q) \wedge Q \to \overline{P}$$

hoặc là viết dưới dạng mô hình suy diễn

$$\frac{\begin{array}{l} P \to Q \\ \overline{Q} \end{array}}{\therefore\ \overline{P}}$$

Ví dụ 1.32. Nếu được thưởng cuối năm An sẽ đi Đà Lạt. Nếu đi Đà Lạt thì An sẽ thăm Thiền Viện. Mà Anh kgồn thăm thiền Viện. Vậy An không được thưởng cuối năm.

Suy luận của đoạn văn trên có đúng không?

Lời giải. Đặt

p_1 ="An được thưởng cuối năm";

p_2 ="An sẽ đi Đà Lạt";

p_3 ="An sẽ thăm Thiền Viện".

Đoạn văn trên được mô tả

$$\frac{\begin{array}{l} p_1 \to p_2 \\ \begin{cases} p_2 \to p_3 \\ \overline{p}_3 \end{cases} \end{array}}{\therefore\ \overline{p}_1} \text{ (Pđ)} \equiv \frac{\begin{array}{l} p_1 \to p_2 \\ \overline{p}_2 \end{array}}{\therefore\ \overline{p}_1} \text{ (Pđ)} \equiv \frac{\overline{p}_1}{\therefore\ \overline{p}_1} \equiv \mathbf{1}$$

Suy luận của đoạn văn trên là đúng.

5. **Tam đoạn luận** Cơ sở của qui tắc tam đoạn luận là hằng đúng

$$(P \to Q) \land (Q \to R) \land (P \to R)$$

hoặc là viết dưới dạng mô hình suy diễn

$$\begin{array}{l} P \to Q \\ Q \to R \\ \hline \therefore (P \to R) \end{array}$$

Ví dụ 1.33. Chỉ ra suy luận dưới đây là đúng

Bính đi chơi thì Bính không học toán rời rạc. Bình không học toán rời rạc thì bình thi trượt toán rời rạc. Mà Bình thích đi chơi. Vậy Bình thi trượt toán rời rạc.

Lời giải. Đặt
p_1 ="Bình đi chơi";
p_2 ="Bình không học toán rời rạc";
p_3 ="Bình thi trượt toán rời rạc".
Đoạn văn trên có thể viết dưới dạng mô hình suy diễn sau đây.

$$\begin{array}{l} \begin{cases} p_1 \to p_2 \\ p_2 \to p_3 \end{cases} \\ \overline{p}_1 \\ \hline \therefore \overline{p}_3 \end{array} \text{(Tđl)} \equiv \begin{array}{l} p_1 \to p_3 \\ \overline{p}_1 \\ \hline \therefore \overline{p}_3 \end{array} \text{(Sdc)} \equiv \begin{array}{l} \overline{p}_3 \\ \hline \therefore \overline{p}_3 \end{array} \equiv \mathbf{1}$$

Suy luận của đoạn văn trên là đúng.

6. **Tam đoạn luận rời** Cơ sở của qui tắc tam đoạn luận là hằng đúng

$$((P \lor Q) \land \overline{P}) \to P$$

hoặc là viết dưới dạng mô hình suy diễn

$$\begin{array}{l} P \lor Q \\ \overline{P} \\ \hline \therefore Q \end{array}$$

Ví dụ 1.34. Chỉ ra biểu thức dưới đây là hằng đúng

$$(p_1 \land (p_1 \to p_2) \land (p_3 \land p_4) \land (p_4 \to \overline{p}_2)) \to (p_3 \lor p_4).$$

Lời giải.

$$\dfrac{\begin{array}{l}\begin{cases} p_1 \\ p_1 \to p_2 \end{cases} \\ p_3 \vee p_4 \\ p_4 \to \overline{p}_2 \end{array}}{\therefore p_3 \vee p_5} \text{(Kđ)} \equiv \dfrac{\begin{array}{l} p_3 \vee p_4 \\ \begin{cases} p_4 \to \overline{p}_2 \\ p_2 \end{cases} \end{array}}{\therefore p_3 \vee p_5} \text{(Pđ)} \equiv$$

$$\dfrac{\begin{array}{l} p_3 \vee p_4 \\ \overline{p}_4 \end{array}}{\therefore p_3 \vee p_5} \text{(Tđlr)} \equiv \dfrac{p_3}{\therefore p_3 \vee p_5} \text{(Suy diễn cộng)} \equiv \mathbf{1}$$

Vậy suy luận đúng.

7. **Qui tắc phản chứng** Cơ sở của qui tắc phản chứng là hằng đúng

$$P \to Q \equiv (P \wedge \overline{Q}) \to 0$$

hoặc là viết dưới dạng mô hình suy diễn

$$\dfrac{P}{\therefore Q} \equiv \dfrac{\begin{array}{c} P \\ \overline{Q} \end{array}}{\therefore \mathbf{0}}$$

Qui tắc này cho phép ta chứng minh $(P \wedge \overline{Q}) \to 0$ thay cho $P \to Q$. Nói cách khác, nếu ta thêm giả thiết phụ vào tiền đề P mà chứng minh được có sự mâu thuẫn thì ta có thể kết luận q từ tiền đề P.

Ví dụ 1.35. Nếu An được lên chức và làm việc nhiều thì An được tăng lương. An được tăng lương thì An mua xe máy. Mà An không mua được xe máy. Vậy An không được lên chức hay An không làm việc nhiều.

Suy luận trên có đúng không?

Lời giải. Đặt
p_1 ="An được lên chức";
p_2 ="An làm việc nhiều";
p_3 ="An được tăng lương";
p_4 ="An mua xe máy";
Suy luận trên tương đương với

$$\frac{\begin{array}{l}(p_1 \wedge p_2) \to p_3\\ p_3 \to p_4\\ \overline{p}_4\end{array}}{\therefore \overline{p}_1 \vee \overline{p}_2}\text{(pc)}\equiv \frac{\begin{array}{l}(p_1 \wedge p_2) \to p_3\\ \begin{cases} p_3 \to p_4\\ \overline{p}_4\end{cases}\\ p_1 \wedge p_2\end{array}}{\therefore \mathbf{0}}\text{(Pđ)}\equiv$$

$$\frac{\begin{array}{l}\begin{cases}(p_1 \wedge p_2) \to p_3\\ \overline{p}_3\end{cases}\\ p_1 \wedge p_2\end{array}}{\therefore \mathbf{0}}\text{(Pđ)}\equiv \frac{\begin{array}{l}\overline{p_1 \wedge p_2}\\ p_1 \wedge p_2\end{array}}{\therefore \mathbf{0}} \equiv \frac{\mathbf{0}}{\therefore \mathbf{0}} \equiv \mathbf{1}$$

8. **Qui tắc từng trường hợp** Cơ sở của qui tắc từng trường hợp là hằng đúng

   $(P_1 \to Q) \wedge (P_2 \to Q) \wedge \ldots \wedge (P_n \to Q) \equiv (P_1 \vee P_2 \vee \ldots \vee P_n) \to Q$

   hoặc là viết dưới dạng mô hình suy diễn

$$\frac{\begin{array}{l}P_1 \to Q\\ P_2 \to Q\\ \ldots\\ P_n \to Q\end{array}}{\therefore (P_1 \vee P_2 \vee \ldots \vee P_n) \to Q}$$

1.6.4. Các ví dụ áp dụng

Dưới đây ta trình bày chứng minh của một số mệnh đề mà không nêu lên một cách chi tiết về các qui tắc suy diễn đã được áp dụng. Người đọc có thể tìm thấy các qui tắc suy diễn được sử dụng trong chứng minh một cách dễ dàng.

▷ **1.1.** Chỉ ra biểu thức sau đây là hằng đúng

$$((a \to b) \wedge (b \to c) \wedge (d \vee \overline{c}) \wedge (\overline{d} \vee e) \wedge \overline{e}) \to \overline{a}$$

▷ **1.1.** Thay $d \vee \overline{c} \equiv c \to d$ và $\overline{d} \vee e \equiv d \to e$. Công thức trên có mô hình suy diễn sau

$$\cfrac{\begin{cases} a \to b \\ b \to c \\ c \to d \\ d \to e \end{cases} \\ \overline{e}}{\therefore \overline{a}} \text{ (Tđl)} \equiv \cfrac{\begin{array}{l} a \to e \\ \overline{e} \end{array}}{\therefore \overline{a}} \text{ (Pđ)} \equiv \cfrac{\overline{a}}{\therefore \overline{a}} \equiv \mathbf{1}$$

▷ **1.2.** Suy luận dưới đây có đúng không

$$\cfrac{\begin{array}{l} a \to b \\ b \to c \\ c \to d \\ a \end{array}}{\therefore d} \text{ (Tđl)} \equiv \cfrac{\begin{array}{l} a \to d \\ a \end{array}}{\therefore d} \text{ (Kđ)} \equiv \cfrac{d}{\therefore d} \equiv \mathbf{1}$$

▷ **1.2.** Suy luận đúng.

▷ **1.3.** Chỉ ra công thức dưới đây là hằng đúng

$$((A \to B) \wedge (\overline{A} \to C) \wedge (C \to D)) \to (\overline{B} \to D)$$

▷ **1.3.** Vì $\overline{\overline{B} \to D} \equiv \overline{B} \wedge \overline{D}$nên áp dụng quy tắc mâu thuẫn ta có suy luân sau

$$\cfrac{\begin{array}{l} A \to B \\ \overline{A} \to C \\ C \to D \end{array}}{\therefore \overline{B} \to D} \equiv \cfrac{\begin{array}{l} A \to B \\ \overline{A} \to C \\ C \to D \\ \overline{B} \\ \overline{D} \end{array}}{\therefore \mathbf{0}} \equiv \cfrac{\begin{array}{l} \begin{cases} A \to B \\ \overline{B} \end{cases} \\ \overline{A} \to C \\ \begin{cases} C \to D \\ \overline{D} \end{cases} \end{array}}{\therefore \mathbf{0}} \equiv \cfrac{\begin{array}{l} \begin{cases} \overline{A} \\ \overline{A} \to C \end{cases} \\ \overline{C} \end{array}}{\therefore \mathbf{0}} \text{ (Kđ)} \equiv$$

$$\equiv \cfrac{\begin{array}{l} \overline{C} \\ C \end{array}}{\therefore \mathbf{0}} \equiv \cfrac{C \wedge \overline{C}}{\therefore \mathbf{0}} \equiv \cfrac{\mathbf{0}}{\therefore \mathbf{0}} \equiv \mathbf{1}$$

Vậy công thức trên là hằng đúng.

▷ **1.4.** Chỉ ra suy luận dưới đây là đúng

$$\begin{array}{l} x \to y \\ \overline{x} \to z \\ z \to w \\ \hline \therefore \overline{y} \to w \end{array}$$

▷ **1.4.** Dùng quy tắc mâu thuẫn ta có biểu thức tương đương

$$\begin{array}{l} x \to y \\ \begin{cases} \overline{x} \to z \\ z \to w \end{cases} \\ \overline{y} \wedge \overline{w} \\ \hline \therefore \mathbf{0} \end{array} (\text{Tđl}) \equiv \begin{array}{l} \begin{cases} x \to y \\ \overline{y} \end{cases} \\ \begin{cases} \overline{x} \to w \\ \overline{w} \end{cases} \\ \hline \therefore \mathbf{0} \end{array} (\text{Pđ}) \equiv$$

$$\equiv \begin{array}{c} \overline{x} \\ x \\ \hline \therefore \mathbf{0} \end{array} \equiv \begin{array}{c} \overline{x} \wedge x \\ \hline \therefore \mathbf{0} \end{array} \equiv \begin{array}{c} \mathbf{0} \\ \hline \therefore \mathbf{0} \end{array} \equiv \mathbf{1}$$

▷ **1.5.** Suy luận dưới đây có đúng không? "Nếu muốn đi họp sáng thứ ba thì An phải dậy sớm. Nếu An đi nghe nhạc tối thứ hai thì An sẽ về muộn. Nếu An về muộn và thức dậy sớm thì An phải đi họp sáng thứ ba và chỉ được ngủ dưới 7 giờ trong ngày. Nhưng An không thể đi họp nếu chỉ đi ngủ dưới 7 giờ trong ngày. Vậy hoặc An không đi nghe nhạc tối thứ hai hoặc An phải bỏ họp sáng thứ ba."

▷ **1.5.** Đặt
p_1 ="An muốn đi họp sáng thứ ba";
p_2 ="An phải dậy sớm";
p_3 ="An đi nghe nhạc tối thứ hai";
p_4 ="An sẽ về muộn";
p_5 ="An nghủ dưới 7 giờ trong một ngày".
Khi đó suy luận trên tương đương với

$$\begin{array}{l} p_1 \to p_2 \\ p_3 \to p_4 \\ (p_4 \wedge p_2) \to (p_1 \wedge p_5) \\ p_5 \to \overline{p}_1 \\ \hline \therefore \overline{p}_3 \vee \overline{p}_1 w \end{array} \text{(Mt)}\equiv \begin{array}{l} p_1 \to p_2 \\ p_3 \to p_4 \\ (p_4 \wedge p_2) \to (p_1 \wedge p_5) \\ p_5 \to \overline{p}_1 \\ \overline{\overline{p}_3 \vee \overline{p}_1} \\ \hline \therefore \mathbf{0} \end{array} \equiv$$

$$\begin{array}{l} \begin{cases} p_1 \to p_2 \\ p_1 \end{cases} \\ \begin{cases} p_3 \to p_4 \\ p_3 \end{cases} \\ (p_4 \wedge p_2) \to (p_1 \wedge p_5) \\ p_5 \to \overline{p}_1 \\ \hline \therefore \mathbf{0} \end{array} \text{(Kđ)}\equiv \begin{array}{l} \begin{cases} \\ \\ p_4 \wedge p_2 \\ (p_4 \wedge p_2) \to (p_1 \wedge p_5) \end{cases} \\ \begin{cases} p_5 \to \overline{p}_1 \\ p_1 \equiv \overline{\overline{p}_1} \end{cases} \\ \hline \therefore \mathbf{0} \end{array}$$

$$\text{(Kđ và phủ định)}\equiv \begin{array}{l} p_1 \wedge p_5 \\ \overline{p}_5 \\ \hline \therefore \mathbf{0} \end{array} \equiv \begin{array}{c} p_5 \wedge \overline{p}_5 \\ \hline \therefore \mathbf{0} \end{array} \equiv \begin{array}{c} \mathbf{0} \\ \hline \therefore \mathbf{0} \end{array} \equiv \mathbf{1}$$

▷ **1.6.** "Nếu An đi làm về muộn thì vợ An sẽ rất giận dữ. Nếu Bình thường xuyên vắng nhà thì vợ Bình cũng sẽ rất giận dữ. Nếu vợ Bình hoặc vợ An giận dữ thì cô Hà bạn họ nhận được than phiền, mà cô Hà không hề nhận được lời than phiền. Vậy An đi làm về sớm và Bình rất ít khi vắng nhà."

Dùng quy tắc suy diễn để chỉ ra suy luận trên là đúng.

▷ **1.6.** Đặt
p_1 ="An đi làm về muộn";
p_2 ="Vợ An sẽ rất giận dữ";
p_3 ="Bình thường xuyên vắng nhà";
p_4 ="Vợ Bình cũng rất giận dữ";
p_5 ="Cô Hà bạn họ nhận được lời than phiền".
Suy luận trên tương đương với

$$\cfrac{\begin{array}{l} p_1 \to p_2 \\ p_3 \to p_4 \\ \begin{cases} (p_2 \vee p_4) \to p_5 \\ \overline{p}_5 \end{cases} \end{array}}{\therefore \overline{p}_1 \vee \overline{p}_3} \text{(Pđ)} \equiv \cfrac{\begin{array}{l} p_1 \to p_2 \\ \begin{cases} \cfrac{p_3 \to p_4}{(p_2 \vee p_4)} \end{cases} \end{array}}{\therefore \overline{p}_1 \vee \overline{p}_3} \equiv$$

$$\cfrac{\begin{cases} \begin{cases} p_1 \to p_2 \\ \overline{p}_2 \end{cases} \\ \begin{cases} p_3 \to p_4 \\ \overline{p}_4 \end{cases} \end{cases}}{\therefore \overline{p}_1 \vee \overline{p}_3} \text{(Pđ)} \equiv \cfrac{\overline{p}_1 \vee \overline{p}_3}{\therefore \overline{p}_1 \vee \overline{p}_3} \equiv \mathbf{1}$$

▷ **1.7.** Chỉ ra suy luận dưới đây là sai

"Ông minh đã khẳng định rằng, nếu không được tăng lương thì ông ta sẽ xin nghỉ việc. Mặt khác, nếu ông Minh nghỉ việc mà vợ ông ta bị mất việc thì phải bán xe máy. Biết rằng, nếu vợ ông Minh hay đi làm muộn thì sẽ mất việc và cuối cùng ông Minh đã được tăng lương. Vậy nếu ông Minh không bán xe thì vợ ông ta không đi làm muộn."

▷ **1.7.** Đặt

p_1 ="Ông Minh được tăng lương";
p_2 ="Ông Minh xin nghỉ việc";
p_3 ="Vợ ông Minh bị mất việc";
p_4 ="Ông Minh phải bán xe";
p_5 ="Vợ ông Minh đi làm muộn";
Đoạn văn trên tương đương với mô hình suy diễn sau

$$\cfrac{\begin{array}{l} \overline{p}_1 \to p_2 \\ (p_2 \wedge p_3) \to p_4 \\ p_5 \to p_3 \\ p_1 \end{array}}{\therefore \overline{p}_4 \to p_5}$$

Chọn $p_4 = 0$ và $p_5 = 1$ thì kết luận sai.

Từ $p_4 = 0$ và $p_5 = 1$ ta có $p_3 = 1, p_2 = 0$ và $p_1 = 1$. Với những giá trị nàythì giả thiết đúng. Vậy giả thiết đúng kết luận sai thì mô hình trên là sai. Hay đoạn văn trên là sai.

▷ **1.8.** Tìm phản ví dụ cho ví dụ dưới đây

a) $\begin{array}{l} X \equiv Y \\ Y \to Z_1 \\ Z_1 \vee \overline{Z}_2 \\ \overline{Z}_2 \to Y \\ \hline \therefore Z_2 \end{array}$ b) $\begin{array}{l} X_1 \\ X_1 \to X_2 \\ X_1 \to (X_3 \vee \overline{X}_2) \\ \overline{X}_3 \vee \overline{X}_4 \\ \hline \therefore X_4 \end{array}$

▷ **1.8.** a) Chọn $X \equiv Y = 1$ ta suy ra $Z_1 = 1$ và $Z_2 = 0$. Như vậy giả thiết đúng kết luận sai. Hay suy luận trên không đúng.

b) Suy luận này sai vì chọn $X_1 = X_2 = X_3 = 1$ và $X_4 = 0$. Khi đó giả thiết đúng kết luận sai.

Ví dụ 1.36. Chứng minh rằng với mọi số nguyên $n, n^3 + 2n$ chia hết cho 3.

Lời giải. *Phân tích.* Suy nghĩ đầu tiên là ta thấy rằng không thể tìm thấy một thừa số 3 trong biểu thức $n^3 + 2n$. Nhưng khi phân tích ra thừa số thì $n^3 + 2n = n(n^2 + 2)$. Phát biểu "$n^3 + 2n$ chia hết cho 3" sẽ đúng nếu n là bội số của 3. Còn các trường hợp khác thì sao?. Ta thử phương pháp phân chứng.

Chứng minh.
Ta có $n^3 + 2n = n(n^2 + 2)$, và số tự nhiên n có một trong 3 dạng ứng với 3 trường hợp dưới đây

- *Trường hợp 1.* $n = 3k$, với k là một số nguyên.
$$n^3 + 2n = 3k(9k^2 + 2) \text{ chia hết cho 3.}$$
- *Trường hợp 2.* $n = 3k + 1$, với k là một số nguyên.
$$\begin{aligned} n^3 + 2n &= (3k+1)((3k+1)2+2) \\ &= (3k+1)(9k^2+6k+3) \\ &= (3k+1)3(3k2+2k+1) \text{ chia hết cho 3.} \end{aligned}$$
- *Trường hợp 3.* $n = 3k + 2$, với k là một số nguyên.
$$\begin{aligned} n^3 + 2n &= (3k+2)((3k+2)^2+2) \\ &= (3k+1)(9k^2+12k+6) \\ &= (3k+1)3(3k^2+4k+2) \text{ chia hết cho 3.} \end{aligned}$$

Trong mọi trường hợp (có thể có) ta đều có $n^3 + 2n$ đều chia hết cho 3.

Vậy ta kết luận $n^3 + 2n$ chia hết cho 3 đối với mọi số nguyên n.

Nhận xét. Chứng minh trên có thể được trình bày ngắn gọn hơn bằng cách sử dụng phép đồng dư modulo 3.

Ví dụ 1.37. Chứng minh rằng nếu n^2 là một số chẳn thì n cũng là một số chẳn.Nếu n^2 là một số chẳn thì n cũng là một số chẳn.

Lời giải. *Phân tích.* Giả sử $n^2 = 2k$ (là số chẳn). Ta thấy khó suy ra n là số chẳn. Nếu biết thông tin gì đó về n thì suy ra điều gì đó về n^2 thì dễ hơn. Ta thử phương pháp phản chứng.

Chứng minh. Ta hãy chứng minh mệnh đề "Nếu n lẻ thì n^2 lẻ". Cho n là một số lẻ, ta có $n = 2k + 1$ (k là một số nguyên). Do đó $n^2 = (2k + 1)^2 = 4k^2 + 4k + 1$ là một số lẻ.

Mệnh đề trong cặp nháy kép là đúng nên mệnh đề phản đảo của nó cũng đúng. Vậy, nếu n^2 là một số chẳn thì n cũng là một số chẳn.

Ví dụ 1.38. Chứng minh rằng nếu $p > 3$ và p nguyên tố thì $p^2 - 1$ chia hết cho 3.

Lời giải. Ta có $(p - 1), p, (p + 1)$ là 3 số nguyên liên tiếp. Trong 3 số nguyên này có một số chia hết cho 3. Nhưng số đó không phải là p vì p là số nguyên tố lớn hơn 3. Do đó $(p - 1)$ chia hết cho 3 hoặc $(p + 1)$ chia hết cho 3. Suy ra $(p - 1)(p + 1)$ chia hết cho 3, tức là $p^2 - 1$ chia hết cho 3.

Ví dụ 1.39. Chứng minh rằng số lượng các số nguyên tố là vô hạn.

Lời giải. Giả sử phát biểu trong mệnh đề là sai. Tức là chỉ có một số hữu hạn, k, số nguyên tố (dương). Ký hiệu k số nguyên tố là $p_1, p_2, \ldots \ldots, p_k$, ở đây k là số nguyên dương. Đặt $n = p^1 p^2 \ldots p_k +$ 1. Số n lớn hớn tất cả k số nguyên tố nên n không nguyên tố. Do đó, từ định lý cơ bản của số học , n phải có một ước số nguyên tố p. p phải là một trong k số nguyên tố. Do đó $p|(p_1p_2\dots p_k)$. Suy ra $p|(n - p_1p_2\dots p_k)$, hay $p|1$.

Như thế, ta có p là một số nguyên tố và $p|1$. Điều này là không thể, hay nói cách khác, ta có một mâu thuẫn.

Vậy, Số lượng các số nguyên tố là vô hạn.

1.7 Bài tập

▷ **1.9.** Trong các phát biểu sau, cho biết phát biểu nào là mệnh đề. Khi là mệnh đề, cho biết chân trị của nó

a) 11 là số nguyên chẵn.

b) 23 là số nguyên tố.

c) $x - 2y = 10$.

d) $\log_2 3 > \log_3 2$.

e) $\sin\dfrac{\pi}{2} = 1$ và $4 > 5$.

f) nếu $2 + 3 = 4$ thì Hồ Chí Minh và Trần Hưng đạo là một người.

▷ **1.10.** Đặt P, Q lần lượt là các mệnh đề
P := " Minh học chăm",
Q:= " Minh có kết quả học tập tốt",
R:= " Minh học giỏi môn Toán".
Hãy việt lại các mệnh đề sau dưới dạng hình thức trong đó có sử dụng các phép nối.
a) Minh học chăm và có kết quả học tập tốt.
b) Minh học chăm nhưng không có kết quả học tập tốt.
c) Minh học chăm hay Minh có kết quả học tập tốt.
d) Nếu Minh học chăm thì Minh có kết quả học tập tốt.
e) Minh có kết quả học tập tốt khi và chỉ khi Minh học chăm.

▷ **1.11.** Lập bảng chân trị cho các dạng mệnh đề sau và cho biết dạng mệnh đề nào là hằng

a) $p \to (p \vee q)$
b) $p \to (p \wedge q)$
c) $(p \wedge q) \to p$
d) $(p \wedge q) \to (p \vee q)$
e) $(p \vee q) \to (p \wedge q)$
f) $(p \to q) \leftrightarrow (\overline{p} \vee q)$

▷ **1.12.** Mệnh đề nào dưới đây là hằng đúng

a) $p \Rightarrow (p \vee q)$
b) $p \Rightarrow (p \wedge q)$
c) $q \Rightarrow (p \to q)$
d) $\overline{p \to q} \Rightarrow p$
e) $(p \to q) \equiv (-p \vee q)$
f) $p \wedge (p \to q) \Rightarrow (p \wedge q)$.

▷ **1.13.** Sử dụng logic mệnh đề để giải bài toàn sau : Trong một phiên tòa xử án 3 bị can có liên quan đến vấn đề tài chánh, trước tòa cả 3 bị cáo đều tuyên thệ khai đúng sự thật và lời khai như sau
Anh A: Chị B có tội và anh C vô tội.
Chị B : Nếu anh A có tội thì anh C cũng có tội.
Anh C: Tôi vô tội nhưng một trong hai người kia là có tội.
Hãy xét xem ai là người có tội ?

▷ **1.14.** Cho các mệnh đề được phát biểu như sau, hãy tìm số lớn nhất các mệnh đề đồng thời là đúng.
a) Quang là người khôn khéo
b) Quang không gặp may mắn
c) Quang gặp may mắn nhưng không khôn khéo
d) Nếu Quang là người khôn khéo thì nó không gặp may mắn
e)Quang là người khôn khéo khi và chỉ khi nó gặp may mắn
f) Hoặc Quang là người khôn khéo, hoặc nó gặp may mắn nhưng không đồng thời cả hai.

▷ **1.15.** Cho a và b là hai số nguyên dương. Biết rằng, trong 4 mệnh đề sau đây có 3 mệnh đề đúng và 1 mệnh đề sai. Hãy tìm mọi cặp số (a, b) có thể có.
1) $a + 1$ chia hết cho b
2) $a = 2b + 5$
3) $a + b$ chia hết cho 3
4) $a + 7b$ là số nguyên tố

▷ **1.16.** Không lập bảng chân trị, sử dụng các công thức tương đương logic, chứng minh rằng các biểu thức mệnh đề sau là hằng đúng

a) $(P \wedge Q) \to P$

b) $P \to (\overline{P} \to P)$

c) $P \to ((Q \to (P \wedge Q))$

d) $\overline{P \vee \overline{Q}} \to \overline{P}$

e) $((P \to Q) \wedge (Q \to R)) \to (P \to R)$

▷ **1.17.** Không lập bảng chân trị, sử dụng các công thức tương đương logic, xét xem biểu thức mệnh đề G có là hệ quả của F không ?

a) $F = P \wedge (Q \vee R), G = (P \wedge Q) \vee R$

b) $F = (P \to Q) \wedge (Q \to R), G = P \to (Q \to R)$

c) $F = P \wedge Q, G = (\overline{P} \to Q) \vee (P \to \overline{Q})$

▷ **1.18.** Tương tự bài tập 16 và 17, chứng minh các tương đương logic sau đây

a) $(P \vee Q) \wedge \overline{\overline{P} \wedge Q} \equiv P$

b) $\overline{\overline{(P \vee Q) \wedge R} \vee \overline{Q}} \equiv Q \wedge R$

c) $((P \vee Q) \wedge (P \vee \overline{Q})) \vee Q \equiv P \vee Q$

d) $\overline{P \vee Q} \vee ((\overline{P} \wedge Q) \vee \overline{Q}) \equiv \overline{Q \wedge P}$

f) $P \vee (P \wedge (P \vee Q) \equiv P$

g) $P \vee Q \vee (\overline{P} \wedge \overline{Q} \wedge R) \equiv P \vee Q \vee R$

h) $((\overline{P} \vee \overline{Q}) \to (P \wedge Q \wedge R) \equiv P \wedge Q$

i) $P \wedge ((\overline{Q} \to (R \wedge R)) \vee \overline{Q \vee (R \wedge S) \vee (R \wedge \overline{S})}) \equiv P$

j) $(P \vee Q \vee R) \wedge (P \vee S \vee \overline{Q}) \wedge (P \vee \overline{S} \vee R) \equiv P \vee (R \wedge (S \vee \overline{Q})$

▷ **1.19.** Sau khi nướng 1 chiếc bánh cho 2 đứa cháu trai và 2 đứa cháu gái đến thăm, Dì Nellie lấy bánh ra khỏi lò nướng và để nguội. Sau đó, cô rời khỏi nhà để đến đóng cửa hàng ở gần đó. Lúc trở về thì có ai đó đã ăn 1/4 chiếc bánh và thậm chí còn đặt lại cái dĩa dơ bên phần bánh còn lại. Vì không còn ai đến nhà Dì ngày hôm đó trừ 4 đứa cháu nên Dì biết ngay là 1 trong 4 đứa đã ăn mà chưa được cho phép. Dì Nellie bèn hỏi 4 đứa thì được các

câu trả lời như sau:
- Charles : Kelly đã ăn phần bánh
- Dawn : Con không ăn bánh
- Kelly : Tyler ăn bánh
- Tyler : Con không ăn, Kelly nói chơi khi bảo rằng con ăn bánh.

Nếu chỉ 1 trong 4 câu trả lời trên là đúng và chỉ 1 trong 4 đứa cháu là thủ phạm, hãy tìm ra người mà Dì Nellie phải phạt ?

▷ **1.20.** Kiểm tra tính tương đương logic của 2 mệnh đề sau :
$F = P \to Q, G = \overline{P \vee Q}$

▷ **1.21.** a. Nếu biết mệnh đề $P \to Q$ là sai, hãy cho biết chân trị của các mệnh đề sau:

$$P \wedge Q, \quad \overline{P} \vee Q, \quad Q \to P$$

b. Cho các biểu thức mệnh đề sau:

$$((P \wedge Q) \wedge R) \to (S \vee M)$$

Xác định chân trị của các biến mệnh đề P, Q, R, S, M nếu các biểu thức mệnh đề trên là sai.

▷ **1.22.** Nếu Q có chân trị là T, hãy xác định chân trị của các biến mệnh đề P, R, S nếu biểu thức mệnh đề sau cũng là đúng

$$(Q \to ((\overline{P} \vee R) \wedge \overline{S})) \wedge (\overline{S} \to (\overline{R} \wedge Q))$$

Chương 2

LÔGIC VỊ TỪ

2.1 Vị từ và lượng từ

2.1.1. Định nghĩa vị từ và ví dụ

Định nghĩa 2.1. *Một vị từ* là một phát biểu $P(x, y, \ldots)$ phụ thuộc theo các biến $x, y, \ldots$ lấy giá trị trên các miền xác định $A, B, \ldots$ nào đó. Khi thay thế các biến trong vị từ bởi các giá trị cụ thể $a, b, \ldots$ thuộc các miền xác định thì ta được một mệnh đề $P(a, b, \ldots)$ có chân trị đúng (1) hoặc sai (0).

Gọi Boole B là tập hợp gồm có hai giá trị : Sai (ký hiệu bởi 0), và Đúng (ký hiệu bởi 1).

Ví dụ 2.1. $P(n) =$ "n là một số nguyên tố" là một vị từ trên tập hợp các số tự nhiên (hoặc trên tập hợp các số nguyên). Ta có thể thấy rằng

- $P(1) = 0$, tức là $P(1) =$"1 là một số nguyên tố" là một mệnh đề sai.
- $P(2) = 1$, tức là $P(2) =$"2 là một số nguyên tố" là một mệnh đề đúng.
- $P(12) = 0$, tức là $P(12) =$"12 là một số nguyên tố" là một mệnh đề sai.
- $P(17) = 1$, tức là $P(17) =$"17 là một số nguyên tố" là một mệnh đề đúng.

Vị từ "n là một số nguyên tố" có thể được xem là một ánh xạ đi từ tập hợp các số tự nhiên $\mathbb{N}$ vào tập hợp Boole B: $P : \mathbb{N} \to B$

$$P(n) = \begin{cases} 1 & \text{khi } n \text{ là nguyên tố,} \\ 0 & \text{khi } n \text{ không là nguyên tố.} \end{cases}$$

Ví dụ 2.2. $P(m, n) =$"m là một ước số của n", với m và n là các biến số tự nhiên, cho ta một vị từ theo 2 biến m và n thuộc tập hợp các số tự nhiên. Ta có

$$P(2, 4) = 1; P(3, 4) = 0.$$

2.1.2. Các phép toán trên vị từ

Cho $P(x, y, \ldots)$ là một vị từ theo các biến $x, y, \ldots$. *Phủ định* của P, ký hiệu là $\overline{P}$, là một vị từ mà khi thay các biến $x, y, \ldots$ bởi các phần tử cụ thể $a, b, \ldots$ tương ứng thì ta được mệnh đề $\overline{P(a, b, \ldots)}$. Nói một cách khác, vị từ $\overline{P}$ được định nghĩa bởi

$$(\overline{P})(x, y, \ldots) = \overline{P(x, y, \ldots)}.$$

Cho $P(x, y, \ldots)$ và $Q(x, y, \ldots)$ là các vị từ theo các biến $x, y, \ldots$. *Phép hội* của P và Q, ký hiệu là $P \wedge Q$, là một vị từ mà khi thay các biến $x, y, \ldots$ bởi các phần tử cụ thể $a, b, \ldots$ tương ứng thì ta được mệnh đề $P(a, b, \ldots) \wedge Q(a, b, \ldots)$. Nói một cách khác, vị từ $P \wedge Q$ được định nghĩa bởi

$$(P \wedge Q)(x, y, \ldots) = P(x, y, \ldots) \wedge Q(x, y, \ldots).$$

Một cách tương tự, các *phép toán tuyển*, *kéo theo* và *tương đương* của 2 vị từ P và Q có thể được định nghĩa như sau:

- $(P \vee Q)(x, y, \ldots) = P(x, y, \ldots) \vee Q(x, y, \ldots)$.
- $(P \rightarrow Q)(x, y, \ldots) = P(x, y, \ldots) \rightarrow Q(x, y, \ldots)$.
- $(P \leftrightarrow Q)(x, y, \ldots) = P(x, y, \ldots) \leftrightarrow Q(x, y, \ldots)$.

2.1.3. Lượng từ và mệnh đề có lượng từ

Ngoài việc thay thế giá trị cụ thể cho các biến trong vị từ để được một mệnh đề ta còn có một cách quan trọng khác để chuyển từ vị từ sang mệnh đề. Đó là cách sử dụng các lượng từ "*với mọi*" và "*tồn tại*" (hay "*có ít nhất một*"). Lượng từ được sử dụng để nói lên rằng vị từ đúng đối với mọi giá trị thuộc miền xác định hay chỉ đúng với một phần các giá trị thuộc miền xác định.

Cho $P(n)$ là một vị từ theo biến số tự nhiên n. Phát biểu "với mọi $n \in \mathbb{N}, P(n)$" hay một cách vắn tắt (hiểu ngầm miền xác định) là "với mọi n, $P(n)$" có nghĩa là P có giá trị đúng trên toàn bộ miền xác định. Nói cách khác, P là ánh xạ hằng có giá trị là 1. Ta sẽ dùng ký hiệu "$\forall$" để thay thế cho lượng từ "*với mọi*".

Phát biểu "Có (ít nhất) một $n \in \mathbb{N}, P(n)$" hay một cách vắn tắt (hiểu ngầm miền xác định) là "Có (ít nhất) một n , $P(n)$" có nghĩa là P có giá trị đúng đối với một hay một số giá trị nào đó thuộc miền xác định. Nói cách khác, P không phải là một ánh xạ hằng 0. Ta sẽ dùng ký hiệu "$\exists$" để thay thế cho lượng từ "có ít nhất một". Lượng từ này còn được đọc một cách khác là "*tồn tại*".

Định nghĩa 2.2. Giả sử $P(x)$ là một vị từ theo biến x (biến x lấy giá trị thuộc một miền xác định đã biết nào đó và miền xác định này có thể được hiểu ngầm, không cần ghi rõ ra). Các cách viết sau đây:

$$\forall x : P(x) \tag{2.1}$$

$$\exists x : P(x) \tag{2.2}$$

lần lượt được dùng để diễn đạt cho các phát biểu sau đây:

"Với mọi x (thuộc miền xác định) ta có $P(x)$ là đúng"

"Có ít nhất mộ x (thuộc miền xác định) sao cho $P(x)$ là đúng".

Ký hiệu $\forall$ và $\exists$ được gọi là *lượng từ với mọi* và *lượng từ tồn tại*.

Các phát biểu (2.1) và (2.2) có chân trị hoàn toàn xác định. Nói cách khác chúng là những mệnh đề. Chân trị của các mệnh đề này được xác định một cách tự nhiên theo ngữ nghĩa thông thường của các lượng từ. Mệnh đề (2.1) là đúng khi và chỉ khi ứng với mỗi giá trị tùy ý x thuộc miền xác định ta đều có mệnh đề $P(x)$ có chân trị đúng. Mệnh đề (2.2) là đúng khi và chỉ khi có một giá trị x nào đó thuộc miền xác định mà ứng với giá trị x đó ta có $P(x)$ có chân trị đúng.

Chú ý.

- Phát biểu "$\forall x : P(x)$" và phát biểu "$\exists x : P(x)$" không phải là vị từ theo biến x nữa mà là các mệnh đề có chân trị xác định là đúng hoặc sai. Trong các phát biểu trên biến x đã được lượng từ hóa và chân trị của phát biểu không phụ thuộc theo biến x nữa. Ta cũng nói rằng biến x bị ràng buộc bởi lượng từ.
- Đối với một vị từ theo nhiều biến thì ta có thể *lượng từ hóa* một số biến nào đó trong vị từ để có một vị từ mới theo các biến còn lại. Chẳng hạn, nếu $P(x, y, \ldots)$ là một vị từ theo các biến $x, y, \ldots$ thì ta có biểu thức

  $$Q(y, \ldots) \equiv \forall x : P(x, y, \ldots)$$

  sẽ là một vị từ theo các biến $y, \ldots$.

  Nếu tất cả các biến của vị từ đều được lượng từ hóa thì ta sẽ có một mệnh đề.Chẳng hạn, nếu $P(x, y)$ làmột vị từ theo 2 biến x, y thì biểu thức$\forall x, \exists y : P(x, y)$ sẽ là một mệnh đề, tức là có chân trị xác định và không phụ thuộc vào các biến x, y nữa.

- Trong nhiều phát biểu người ta cò dùng cụm từ "tồn tại duy nhất", ký hiệu bởi $\exists!$, như là một sự *lượng từ hóa đặc biệt*.

Ví dụ 2.3. 1. Cho vị từ $P(n) =$"n là một số nguyên tố". Mệnh đề "Với mọi số tự nhiên n ta có n là nguyên tố" có thể được viết như sau:

$$\forall n \in \mathbb{N} : P(n)$$

và mệnh đề này có chân trị là 0 (sai).

2. Mệnh đề "Với mọi số nguyên n ta có $2n - 1$ là một số lẻ" có thể được viết dưới dạng ký hiệu như sau

$$\forall n \in \mathbb{Z} : 2n - 1 \text{ lẻ}$$

và mệnh đề này có chân trị là 1 (đúng).

Ta ký hiệu một vị từ $P(n) =$"$2n - 1$ là một số lẻ", khi đó ta có thể viết

$$\forall n \in \mathbb{Z} : P(n).$$

3. Mệnh đề "Ta có $x^2 > 0$, với mọi số thực x khác 0" có thể được viết là

$$\forall x \in \mathbb{R} \setminus \{0\} : x^2 > 0$$

và mệnh đề này có chân trị là 1 (đúng).

Ta cũng có thể ký hiệu $Q(x) =$"$x^2 > 0$", khi đó

$$\forall x \in \mathbb{R} \setminus \{0\} : Q(x).$$

Ví dụ 2.4. Chứng minh rằng nếu n là một số chẵn thì n^2 là số chẵn.

Mệnh đề cần chứng minh (là đúng) được viết dưới dạng

$$\forall n \in \mathbb{Z} : n \text{ chẵn } \rightarrow n^2 \text{ chẵn.}$$

Từ đó ta có thể trình bày chứng minh như sau: Cho n là một số nguyên tùy ý. Ta có n chẵn suy ra $n = 2m$, với m là một số nguyên nào đó suy ra $n^2 = 4m^2$, suy ra n^2 chẵn. Vậy phát biểu trên là đúng.

2.1.4. Quy tắc phủ định mệnh đề có lượng từ

Dựa vào cách xác định chân trị của các mệnh đề có lượng từ theo ngữ nghĩa tự nhiên của các phát biểu, ta có các qui tắc phủ định mệnh đề có lượng từ sau đây

$$\overline{\forall x : P(x)} \equiv \exists x : \overline{P}(x) \tag{2.3}$$

$$\overline{\exists x : P(x)} \equiv \forall x : \overline{P}(x) \tag{2.4}$$

Ví dụ 2.5. Tìm phủ định của mệnh đề "tồn tại một số thực x sao cho $x^2 < 0$".

Lời giải. Đặt $P(x) = "x^2 < 0"$. Mệnh đề đã cho được viết dưới dạng ký hiệu như sau "$\exists x : P(x)$". Áp dụng luật phủ định mệnh đề có lượng từ, ta có mệnh đề phủ định cần tìm có dạng :

"$\forall x : \overline{P}(x)$".

Vậy mệnh đề phủ định là: "Với mọi số thực x, $x^2 \geq 0$". ☺

Chú ý.

Từ các qui tắc trên ta có thể nói chung về qui tắc phủ định mệnh đề có lượng từ như sau: *Nếu trong một mệnh đề có lượng từ ta thay thế lượng từ $\forall$ bởi lượng từ $\exists$, lượng từ $\exists$ bởi lượng từ $\forall$, và biểu thức vị từ được thay thế bởi phủ định của nó thì ta sẽ được mệnh đề phủ định của mệnh đề có lượng từ ban đầu.* Qui tắc này cũng áp dụng được cho các mệnh đề với nhiều lượng từ.

Ví dụ 2.6. Cho $P(x, y, z)$ là một vị từ phụ thuộc vào biến bộ ba $(x, y, z) \in A \times B \times C$. Miền xác định là tích Đê-Cat của 3 tập hợp A, B, C. Trong trường hợp này ta nói vị từ P là một vị từ theo 3 biến x, y, z. Miền xác định tương ứng của 3 biến này là A, B, C. Hãy tìm phủ định của mệnh đề sau

$$\forall x \in A, \exists y \in B, \exists z \in C : P(x, y, z).$$

Lời giải. Theo qui tắc chung ta có

$$\overline{\forall x \in A, \exists y \in B, \exists z \in C : P(x, y, z)} \equiv$$

$$\exists x \in A, \forall y \in B, \forall z \in C : \overline{P}(x, y, z).$$

Thật ra nếu thực hiện từng bước theo các qui tắc (2.3) và (2.4) ta cũng đạt được mệnh đề phủ định như trên

$$\begin{aligned}
&\overline{\forall x \in A, \exists y \in B, \exists z \in C : P(x,y,z)} \equiv \\
&\exists x \in A, \overline{\exists y \in B, \exists z \in C : P(x,y,z)} \equiv \\
&\exists x \in A, \forall y \in B, \overline{\exists z \in C : P(x,y,z)} \equiv \\
&\exists x \in A, \forall y \in B, \forall z \in C : \overline{P}(x,y,z).
\end{aligned}$$

Ví dụ 2.7. Với một hàm số f xác định ở một lân cận của điểm $a \in \mathbb{R}$ (a là một số thực), ta có định nghĩa sự liên tục của f tại a như sau :

f liên tục tại a nếu và chỉ nếu cho một số dương ϵ tùy ý, ta có một số dương δ sao cho $|x - a| < \delta$ suy ra $|f(x) - f(a)| < \epsilon$.

Như vậy f liên tục tại a khi và chỉ khi mệnh đề sau đây đúng

"cho số dương ϵ tùy ý, ta có một số dương δ sao cho với mọi x ta có $|x - a| < \delta$ suy ra $|f(x) - f(a)| < \epsilon$".

Hãy tìm phủ định của mệnh đề trên.

Lời giải. Mệnh đề trên được viết là

$$\forall \epsilon > 0, \exists \delta > 0 : (\forall x : |x - a| < \delta \to |f(x) - f(a)| < \epsilon).$$

Theo qui tắc phủ định mệnh đề có lượng từ, phủ của mệnh đề trên là

$$\begin{aligned}
&\exists \epsilon > 0, \forall \delta > 0 : \overline{\forall x : |x - a| < \delta \to |f(x) - f(a)| < \epsilon} \\
&\equiv \exists \epsilon > 0, \forall \delta > 0 : (\exists x : \overline{|x - a| < \delta \to |f(x) - f(a)| < \epsilon}) \\
&\equiv \exists \epsilon > 0, \forall \delta > 0 : (\exists x : |x - a| < \delta \wedge \overline{|f(x) - f(a)| < \epsilon}) \\
&\equiv \exists \epsilon > 0, \forall \delta > 0 : (\exists x : |x - a| < \delta \wedge |f(x) - f(a)| \geq \epsilon) \\
&\equiv \exists \epsilon > 0, \forall \delta > 0 : \exists x : |x - a| < \delta \wedge |f(x) - f(a)| \geq \epsilon
\end{aligned}$$

Như vậy ta có thể phát biểu mệnh đề phủ định như sau "Tồn tại một số dương ϵ sao cho ứng với số dương δ tùy ý có một số thực x thoả điều kiện $|x - a| < \delta$ và $|f(x) - f(a)| \geq \epsilon$".

Như vậy ta có thể phát biểu mệnh đề phủ định như sau: "Tồn tại một số dương ϵ sao cho ứng với mỗi số dương δ tùy ý ta có một số thực x thỏa điều kiện $|x - a| < \delta$ và $|f(x) - f(a)| \geq \epsilon$".

2.1.5. Một số quy tắc dùng trong suy luận

1. Các lượng từ và các mệnh đề có lượng từ

Cho $P(x)$ là một vị từ một biến trên miền xác định nào đó (ví dụ như $D = \mathbb{N} = \{1, 2, ..., n\}$). Khi đó công thức $\forall x : P(x)$ và $\exists x : P(x)$ là các *mệnh đề có lượng từ* hoặc đúng hoặc sai trên miền xác định và theo định nghĩa vị từ và mệnh đề ta có

$$\forall x \in \mathbb{N} : P(x) \equiv P(1) \wedge P(2) \wedge \ldots \wedge P(n);$$
$$\exists x \in \mathbb{N} : P(x) \equiv P(1) \vee P(2) \vee \ldots \vee P(n);$$

Tuy nhiên nếu ta xét vị từ hai biến $P(x, y)$ trên miền xác định $D = D_1 \times D_2$ thì công thức $\forall x : P(x, y)$ và $\exists x : P(x, y)$ không phải là một mệnh đề có lượng từ nữa mà là vị từ theo biến y. x là biến bị ràng buộc bởi lượng từ, còn biến y là biến tự do. Nhưng biểu thức $\forall x : (\exists y : P(x, y))$ là một mệnh đề hoàn toàn xác định và không phụ thuộc vào x, y.

2. Thay đổi thứ tự lượng từ hóa của 2 biến

Cho một vị từ $P(x, y)$ theo 2 biến x, y. Nếu lượng từ hóa cả 2 biến x, y trong đó ta lượng từ hóa biến y trước và lượng từ hóa biến x sau thì sẽ được 4 mệnh sau đây

- $\forall x, \forall y : P(x, y)$;
- $\exists x, \forall y : P(x, y)$;
- $\forall x, \exists y : P(x, y)$;
- $\exists x, \exists y : P(x, y)$;

Tương tự ta cũng có 4 mệnh đề lượng từ hóa từ vị từ $P(x, y)$ trong đó ta lượng từ hóa biến x trước và lượng từ hóa biến y sau

- $\forall y, \forall x : P(x, y)$;
- $\exists y, \forall x : P(x, y)$;
- $\forall y, \exists x : P(x, y)$;
- $\exists y, \exists x : P(x, y)$;

Định lý dưới đây cho ta một số tính chất liên quan đến thứ tự của việc lượng từ hóa các biến trong các mệnh đề có lượng từ.

Định lý 2.1. *Giả sử $P(x, y)$ là một vị từ theo 2 biến x, y thì các mệnh đề sau là đúng*

- $(\forall x, \forall y : P(x, y)) \equiv (\forall y, \forall x : P(x, y))$;
- $(\exists x, \exists y : P(x, y)) \equiv (\exists y, \exists x : P(x, y))$;

3. Qui tắc đặc biệt hóa phổ dụng

Qui tắc 1. *Giả sử một mệnh đề có lượng từ trong đó biến x với miền xác định là A, được lượng từ hóa và bị ràng buộc bởi lượng từ phổ dụng $\forall$, và mệnh đề là đúng. Khi đó nếu thay thế x bởi $a \in A$ thì ta sẽ được một mệnh đề đúng.*

Ví dụ 2.8. Biết rằng phát biểu "mọi số nguyên tố lớn hơn 2 đều là số lẻ" là một mệnh đề đúng. Cho a là một số nguyên tố lớn hơn 2 (cố định nhưng tùy ý). Hãy chứng minh rằng a là một số lẻ.

Lời giải. Đặt $P(n) =$"n là số nguyên tố lớn hơn 2", và $Q(n) =$"n là số lẻ". Ta có $P(n)$ và $Q(n)$ là các vị từ theo biến số tự nhiên n, và ta có mệnh đề đúng sau đây

$$n : P(n) \to Q(n)$$

Theo qui tắc trên ta suy ra $P(a) \to Q(a) \equiv 1$. Theo giả thiết ta cũng có $P(a) = 1$. Suy ra $Q(a) = 1$ (qui tắc suy diễn Khẳng định). Vậy ta có mệnh đề "a là một số lẻ" là đúng.

Ví dụ 2.9. Trong các định lý Toán học ta thường thấy các khẳng định là các mệnh đề lượng từ hóa phổ dụng. Ví dụ như các trường hợp bằng nhau của 2 tam giác bất kỳ. Khi áp dụng ta sẽ đặc biệt hóa cho 2 tam giác cụ thể.

4. Qui tắc tổng quát hóa phổ dụng

Qui tắc 2. *Nếu ta thay thế biến x trong vị từ $P(x)$ bởi một phần tử a cố định nhưng tùy ý thuộc miền xác định của biến x mà mệnh đề nhận được có chân trị là đúng, tức là $P(a) = 1$, thì mệnh đề lượng từ hóa*

$$\forall x : P(x)$$

là một mệnh đề đúng.

Nhận xét. Nếu xem vị từ $P(x)$ như là một hàm lấy giá trị Bool trên miền xác định A của biến x thì ta có mệnh đề lượng từ hóa

$$\forall x : P(x)$$

là một mệnh đề đúng khi và chỉ khi P là hàm hằng 1.

Từ các qui tắc trên ta có thể chứng minh được một số tính chất suy diễn được phát biểu trong các mệnh đề sau đây.

Mệnh đề 2.1. *Cho $P(x)$ và $Q(x)$ là các vị từ theo biến x lấy giá trị trong tập hợp A (miền xác định của biến x là tập hợp A), và a là một phần tử cố định tùy ý thuộc A. Khi ấy ta có qui tắc suy diễn sau đây*

$$\begin{array}{l} \forall x : P(x) \to Q(x) \\ P(a) \\ \hline \therefore Q(a) \end{array}$$

Mệnh đề 2.2. *Cho $P(x), Q(x)$ và $R(x)$ là các vị từ theo biến x lấy giá trị trong tập hợp A (miền xác định của biến x là tập hợp A). Ta có qui tắc suy diễn sau đây*

$$\begin{array}{l} \forall x : P(x) \to Q(x) \\ \forall x : Q(x) \to R(x) \\ \hline \therefore (\forall x : P(x) \to R(x)) \end{array}$$

Mệnh đề 2.3. *Cho a là phần tử bất kỳ trong miền xác định của vị từ $P(x)$, ta có suy diễn tương đương sau*

$$\frac{\forall x : P(x)}{\therefore P(a)} \equiv \frac{\forall x : P(x)}{\therefore \exists x : P(x)}$$

Mệnh đề 2.4. *$P(x)$ là vị từ xác định trên $X = X_1 \cup X_2 \cup \ldots \cup X_m$. Khi đó*

$$\begin{array}{l} \forall x \in X_1 : P(x) \\ \forall x \in X_2 : P(x) \\ \ldots \\ \forall x \in X_m : P(x) \\ \hline \therefore \forall x \in X : P(x) \end{array}$$

2.2 Một số ví dụ sử dụng vị từ

Ví dụ 2.10. Mọi sinh viên công nghệ thông tin đều học môn toán rời rạc. An là sinh viên CNTT. Vậy An học môn toán rời rạc.

Chỉ ra suy luận trên là đúng.

Lời giải. Đặt $P(x)$ ="x là sinh viên CNTT"; $Q(x)$ ="x là hoạc môn Toán rời rạc" là hai vị từ một biến trên miền các sinh viên.

Đoạn văn trên tương đương với biểu thức trong loogic vị từ cấp 1. $(\forall x : P(x)) \to Q(x) \wedge P(x) \to Q(a)$ trên miền xác định sinh viên, ở đây a là sinh viên An trong tập hợp các sinh viên.

Mô hình suy diễn của công thức trên là

$$\begin{array}{l} \forall x : P(x) \to Q(x) \\ P(a) \\ \hline \therefore Q(a) \end{array} \tag{2.5}$$

Theo Bổ đề 2.1 thì mô hình (2.5) là đúng, nghĩa là suy luận trên đúng.

$$P(a) \to Q(a)$$

Chú ý. Sự tương đương suy diễn trên với $\dfrac{P(a)}{\therefore Q(a)}$ trong lôgic mệnh đề là và mô hình suy diễn này đúng theo quy tắc suy diễn khẳng định.

Ví dụ 2.11. Áp dụng mô hình suy diễn trong Bổ đề 2.2 để chỉ ra cách giải phương trình $4x - 5 = 15$ sau đây là đúng:

" Giả sử $4x - 5 = 15$, khi đó $4x = 20$. Giả sử $4x = 20$, khi đó $x = 5$. Như vậy $4x - 5 = 15$ thì $x = 5$".

Lời giải. Đặt $P(x) = "4x - 5 = 15"$; $Q(x) = "4x - 20 = 0"$; $R(x) = "x = 5"$;

Lý luận trên có thể diễn đạt theo mô hình suy diễn sau

$$\begin{array}{l} \forall x : P(x) \to Q(x) \\ \forall x : Q(x) \to R(x) \\ \hline \therefore (\forall x : P(x) \to R(x)) \end{array}$$

trên miền các số thực. Theo Bổ đề 2.2 ta có cách giải phương trình $4x - 5 = 15$ là đúng.

Ví dụ 2.12. Mô hình suy diễn dưới đây trên miền D có đúng không?

$$\begin{array}{l} \forall x : (P(x) \to (Q(x) \wedge R(x))) \\ \forall x : (P(x) \wedge F(x)) \\ \hline \therefore \forall x : (R(x) \wedge F(x)) \end{array} \tag{2.6}$$

Lời giải. Lấy a là phần tử cố định bất kỳ của D, thay $x = a$ vào (2.6) ta được mô hình suy diễn trong loogic mệnh đề là

$$\begin{array}{l} P(a) \to (Q(a) \wedge R(a)) \\ P(a) \wedge F(a) \\ \hline \therefore R(a) \wedge F(a) \end{array} \equiv \begin{array}{l} \begin{cases} P(a) \to (Q(a) \wedge R(a)) \\ P(a) \end{cases} \\ F(a) \\ \hline \therefore R(a) \wedge F(a) \end{array} \quad \text{(Kđ)}$$

$$\equiv \begin{array}{l} Q(a) \wedge R(a) \\ F(a) \\ \hline \therefore R(a) \wedge F(a) \end{array} \equiv \begin{array}{l} Q(a) \wedge R(a) \wedge F(a) \\ \hline \therefore R(a) \wedge F(a) \end{array} \equiv \mathbf{1}$$

Vậy (2.6) là đúng.

Ví dụ 2.13. Chỉ ra mô hình suy diễn dưới đây là đúng trên miền D.

$$\begin{array}{l} \forall x : (P(x) \vee Q(x)) \\ \exists x : \overline{P}(x) \\ \forall x : (\overline{Q}(x) \vee R(x)) \\ \forall x : (F(x) \to \overline{R}(x)) \\ \hline \therefore \exists x : \overline{F}(x) \end{array} \tag{2.7}$$

Lời giải. Áp dụng quy tắc tổng quát hóa phổ dụng. Lấy $a \in D$ bất kỳ sao cho $\overline{P}(x)$ đúng, thay $x = a$ vào (2.7) ta được mô hình suy diễn sau

$$\begin{array}{l} \begin{cases} P(a) \vee Q(a)) \\ \overline{P}(a) \end{cases} \\ \overline{Q}(a) \vee R(a) \\ F(a) \to \overline{R}(a) \\ \hline \therefore \overline{F}(a) \end{array} \equiv \begin{array}{l} \begin{cases} \overline{Q}(a) \\ \overline{Q}(a) \vee R(a) \end{cases} \\ F(a) \to \overline{R}(a) \\ \hline \therefore \overline{F}(a) \end{array} \equiv \begin{array}{l} \begin{cases} R(a) \\ F(a) \to \overline{R}(a) \end{cases} \\ \hline \therefore \overline{F}(a) \end{array}$$

$$\equiv \frac{\overline{F}(a)}{\therefore \overline{F}(a)} \equiv \mathbf{1}.$$

Vậy (2.7) là đúng.

Ví dụ 2.14. Chỉ ra mô hình sau đây là đúng trong miền xác định D.

$$\begin{array}{l} \forall x : P(x) \to Q(x) \\ \forall x : R(x) \to F(x) \\ \forall x : ((F(x) \wedge Q(x)) \to (P(x) \wedge H(x))) \\ \forall x : H(x) \to \overline{P}(x) \\ \hline \therefore \forall x : (\overline{R}(x) \vee \overline{P}(x)) \end{array} \tag{2.8}$$

Lời giải. Với mọi $a \in D$, thay $x = a$ trong (2.8) ta được mô hình suy diễn có dạng

$$\begin{array}{l} P(a) \to Q(a) \\ R(a) \to F(a) \\ (F(a) \wedge Q(a)) \to (P(a) \wedge H(a)) \\ H(a) \to \overline{P}(a) \\ \hline \therefore \overline{R}(a) \vee \overline{P}(a) \end{array} \equiv$$

$$\begin{array}{l} \begin{cases} P(a) \to Q(a) \\ P(a) \end{cases} \\ \begin{cases} R(a) \to F(a) \\ R(a) \end{cases} \\ (F(a) \wedge Q(a)) \to (P(a) \wedge H(a)) \\ H(a) \to \overline{P}(a) \\ \hline \therefore \mathbf{0} \end{array} \equiv$$

$$\begin{array}{l} \begin{cases} F(a) \wedge Q(a) \\ (F(a) \wedge Q(a)) \to (P(a) \wedge H(a)) \end{cases} \\ H(a) \to \overline{P}(a) \\ \hline \therefore \mathbf{0} \end{array} \equiv$$

$$\begin{array}{l} P(a) \\ \begin{cases} H(a) \\ H(a) \to \overline{P}(a) \end{cases} \\ \hline \therefore \mathbf{0} \end{array} \equiv \begin{array}{l} P(a) \\ \overline{P}(a) \\ \hline \therefore \mathbf{0} \end{array} \equiv \frac{\mathbf{0}}{\therefore \mathbf{0}} \equiv \mathbf{1}.$$

2.3 Các phương pháp chứng minh cơ bản

2.3.1. Khái niệm về chứng minh

Một hệ thống toán học bao gồm các tiên đề, các định nghĩa, và những khái niệm không được định nghĩa. Các tiên đề được giả định là đúng. Các định nghĩa được sử dụng để xây dựng hay đưa ra những khái niệm mới trên cơ sở những khái niệm đã có. Một số thuật ngữ, khái niệm sẽ không được định nghĩa rõ ràng nhưng được ngầm định nghĩa bởi các tiên đề. Trong một hệ toán học chúng ta có thể suy ra được các định lý. Một định lý là một mệnh đề được chứng minh là đúng. Một số loại định lý được xem là các bổ đề, các hệ quả.

Một lập luận (hay lý luận) chỉ ra được tính đúng đắn của mệnh đề phát biểu trong định lý được gọi là chứng minh.

Logic là cơ sở để thực hiện việc chứng minh, đặc biệt là các luật logic và các luật suy diễn. Trong phần này chúng ta sẽ đề cập đến việc xây dựng một chứng minh toán học. Các cấu trúc chứng minh thường được sử dụng là chứng minh trực tiếp, chứng minh bằng cách phân chia trường hợp (phân chứng), phản chứng, phản ví dụ, và chứng minh qui nạp.

2.3.2. Chứng minh trực tiếp

Chứng minh trực tiếp là phương pháp chứng minh suy diễn trực tiếp dẫn từ giả thiết đến kết luận thông qua việc áp dụng các luật suy diễn (hay qui tắc suy diễn), các định lý, các nguyên lý và các kết quả đã biết.

Đây là một kiểu tư duy giải bài toán rất tự nhiên và người ta thường xuyên sử dụng. Trong khi suy nghĩ để tìm ra cách chứng minh theo phương pháp này người ta thường phải tự trả lời các câu hỏi sau đây

- Ta sẽ dùng luật suy diễn nào?
- Các định lý nào, các kết quả nào có thể sử dụng được để ta suy ra được một điều gì đó từ những sự kiện, những yếu tố hiện

đang có?

- Việc áp dụng định lý có khả năng sẽ dẫn đến kết luận hay kết quả mong muốn hay không?
- Trong trường hợp ở một bước suy diễn nào đó có nhiều định lý hay nhiều luật nào đó có thể áp dụng được và cũng có khả năng sẽ dẫn đến kết luận hay kết quả mong muốn thì ta sẽ chọn cái nào?
- Đến một giai đoạn nào đó, khi gặp phải sự bế tắc thì ta sẽ phải tự hỏi rằng phải chăng bài toán không có lời giải, hay vì kiến thức của ta chưa đủ, hay ta phải sử dụng một phương pháp chứng minh nào khác?

Quả thật là không thể trả lời được các câu hỏi một cách đầy đủ và chính xác. Nó phụ thuộc chủ yếu vào kiến thức, kinh nghiệm của người giải bài toán và cả sự nhạy bén, tính năng động sáng tạo của họ. Tuy nhiên Những câu hỏi trên cho ta một sự định hướng chung của quá trình suy nghĩ. Ngoài ra, cũng cần nói thêm rằng chúng là cơ sở cho việc phát triển các hệ chương trình trợ giúp giải toán một cách "thông minh" trên máy tính được thiết kế theo phương pháp chứng minh này.

Dưới đây, chúng ta sẽ xem xét một 2 ví dụ về phương pháp chứng minh trực tiếp.

Ví dụ 2.15. Giả sử p, r, s, t, u là các mệnh đề sao cho ta có các mệnh đề sau đây là đúng
(1) $p \to r$
(2) $r \to s$
(3) $t \vee \overline{s}$
(4) $\overline{t} \vee u$
(5) $\overline{u}$.
Hãy chứng minh mệnh đề p là sai (tức là chứng minh mệnh đề $\overline{p}$ là đúng).

Lời giải. Áp dụng luật suy diễn tam đoạn luận, từ (1) và (2) ta suy ra

(6) $p \to s$.
Áp dụng luật logic về phép toán kéo theo ta có thể viết lại (3) dưới dạng
(7) $s \to t$.
Áp dụng luật suy diễn tam đoạn luận, từ (6) và (7) ta suy ra
(8) $p \to t$.
Áp dụng luật logic về phép toán kéo theo ta có thể viết lại (4) dưới dạng
(9) $t \to u$.
Áp dụng luật suy diễn tam đoạn luận, từ (8) và (9) ta suy ra
(10) $p \to u$.
Áp dụng luật suy diễn Modus Tollens, từ (10) và (5) ta suy ra
(11) $\overline{p}$.
Vậy mệnh đề $\overline{p}$ là đúng.

Ví dụ 2.16. Cho $P(x), Q(x)$ và $R(x)$ là các vị từ theo biến $x(x \in A)$, và a là một phần tử cố định nhưng tùy ý của tập hợp A. Giả sử ta có các mệnh đề sau đây là đúng
(1) $\forall x \in A : P(x) \to Q(x)$
(2) $\forall x \in A : Q(x) \to R(x)$
(3) $P(a)$
Chứng minh rằng mệnh đề $R(a)$ là đúng.

Lời giải. Áp dụng kết quả trong Bổ đề 2.2, từ (1) và (2) ta suy ra
(4) $\forall x \in A : P(x) \to R(x)$.
Áp dụng kết quả trong Bổ đề 2.1, từ (3) và (4) ta suy ra
(5) $R(a)$.
Vậy mệnh đề $R(a)$ là đúng.

2.3.3. Chứng minh phản chứng

Phương pháp chứng minh trực tiếp không phải bao giờ cũng sử dụng được trong việc chứng minh ngay cả đối với những bài toán khá đơn giản như bài toán sau đây

Bài toán. Chứng minh rằng không có hai số nguyên dương m và n sao cho $\dfrac{m}{n} = \sqrt{2}$.

Bằng cách suy nghĩ để tìm một cách chứng minh trực tiếp ta sẽ gặp phải bế tắc: Với $q = \dfrac{m}{n}$ là một số hữu tỉ cho trước (m và n là các số nguyên dương) ta không biết làm thế nào để suy ra một cách trực tiếp rằng $q^2 \neq 2$.

Để chứng minh bằng phản chứng một khẳng định hay một mệnh đề nào đó, ta tìm cách rút ra từ mệnh đề đó một điều rõ ràng là vô lý hay một sự mâu thuẫn. Về mặt kỹ thuật ta thường giả sử rằng mệnh đề cần chứng minh là sai rồi từ đó suy ra một điều mâu thuẫn với giả thiết hay các tiền đề của bài toán, từ đó đi đến kết luận rằng mệnh đề là đúng. Ngoài ra phép chứng minh phản chứng còn có thể được thực hiện như sau:

Ta giả sử mệnh đề cần chứng minh là sai, kết hợp với giả thiết đã cho để suy ra được một điều mâu thuẫn nào đó rồi từ đó kết luận rằng mệnh đề là đúng.

Cơ sở cho phương pháp chứng minh này là qui tắc chứng minh phản chứng

$$p \to q \leftrightarrow (p \wedge \overline{q}) \to 0.$$

Trở lại bài toán trên như một ví dụ, chúng ta có thể thực hiện việc chứng minh một cách dễ dàng nhờ phương pháp chứng minh phản chứng.

Ví dụ 2.17. Chứng minh rằng không có hai số nguyên dương m và n sao cho $\dfrac{m}{n} = \sqrt{2}$.

Lời giải. *Chứng minh phản chứng*. Giả sử ta có mệnh đề ngược lại của điều cần phải chứng minh, tức là giả sử rằng:

(1) Có hai số nguyên dương m và n sao cho $\dfrac{m}{n} = \sqrt{2}$.

Vì một phân số có thể viết dưới dạng tối giản, nên ta có thể giả thiết thêm rằng các số dương m và n trong mệnh đề (1) nguyên tố cùng nhau, tức là: m và n không có ước số chung lớn hơn 1.

Do $\dfrac{m}{m} = \sqrt{2} \Leftrightarrow m^2 = 2n^2$, từ (1) ta suy ra

(2) Có hai số nguyên dương m và n nguyên tố cùng nhau mà

$m^2 = 2n^2$.

Với m và n là hai số nguyên dương thỏa mãn điều kiện trong mệnh đề (2) ở trên thì ta dễ dàng lần lượt suy ra được các khẳng định sau đây
(3) m và n nguyên tố cùng nhau.
(4) $m^2 = 2n^2$.
(5) m là số chẵn.
(6) m là số chẵn, tức là $m = 2k$ với k là một số nguyên dương.
(7) $n^2 = 2k^2$.
(8) n^2 là số chẵn.
(9) n là số chẵn.
(10) 2 là một ước số chung của m và n, và $2 > 1$.
Sự mâu thuẫn do (3) và (10).
Từ lập luận trên ta đi đến kết luận:
Không có hai số nguyên dương m và n sao cho $\dfrac{m}{n} = \sqrt{2}$.

Chú ý. Ngoài cách chứng minh phản chứng ta còn có thể thực hiện phép chứng minh gián tiếp mà về thực chất phương pháp này là cùng loại với phương pháp chứng minh phản chứng. Trong cách chứng minh gián tiếp người ta thiệt lập sự đúng đắn của một mệnh đề bằng cách chứng minh rằng mệnh đề ngược lại (tức là mệnh đề phủ định của mệnh đề đó) là sai.

2.3.4. Chứng minh bằng cách chia trường hợp

Trong phương pháp chứng minh bằng cách chia trường hợp, để chứng minh một sự khẳng định nào đó ta xem xét tất cả các trường hợp có thể xảy ra đối với các sự kiện hay các yếu tố liên quan trong giả thiết và chứng minh rằng trong mỗi trường hợp ta đều có mệnh đề cần chứng minh là đúng, và từ đó đi đến kết luận rằng từ giả thiết ta có mệnh đề cần chứng minh là đúng.

Cơ sở cho phương pháp chứng minh này là qui tắc chứng minh theo trường hợp

$$(p_1 \to q) \land (p_2 \to q) \land \ldots \land (p_n \to q) \Rightarrow (p_1 \lor p_2 \lor \ldots \lor p_n) \to q$$

Để chứng minh khẳng định q (là đúng), chúng ta phân tích giả thiết để có được một sự khẳng định đúng dưới dạng:

$$p_1 \vee p_2 \vee \ldots \vee p_n$$

rồi tìm cách chứng minh rằng từ mệnh đề p_k suy ra được mệnh đề q, ứng với mỗi k từ 1 tới n.

Chúng ta hãy trở lại một bài toán về số nguyên đã được trình bày trong phần trước về các ví dụ áp dụng của logic trong việc lập luận và chứng minh.

Ví dụ 2.18. Cho một số nguyên n, hãy chứng minh rằng $n^3 + 2n$ chia hết cho 3.

Lời giải. Tóm tắt chứng minh: Với một số nguyên n, gọi r là dư số trong phép chia n cho 3, ta có 3 trường hợp
Trường hợp 1. $r = 0$
Trường hợp 1. $r = 1$
Trường hợp 1. $r = 2$
Nói cách khác ta có $(r = 0) \vee (r = 1) \vee (r = 2)$.
Trong mỗi trường hợp ta đều suy ra được $n^3 + 2n$ chia hết cho 3. Ở đây không trình bày lại chi tiết việc tính toán suy diễn trong mỗi trường hợp.

Từ đó đi đến kết luận $n^3 + 2n$ chia hết cho 3.

2.3.5. Phản ví dụ

Nói một cách tổng quát, phản ví dụ là việc chỉ ra một tình huống hay trường hợp sai của một khẳng định phổ quát để chứng tỏ rằng khẳng định phổ quát đó là sai.

Chẳng hạn như để chứng minh mệnh đề

$$x \in A : P(x)$$

là sai ta chỉ cần đưa ra một phần tử a cụ thể thuộc tập hợp A mà $P(a)$ là sai. Thật ra làm như vậy tức là ta đã chứng minh mệnh đề

$$\exists x \in A : \overline{P}(x)$$

(có cùng chân trị với mệnh đề $\overline{\forall x \in A : P(x)}$ là đúng).

Chúng ta có thể nêu lên một bài toán khác mà đối với nó ta phải dùng phản ví dụ. Đó là bài toán chứng minh một phép suy diễn từ $p_1, p_2, \ldots, p_n$ suy ra q là sai. Để chứng minh phép suy diễn là sai ta phải chứng minh rằng

$$p_1 \wedge p_2 \wedge \ldots \wedge p_n \to q$$

không phải là hằng đúng. Để làm điều này chúng ta chỉ cần tìm và chỉ ra một trường hợp cụ thể của các biến mệnh đề mà ứng với chúng ta có các tiền đề $p_1, p_2, \ldots, p_n$ đều đúng nhưng kết luận q là sai.

Ví dụ 2.19. Hãy kiểm tra suy luận sau đây

$$\begin{array}{l} p \to r \\ p \\ \bar{r} \to q \\ \hline \therefore q \end{array}$$

Sử dụng các phương pháp để kiểm tra một phép suy luận ta có thể thấy được suy luận trên là không đúng. Để tìm một phản ví dụ ta chỉ cần chỉ ra một trường hợp về chân trị của các biến mệnh đề sao cho các tiền đề trong phép suy luận là đúng còn kết luận là sai. Về mặt kỹ thuật ta sẽ tìm p, q, và r thỏa mãn các đẳng thức sau đây
$p \to r = 1$
$p = 1$
$\bar{r} \to q = 1$
$q = 0.$

Dễ dàng tìm thấy một trường hợp phản ví dụ là $p = 1, q = 0, r = 1$.

Vậy suy luận đã cho là không đúng.

2.4 Phương pháp quy nạp

2.4.1. Tập hợp số tự nhiên

Tập hợp gồm tất cả các số tự nhiên (hay các số đếm) được ký hiệu là $\mathbb{N}$:

$$\mathbb{N} = \{0, 1, 2, \ldots\}$$

Các phép toán.

Trên tập hợp số tự nhiên $\mathbb{N}$ ta có phép toán cộng (ký hiệu là +), và phép toán nhân (ký hiệu là .). Phép toán cộng 2 số tự nhiên a và b cho ta tổng số cũng là một số tự nhiên được viết là $a + b$; còn phép toán nhân 2 số tự nhiên a và b cho ta tích số cũng là một số tự nhiên được viết là $a.b$.

Phép toán cộng và nhân các số tự nhiên có các tính chất sau đây:

1. Phép cộng (+) và phép nhân (.) có tính giao hoán và kết hợp, nghĩa là với mọi số tự nhiên a, b, c ta có:
   - $a + b = b + a$
   - $(a + b) + c = a + (b + c)$
   - $a.b = b.a$
   - $(a.b).c = a.(b.c)$
2. Phép nhân (.) phân phối đối với phép cộng (+), nghĩa là với mọi số tự nhiên a, b, c ta có:
   - $a.(b + c) = a.b + a.c$

     Ở đây, phép toán nhân có độ ưu tiên cao hơn phép toán cộng.
3. Phép cộng và phép nhân lần lượt nhận 0 và 1 là phần tử trung hòa, nghĩa là ta có:
   - $\forall a \in \mathbb{N} : a + 0 = 0 + a = a$
   - $\forall a \in \mathbb{N} : a.1 = 1.a = a$

Thứ tự trên $\mathbb{N}$**.**

Ngoài các phép toán, trên N còn có *một quan hệ thứ tự* $\leq$ được định nghĩa như sau:

$$a \leq b \Leftrightarrow \exists c \in \mathbb{N} : a + c = b$$

Đây là một quan hệ thứ tự toàn phần trên $\mathbb{N}$, và theo quan hệ thứ tự này ta có:

$$0 < 1 < 2 < ... < n < n + 1 < ...$$

0 là phần tử nhỏ nhất của tập $\mathbb{N}$, nhưng tập $\mathbb{N}$ không có phần tử lớn nhất.

Thứ tự $\leq$ trên tập hợp $\mathbb{N}$ có một tính chất rất quan trọng được phát biểu trong mệnh đề sau đây.

Mệnh đề 2.5. *Cho A là một tập hợp các số tự nhiên khác rỗng. Khi đó A có phần tử nhỏ nhất, nghĩa là tồn tại* $a \in A$ *sao cho* $\forall x \in A, a \leq x$.

Chú ý. Phần tử a trong mệnh đề trên là duy nhất và được ký hiệu là $\min(A)$. Tính chất được phát biểu trong mệnh đề là cơ sở cho tính đúng đắn của các qui tắc chứng minh qui nạp sẽ được trình bày trong mục sau.

2.4.2. Các nguyên lý quy nạp thường dùng

Cho n_0 là một số tự nhiên, và $P(n)$ là một vị từ theo biến tự nhiên $n \geq n_0$. Vấn đề được đặt ra là chứng minh tính đúng đắng của mệnh đề sau đây:

$$\forall n \geq n_0 : P(n).$$

Phương pháp chứng minh qui nạp thường được sử dụng để chứng minh khẳng định trên. Phép chứng minh qui nạp được thực hiện dựa vào các nguyên lý qui nạp. Chúng ta sẽ nêu lên hai dạng nguyên lý qui nạp thường được sử dụng. Trong các áp dụng nguyên lý qui nạp, n_0 thường là 0 hoặc 1. Hai dạng nguyên lý qui nạp, được gọi là dạng qui nạp yếu và dạng qui nạp mạnh sẽ được viết dưới dạng mô hình của các luật suy diễn.

Định lý 2.2 (Nguyên lý qui nạp dạng yếu).

$$\begin{array}{ll} \textit{(cơ sở)} & P(n_0) \\ \textit{(qui nạp)} & \forall k \geq n_0 : P(k) \rightarrow P(k+1) \\ \hline & \therefore \forall n \geq n_0 : P(n) \end{array}$$

Chứng minh.

Đặt A là tập hợp các số tự nhiên $n \geq n_0$ mà $P(n)$ sai. Ta chỉ cần chứng minh rằng $A = \emptyset$ (tập hợp rỗng) với giả thiết rằng ta có hai khẳng định trong phần cơ sở và phần qui nạp trong nguyên lý trên. Ta sẽ chứng minh điều này bằng phương pháp phản chứng.

Giả sử $A \neq \emptyset$. Theo tính chất của thứ tự trên tập số tự nhiên $\mathbb{N}$ (xem mệnh đề ở mục trên), A có phần tử nhỏ nhất. Gọi a là phần tử nhỏ nhất của tập hợp A. Vì $P(n_0)$ đúng nên $a \geq n_0 + 1$, hay $a - 1 \geq n_0$. Do $a = \min(A)$, nên $a - 1 \notin A$ và do đó $P(a-1)$ đúng. Vì $P(a-1)$ đúng nên ta cũng có $P(a)$ đúng theo khẳng định ở phần qui nạp, nghĩa là ta cũng có $a \notin A$. Điều này cho ta một sự mâu thuẫn (vì $a = \min(A)$). Vậy $A = \emptyset$. Ta có điều cần chứng minh.

Định lý 2.3 (Nguyên lý qui nạp dạng mạnh).

$$\begin{array}{ll} \textit{(cơ sở)} & P(n_0) \\ \textit{(qui nạp)} & \forall k \geq n_0 : P(n_0) \wedge P(n_0+1) \wedge \ldots \wedge P(k) \rightarrow P(k+1) \\ \hline & \therefore \forall n \geq n_0 : P(n) \end{array}$$

Theo các nguyên lý trên, chứng minh qui nạp bao gồm 2 bước :

Bước cơ sở. Ở bước cơ sở, ta phải kiểm chứng để khẳng định $P(n_0)$ là đúng.

Bước qui nạp. Ở bước qui nạp, ứng với một số tự nhiên k tùy ý, ta phải chứng minh một mệnh đề kéo theo. Giả thiết trong mệnh đề kéo theo ở bước 2 được gọi là ***giả thiết qui nạp***. Giả thiết qui nạp ở dạng qui nạp yếu là $P(k)$, và ở dạng mạnh là $P(n_0) \wedge P(n_0+1) \wedge ... \wedge P(k)$.

Nguyên lý qui nạp có rất nhiều biến thể trong việc vận dụng. Chẳng hạn, từ hai nguyên lý trên ta có thể rút ra một nguyên lý qui nạp có dạng sau đây:

$$\begin{array}{ll} \text{(cơ sở)} & P(0) \wedge P(1) \\ \text{(qui nạp)} & \forall k \geq 1 : P(k-1) \wedge P(k) \wedge P(k+1) \\ \hline & \therefore \forall n \geq 0 : P(n) \end{array}$$

Trong chứng minh mệnh đề sau đây, ta sử dụng dạng qui nạp biến thể này.

Ví dụ 2.20. Cho dãy số

$$x_0, x_1, ..., x_n, ...$$

được định nghĩa bởi
$x_0 = 0$; $x_1 = 1$; và $x_n = 3x_{n-1} - 2x_{n-2}$ với mọi $n \geq 2$.

Khi đó ta có: $x_n = 2^n - 1$, với mọi $n \geq 0$.

Lời giải.

Đặt $P(n) = "x_n = 2^n - 1"$.

Dễ thấy rằng $P(0)$ và $P(1)$ là đúng. Bây giờ, ta chỉ cần thực hiện bước qui nạp để hoàn thành phép chứng minh qui nạp.

Giả sử $P(k-1)$ và $P(k)$ đúng với một số tự nhiên (tùy ý) $k \geq 1$.

Thế thì $x_{k-1} = 2^{k-1} - 1$ và $x_k = 2^k - 1$.

Do đó

$$\begin{aligned} x_{k+1} &= 3x_k - 2x_{k-1} \\ &= 3(2^k - 1) - 2(2^{k-1} - 1) \\ &= 3 * 2^k - 3 - 2^k - 2 \\ &= 2 * 2^k - 1 = 2^{k+1} - 1. \end{aligned}$$

Suy ra $P(k+1)$ đúng. ☺

Vậy theo nguyên lý qui nạp (dạng biến thể được phá biểu ở trên) ta kết luận: $P(n)$ đúng với mọi $n \geq 0$.

2.4.3. Các ví dụ quy nạp

Ví dụ 2.21. Chứng minh rằng với mọi $n \geq 1 (n \in N)$ ta có

$$1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

Chứng minh. Với mỗi số tự nhiên $n \geq 1 (n \in N)$, đặt

$$P(n) = "1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}"$$

Ta sẽ chứng minh mệnh đề $\forall n \geq 1 : P(n)$ bằng phương pháp qui nạp (dạng yếu), nghĩa là thực hiện 2 bước chứng minh trong phép chứng minh qui nạp.

- *Bước cơ sở*: Khi $n = 1$ thì $P(1)$ là mệnh đề $"1 = \frac{1.(1+1).(2.1+1)}{6}"$. Vế phải của đẳng thức trong mệnh đề tính ra bằng 1, nên ta có $P(1)$ đúng.
- *Bước qui nạp*: Cho n là một số tự nhiên túy ý lớn hơn 0, nghĩa là $n \geq 1$, và giả sử rằng $P(n)$ đúng, tức là ta có (GTQN)

  $$1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

  Bây giờ ta sẽ chứng minh $P(n+1)$ cũng đúng, tức là chứng minh rằng

  $$1^2 + 2^2 + \cdots + n^2 + (n+1)^2 = \frac{(n+1)(n+2)(2n+3)}{6}$$

  Thật vậy, từ (GTQN) ta suy ra

  $$\begin{aligned} 1^2 + 2^2 + \cdots + n^2 + (n+1)^2 &= \frac{n(n+1)(2n+1)}{6} + (n+1)^2 \\ &= \frac{n+1}{6}(n(2n+1) + 6(n+1)) \\ &= \frac{n+1}{6}(2n^2 + 7n + 6)) \\ &= \frac{n+1}{6}(n+2)(2n+3) \\ &= \frac{(n+1)(n+2)(2n+3)}{6}. \end{aligned}$$

  Tức là ta đã suy ra được $P(n+1)$ cũng đúng.

  Hai phần (phần cơ sở và phần qui nạp) trong phép chứng minh qui nạp đã được chứng minh.

  Vậy theo nguyên lý qui nạp ta kết luận rằng với mọi $n \geq 1 (n \in$

$\mathbb{N}$) ta có

$$1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

Ví dụ 2.22. Chứng minh định lý sau đây:
Cho a và b là 2 số nguyên tự nhiên, với $b \geq 0$. Khi đó, có duy nhất 2 số tự nhiên q và r thỏa 2 điều kiện sau đây
(1) $a = q.b + r$
(2) $0 \leq r < b$

Chứng minh.

Tính duy nhất của q và r trong định lý trên có thể kiểm chứng dễ dàng. Ở đây ta sẽ chứng minh sự tồn tại của q và r. Cho b là một số tự nhiên khác 0 tùy ý nhưng cố dịnh. Đặt

$P(a) =$"Tồn tại các số tự nhiên q và r thỏa mãn 2 điều kiện (1) và (2)"

Ta sẽ chứng minh mệnh đề sau đây là đúng

$$\forall a \in \mathbb{N} : P(a)$$

bằng cách sử dụng nguyên lý qui nạp dạng mạnh.

Bước cơ sở: Xét trường hợp $a = 0$, ta thấy với $q = 0$ và $r = 0$ thì các điều kiện (1) và (2) được thỏa mãn. Vậy ta có $P(0)$ đúng.

Bước qui nạp: Cho a là một số tự nhiên t y ý và giả sử rằng các mệnh đề $P(0), P(1), ..., P(a)$ đều đúng (GTQN). Ta sẽ chứng minh $P(a+1)$ cũng đúng bằng cách xét 2 trường hợp.

- Trường hợp 1: $a + 1 < b$.

  Ta chọn $q = 0$ và $r = a + 1$ thì ta có q và r thỏa các điều kiện

  (1) $a + 1 = q.b + r$

  (2) $0 \leq r < b$

  Vậy trong trường hợp 1 này thì $P(a+1)$ đúng.

- Trường hợp 2: $a + 1 \geq b$.

  Đặt $a' = a + 1 - b$, ta có $0 \leq a' \leq a$. Từ (GTQN) ta suy ra $P(a')$ đúng, tức là có các số tự nhiên q' và r' sao cho $a' = q'.b + r'$, và $0 \leq r' < b$.

Suy ra các số tự nhiên $q = q' + 1$ và $r = r'$ sẽ thỏa mãn 2 điều kiện

(1) $a + 1 = q.b + r$

(2) $0 \leq r < b$

Vậy ta cũng có $P(a + 1)$ đúng.

Tóm lại, ta đã chứng minh phần cơ sở và phần qui nạp trong nguyên lý qui nạp. Từ đó có thể kết luận rằng

$$\forall a \in \mathbb{N} : P(a).$$

Ghi chú: Định lý trên là cơ sở cho việc định nghĩa phép toán div (chia lấy thương số) và phép toán mod (chia lấy dư số). Định lý này còn được gọi là thuật chia Euclide. Xem xét việc chứng minh của định lý trên theo phương pháp qui nạp, chúng ta có thể rút ra một thuật toán (khái niệm thuật toán sẽ được trình bày trong chương 3). Dạng tổng quát của định lý, cho các số nguyên, được phát biểu trong bài tập 6.

Ví dụ 2.23. Chứng minh rằng đẳng thức sau đúng với mọi $n \geq 1$

$$1.1! + 2.2! + \cdots + n.n! = (n+1)! - 1.$$

Ví dụ 2.24. Chứng minh đẳng thức sau đúng với mọi $n > 1, a > -1$ và $a \neq 0$

$$(1 + a)^n > 1 + na.$$

Ví dụ 2.25. Chứng minh rằng tích của ba số tự nhiên liên tiếp chia hết cho 6.

Ví dụ 2.26. Tính tổng của n số tự nhiên lẻ đầu tiên $T(n) = 1 + 3 + 5 + \cdots + (2n - 1)$.

Ví dụ 2.27. Tính tổng

$$T(n) = \frac{1}{1.2} + \frac{1}{2.3} + \cdots + \frac{1}{n(n+1)}.$$

2.5 Bài tập

▷ **2.1.** Vị từ là gì? Cho ví dụ.

▷ **2.2.** Phát biểu qui tắc phủ định mệnh đề có lượng từ (hay mệnh đề lượng từ hóa) và cho ví dụ cụ thể.

▷ **2.3.** P(n) là vị từ "nếu $4|n$ thì $2|n$". Cho biết chân trị của các mệnh đề sau a) $P(12)$;
b) $P(10)$;
c) $\exists n : P(n)$;
d) $\forall n : P(n)$.

▷ **2.4.** Hãy cho biết chân trị của mỗi mệnh đề dưới đây và viết mệnh đề phủ định của mệnh đề đó.
a) $\exists x : x + 3 = 5$;
b) $\forall x : x + 3 = 5$;
c) $\exists x, \exists y : x + y = 3$;
d) $\exists x, \forall y : x + y = 3$;
e) $\forall x, \exists y : x + y = 3$;
f) $\forall x, \forall y : x + y = 3$.
Trong các mệnh đề trên các biến x và y là các biến thực.

▷ **2.5.** Hãy cho biết chân trị của mỗi mệnh đề dưới đây và viết mệnh đề phủ định của mệnh đề đó.
a) $\exists x, \exists y : (x^2 = y^2) \to (x = y)$;
d) $\exists x, \forall y : (x^2 = y^2) \to (x = y)$;
e) $\forall x, \exists y : (x^2 = y^2) \to (x = y)$;
f) $\forall x, \forall y : (x^2 = y^2) \to (x = y)$.

▷ **2.6.** Hãy sử dụng các ký hiệu toán học và logic để viết lại mệnh đề sau đây:

Với mọi số thực dương x, có một số tự nhiên n sao cho x bằng 2^n hoặc x nằm giữa 2^n và 2^{n+1}.

Cho biết mệnh đề này đúng hay sai, và viết ra mệnh đề phủ định của nó.

▷ **2.7.** Trong bài tập này ký hiệu n chỉ một biến nguyên. Cho các vị từ
$P(n) = "0 < n^2 \leq 4"$

$R(n) = "0 < n^3 \leq 8"$
$S(n) = "0 < n \leq 2"$
a) Ứng với mỗi vị từ trên hãy cho biết tập hợp các giá trị n làm cho vị từ có chân trị đúng ($= 1$).
b) Trong các vị từ trên, những vị từ nào tương đương với nhau.
c) Mệnh đề "$\forall n : R(n) \to P(n)$" là đúng hay sai?

▷ **2.8.** 1) Cho $P(x)$ là "$x + 1 > x$". Xác định giá trị chân lý của $\forall x : P(x)$ trên trường các số thực.

2) Cho $P(x)$ là "$x < 2$". Xác định giá trị chân lý của $\forall x : P(x)$ trên trường các số thực.

3) Cho $P(x)$ là "$x \leq 10$". Xác định giá trị chân lý của $\forall x : P(x)$ trên trường các số tự nhiên không vượt quá 4.

4) Cho $P(x)$ là "$x < 2$". Xác định giá trị chân lý của $\exists x : P(x)$ trên trường các số thực.

5) Cho $P(x)$ là "$x = x + 1$". Xác định giá trị chân lý của $\exists x : P(x)$ trên trường các số thực.

▷ **2.9.** 1) Cho $P(x, y)$ là "$x + y = y + x$" trên trường số thực. Xác định giá trị chân lý của $\forall x : (\forall y : P(x, y))$.

2) Cho $Q(x, y)$ là "$x + y = 0$" trên trường số thực. Xác định giá trị chân lý của $\exists x : (\forall y : Q(x, y))$ và $\forall x : (\exists y : Q(x, y))$.

3) Cho $Q(x, y, z)$ là "$x + y = z$" trên trường số thực. Xác định giá trị chân lý của
$\forall x : (\forall y : (\exists z Q(x, y, z)))$ và $\exists z : (\forall x : (\forall y : Q(x, y, z)))$.

▷ **2.10.** Biểu diễn các câu sau thành biểu thức lôgic

1) "Mọi người đều có chính xác một người bạn rất tốt".

2) "Nếu một người phụ nữ nào đó đã sinh đẻ, thì người đó là mẹ của một người nào đó".

▷ **2.11.** $F(x, y)$ là câu "x yêu y", với x và y trong trường mọi người trên thế giới. Hãy dùng các lượng từ để diễn đạt các câu sau:

a) "Mọi người đều yêu Jerry";

b) "Mọi người đều yêu một người nào đó";

c) "Có một người mà tất cả mọi người đều yêu";

d) "Không có ai yêu tất cả mọi người";

e) "Có một người không ai yêu".

▷ **2.12.** Cho $P(x)$ là câu "x học ở lớp hơn 5 giờ mỗi ngày trong tuần" trên miền xác định các sinh viên. Hãy diễn đạt các lượng từ sau thành câu thông thường.

a) $\exists x : P(x)$;

b) $\forall x : P(x)$;

c) $\exists x : \overline{P}(x)$;

d) $\forall x : \overline{P}(x)$;

▷ **2.13.** Cho $P(x, y)$ là câu "x đã học môn y" với trường của x là tập hợp tất cả sinh viên trong lớp, còn trường của y là tập hợp các môn tin học của khoa đã quy định. Hãy diễn đạt các mệnh đề lượng từ sau thành câu thông thường:

a) $\exists x, \exists y : P(x, y)$;

b) $\exists y, \forall x : P(x, y)$.

▷ **2.14.** Cho $P(x), Q(x)$ và $F(x)$ là các câu tương ứng sau "x là giáo sư", "x là kẻ ngu ngốc" và "x là kẻ vô tích sự". Bằng cách dùng lượng từ, các liên từ lôgic cùng với $P(x), Q(x), F(x)$ để diễn đạt các câu sau trên trường tất cả mọi người:

a) "Không có giáo sư nào là kẻ ngu ngốc";

b) "Mọi kẻ ngu ngốc đều là vô tích sự";

c) "Không có giáo sư nào là vô tích sự".

▷ **2.15.** Chứng minh rằng công thức $\overline{\exists x, \forall y : P(x)}$ có cùng giá trị chân lý với công thức $\forall x, \exists y : \overline{P(x)}$

▷ **2.16.** Chứng minh rằng $(\forall x : P(x)) \vee (\forall x : Q(x))$ và $(\forall x : (P(x) \vee Q(x)))$ là không tương đương lôgic, còn $(\forall x : P(x)) \wedge A$ và $(\forall x : (P(x) \wedge A)$ là tương đương lôgic, với A là mệnh đề không có lượng từ.

▷ **2.17.** Chứng tỏ $(\exists x : P(x)) \wedge (\exists x : Q(x))$ và $(\exists x : (P(x) \wedge Q(x)))$ là không tương đương lôgic.

▷ **2.18.** Xác định giá trị chân lý các biểu thức sau

a) $(\exists x! : P(x)) \to (\exists x : P(x))$;

b) $(\forall x : P(x)) \to (\exists x! : P(x))$;

c) $(\exists x! : \overline{P}(x)) \to \overline{\forall x : P(x)}$.

Ở đây $(\exists x! : P(x))$ là ký hiệu mệnh đề "Tồn tại duy nhất một x sao cho $P(x)$ là đúng" trên trường các số nguyên.

▷ **2.19.** Chứng minh các đẳng thức sau

a) $1.2 + 2.3 + \cdots + n(n+1) = \dfrac{n(n+1)(n+2)}{3}$ với mọi $n \geq 1$.

b) $1.2.3 + 2.3.4 + \cdots + n(n+1)(n+2) = \dfrac{n(n+1)(n+2)(n+3)}{4}$ với mọi $n \geq 1$.

▷ **2.20.** Chứng minh đẳng thức

$$\sum_{i=1}^{n} i^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}$$

với mọi $n \geq 1$

▷ **2.21.** Chứng minh công thức De Morgan tổng quát sau

$$\overline{\bigcap_{i=1}^{n} A_i} = \bigcup_{i=1}^{n} \overline{A}_i \text{ và } \overline{\bigcup_{i=1}^{n} A_i} = \bigcap_{i=1}^{n} \overline{A}_i$$

▷ **2.22.** Các số điều hòa được định nghĩa như sau

$$H_k = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k}$$

Dùng phương pháp quy nạp toán học chứng minh rằng $H_{2^n} \geq 1 + \frac{n}{2}$, n là số nguyên không âm.

▷ **2.23.** Dùng phương pháp quy nạp toán học chứng minh rằng Nếu A có n phần tử thì số các tập con của A là 2^n $(n \geq 0)$.

Chương 3

CÁC PHƯƠNG PHÁP ĐẾM

3.1 Tập hợp

3.1.1. Khái niệm tập hợp

Khái niệm *tập hợp* được dùng để chỉ một sưu tập hay một nhóm các đối tượng nào đó mà ta đang quan tâm xem xét, và sưu tập này phải được xác định tốt. Các đối tượng trong sưu tập hay trong nhóm này sẽ được gọi là các *phần tử* hay các thành viên của tập hợp. Tính xác định tốt (hay nói vắn tắt là tính xác định) của tập hợp được hiểu theo nghĩa là với một đối tượng nào đó mà ta đang quan tâm thì ta có thể xác định được đích xác rằng trường hợp nào là đúng trong hai trường hợp sau đây

- Trường hợp 1: đối tượng là một phần tử của tập hợp. Trong trường hợp này ta nói đối tượng *thuộc về* tập hợp.
- Trường hợp 2: đối tượng không phải là một phần tử của tập hợp. Trong trường hợp này ta nói đối tượng *không thuộc về* tập hợp.

Để thuận tiện cho việc đề cập đến tập hợp về sau, mỗi tập hợp thường được đặt cho một tên, chẳng hạn như $A, B, C\ldots$. Ta cũng dùng ký hiệu $\in$ để diễn đạt quan hệ "thuộc về" của một phần tử đối với một tập hợp. Khi x là một phần tử thuộc về tập hợp A, thì ta viết $x \in A$ và đọc là "x thuộc A", hay đọc là "A chứa phần tử x".

Ngược lại, nếu x không phải là một phần tử của tập hợp A thì ta viết $x \notin A$ và đọc là "x không thuộc A", hay đọc là "A không chứa phần tử x".

- **Tập hợp bằng nhau**: Hai tập hợp A và B sẽ được xem là *bằng nhau* khi chúng có cùng các phần tử, tức là mỗi phần tử thuộc

A đều là phần tử thuộc B và ngược lại. Khi ấy, ta viết là $A = B$.

- **Tập hợp rỗng:** Tập hợp không có phần tử nào được gọi là *tập hợp rỗng*, và được ký hiệu là $\emptyset$.
- **Cách xác định một tập hợp:** Để *xác định một tập hợp* ta có thể dùng các cách sau đây:
  - *Cách liệt kê:* Ta liệt kê tất cả các phần tử của tập hợp giữa ký hiệu ngoặc { và }.

    Ví dụ 3.1. $A = \{a, b, c\}; \quad B = \{0, 1\}$.

  - *Cách nêu đặc trưng của phần tử:*

    Theo cách này, để xác định một tập hợp A ta sẽ nêu lên "tính chất" dùng để xác định xem phần tử trong một không gian U có thuộc về tập hợp A hay không: phần tử x của U sẽ thuộc A khi x thỏa "tính chất", và x không thuộc A khi x không thỏa "tính chất". Từ "tính chất" thường được thể hiện dưới dạng một vị từ $p(x)$ theo biến $x \in U$. Khi ấy, tập hợp A sẽ được viết như sau:

    $$A = \{x \in U | p(x)\}$$

    hay vắn tắt (hiểu ngầm tập U) là

    $$A = \{x | p(x)\}$$

    Ví dụ 3.2. $A = \{n \in N | n$ là số nguyên tố $\}$
    $B = \{n \in N|$ có một số tự nhiên m sao cho $n = m^2\}$.

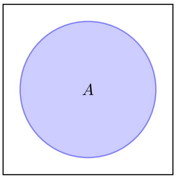
  - *Cách xác định tập hợp* dưới dạng ảnh của một tập hợp khác A' qua một phép tương ứng f mà ứng với mỗi $x \in A'$ ta có một phần tử tương ứng $f(x)$ duy nhất trong U. Khi ấy ta viết

    $$A = \{f(x) | x \in A'\}$$

    Ghi chú: Phép tương ứng f được nói trên đây chính là một ánh xạ.

    Ví dụ 3.3. $B = \{n^2 | n \in N\}; \quad C = \{(2n+1)^2 | n \in N\}$.

  - *Sơ đồ Venn*



Hình 3.1. Tập hợp A

3.1.2. Quan hệ "bao hàm trong" và tập hợp con

Định nghĩa 3.1. Cho A và B là hai tập hợp mà các phần tử của chúng đều thuộc một tập hợp lớn U (hay còn gọi là tập vũ trụ). Ta nói tập A *bao hàm trong* (hay chứa trong) tập B nếu mỗi phần tử của tập hợp A đều thuộc tập hợp B. Ta cũng nói rằng B bao hàm A (hay B chứa A), và viết là: $A \subset B$ (hay $B \supset A$).

Khi $A \subset B$ ta nói A là một *tập hợp con* của tập hợp B.

Ví dụ 3.4. $\{0, 1, 2\} \subset \{n \in N | n < 10\}$;
$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$,
trong đó $\mathbb{N}$ là tập hợp các số tự nhiên, $\mathbb{Z}$ là tập hợp các số nguyên, $\mathbb{Q}$ là tập hợp các số hữu tỉ, $\mathbb{R}$ là tập hợp các số thực, $\mathbb{C}$ là tập hợp các số phức.

Cho X là một tập hợp. Sưu tập tất cả các tập hợp con của X được ký hiệu là $P(X)$. Nói một cách khác, $P(X)$ là một tập hợp mà mỗi phần tử của nó là một tập hợp con của X.

Tính chất.

- $\emptyset \subset A$ và $A \subset A$, với mọi tập hợp A.
- $(A \subset B)$ và $(B \subset A) \Rightarrow (A = B)$.
- $(A \subset B)$ và $(B \subset C) \Rightarrow (A \subset C)$.
- $X \subset Y \Rightarrow P(X) \subset P(Y)$.
- Nếu tập hợp X có n phần tử ($n \in N$) thì tập hợp $P(X)$ có 2^n phần tử.

3.1.3. Các phép toán trên tập hợp

Trong mục này chúng ta sẽ nêu lên định nghĩa các phép toán tập hợp trên các tập hợp con của một tập hợp vũ trụ U cho trước, và phát biểu một số tính chất liên quan đến các phép toán.

Định nghĩa 3.2. *Giao* của 2 tập hợp A và B, ký hiệu bởi $A \cap B$, là tập hợp gồm tất cả các phần tử của U mà vừa thuộc tập A vừa thuộc tập B.

$$A \cap B = \{x|(x \in A) \wedge (x \in B)\}.$$

Ví dụ 3.5. a) Cho $A = \{1, 2, 3, 4\}; B = \{5, 6, 7, 8\}$. Ta có $A \cup B = \{1, 2, 3, 4, 5, 6, 7, 8\}$

b) Cho $C = \{a, b, c\}; D = \{c, d, e\}$. Ta có $C \cup D = \{a, b, c, d, e\}$

Định nghĩa 3.3. *Hợp* của 2 tập hợp A và B, ký hiệu bởi $A \cup B$, là tập hợp gồm tất cả các phần tử của U sao cho nó thuộc tập A hay thuộc tập B.

$$A \cup B = \{x|(x \in A) \vee (x \in B)\}.$$

Ví dụ 3.6. a) Cho $A = \{1, 2, 3, 4\}; B = \{5, 6, 7, 8\}$. Ta có $A \cap B = \emptyset$

b) Cho $C = \{a, b, c\}; D = \{c, d, e\}$. Ta có $C \cap D = \{e\}$

Định nghĩa 3.4. *Hiệu* của 2 tập hợp A và B, ký hiệu bởi $A \setminus B$ (hay $A - B$), là tập hợp gồm tất cả các phần tử của U sao cho nó thuộc tập A và không thuộc tập B.

$$A \setminus B = \{x|(x \in A) \wedge (x \notin B)\}.$$

Ví dụ 3.7. a) Cho $A = \{1, 2, 3, 4\}; B = \{5, 6, 7, 8\}$. Ta có $A \setminus B = \emptyset$

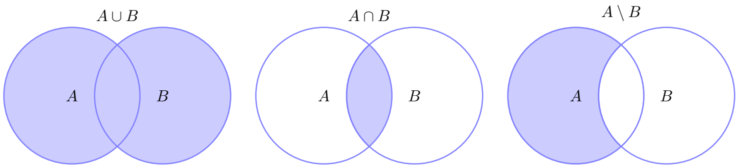
b) Cho $C = \{a, b, c\}; D = \{c, d, e\}$. Ta có $C \setminus D = \{e\}$

Định nghĩa 3.5. *Phần bù* của tập A (trong U), ký hiệu bởi $\overline{A}$, là tập hợp tất cả các phần tử của U mà không thuộc A. Nói cách khác,

$$\overline{A} = U \setminus A$$

Ví dụ 3.8. a) Cho $A = \{1, 2, 3, 4, 5\}; B = \{4, 5\}$. Ta có $\overline{A} = \{4, 5\}$

Sơ đồ Venn



Hình 3.2. Phép toán trên tập hợp

Các tính chất của phép toán tập hợp.

- Tính giao hoán

$$A \cap B = B \cap A$$
$$A \cup B = B \cup A$$

- Tính kết hợp

$$(A \cap B) \cap C = A \cap (B \cap C)$$
$$(A \cup B) \cup C = A \cup (B \cup C)$$

- Tính phân bố:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

- Luật De Morgan

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$
$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

- Phần tử trung hòa

$$A \cup \emptyset = A$$
$$A \cap U = A$$

- Phần bù

$$A \cup \overline{A} = U$$
$$A \cap \overline{A} = \emptyset$$

- Tính thống trị

$$A \cup U = U$$
$$A \cap \emptyset = \emptyset$$

3.1.4. Lực lượng của tập hợp

Định nghĩa 3.6. *Lực lượng* của một tập hợp A là số phần tử của tập hợp A, ký hiệu là $|A|$

Khi đó ta có ba công thức thường gặp khi phải tính số phần tử của một tập hợp.

1. $|A \cup B| = |A| + |B| - |A \cap B|$.
2. $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$.
3. $|A \cup B \cup C \cup D| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |A \cap D| - |B \cap C| - |B \cap D| - |C \cap D| + |A \cap B \cap C| + |A \cap B \cap D| + |A \cap C \cap D| + |B \cap C \cap D| - |A \cap B \cap C \cap D|$.

3.1.5. Tích Decartes của các tập hợp

- **Tích Descartes của 2 tập hợp:**

  Cho 2 tập hợp A và B. Tích Descartes của tập hợp A và tập hợp B, được ký hiệu bởi $A \times B$, là tập hợp gồm tất cả các cặp (a, b) sao cho $a \in A$ và $b \in B$.

  $$A \times B = \{(a, b) | a \in A \wedge b \in B\}$$

  Trong trường hợp $B = A$, ta ký hiệu $A \times B$ là A^2 .

- **Tích Descartes của nhiều tập hợp:**

  Cho n tập hợp $A_1, A_2, \ldots, A_n (n > 1)$. Tích Descartes của n tập hợp $A_1, A_2, \ldots, A_n$, được ký hiệu bởi $A_1 \times A_2 \times \ldots \times A_n$, là tập hợp gồm tất cả các bộ n phần tử $(a_1, a_2, \ldots, a_n)$ với $a_i \in A_i$ với mọi $i = 1, \ldots, n$.

  Trong trường hợp $A_1 = A_2 = ... = A_n = A$ thì tập hợp tích $A_1 \times A_2 \times \ldots \times A_n$ sẽ được viết là A.

- **Lượng lượng của tập tích Decartes.**

  1. $|A \times B| = |A|.|B|$
  2. $|A \times B \times C| = |A|.|B|.|C|$
  3. $|A_1 \times A_2 \times \ldots \times A_n| = |A_1|.|A_2|.\ldots.|A_n| = \prod_{i=1}^{n} |A_i|$.

3.1.6. Biểu diễn tập hợp trong máy tính

Có nhiều cách biểu diễn tập hợp trên máy tính. Dưới đây giới thiệu một cách biểu diễn tập hợp trong máy tính bằng cách lưu trữ các phần tử của nó dưới dạng sắp tùy ý các phần tử của nó dưới dạng sắp tùy ý các phần tử của tập vũ trụ.

Giả sử X là một tập vũ trụ và $A \subseteq X$ (với giả thiết dung lượng bộ nhớ của máy tính không bé hơn lực lượng của X)

Giả sử $|X| = n$. khi đó ta sắp các phần tử của $X = \{a_1, a_2, ..., a_n\}$. Ta có thể biểu diễn tập A trên máy tính bằng một xâu bit có chiều dài n, trong đó bit thứ i là 1 nếu $a_i \in A$, còn bit thứ i là 0 nếu $a_i \notin A$ $(1, 2, ..., n)$.

Ví dụ 3.9. Cho $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ (sắp xếp các phần tử của X theo thứ tự tăng dần).

a) Xác định xâu bit của tập $A = \{1, 3, 5, 7, 9\} \subset X$

b) Xác định xâu bit của tập $B = \{2, 4, 6, 8, 10\} \subset X$

b) Xác định xâu bit của các phần tử không vượt quá 5 trong X tức là tìm xâu bít của $C = \{1, 2, 3, 4, 5\} \subset X$

Lời giải. a) Xâu bit của tập $A = \{1, 3, 5, 7, 9\}$ là 1010101010.

b) Xâu bit của tập $B = \{2, 4, 6, 8, 10\}$ là 0101010101.

b) Xâu bít của $C = \{1, 2, 3, 4, 5\}$ là 1111100000.

Ví dụ 3.10. Cho $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ và $A = \{1, 3, 5, 7, 9\}$, $B = \{2, 4, 6, 8, 10\}$, $B = \{1, 2, 3, 4, 5\}$.

a) Tìm xâu bít của $A \cup B$.

b) Tìm xâu bít của $A \cap B$.

c) Tìm xâu bít của $\overline{A}$ và $\overline{B}$.

Lời giải. Xâu bit của tập $A = \{1, 3, 5, 7, 9\}$ là 1010101010. Xâu bit của tập $B = \{2, 4, 6, 8, 10\}$ là 0101010101.

a) Xâu bít của $A \cup B$ là $1010101010 \vee 0101010101 = 1111111111$.

b) Xâu bít của $A \cap B$ là $1010101010 \wedge 0101010101 = 0000000000$.

c) Vì $\overline{A} = \{1, 3, 5, 7, 9\}$. Xâu bít của $\overline{A}$ là 0101010101.

c) Vì $\overline{B} = \{2, 4, 6, 8, 10\}$. Xâu bít của $\overline{B}$ là 1010101010.

3.2 Các nguyên lý đếm

Từ lâu, người ta đã nghiên cứu việc liệt kê, đếm các phần tử hay các đối tượng có những tính chất nào đó để giải quyết một số vấn đề cần thiết được đặt ra. Chẳng hạn, phép đếm được sử dụng trong việc phân tích và đánh giá độ phức tạp của thuật toán. Kỹ thuật đếm còn được sử dụng trong việc tính toán xác suất của các sự kiện.

Trong mục này chúng ta sẽ trình bày các qui tắc cơ bản của phép đếm. Chúng sẽ giúp ích rất nhiều cho việc giải nhiều vấn đề liên quan đến việc liệt kê, sắp xếp và đếm.

3.2.1. Phép đếm

Cho A là một tập hợp khác rỗng, để xác định số phần tử của tập hợp A ta thường thực hiện việc đếm bằng cách lần lượt gán cho các phần tử của A các số tự nhiên kế tiếp nhau, và số tự nhiên đầu tiên (được dùng để gán cho phần tử đầu tiên được xem xét) là 1. Nếu quá trình này kết thúc với số tự nhiên n (được gán cho phần tử cuối cùng) thì ta nói A là một tập hợp hữu hạn và có n phần tử. Thật ra khi thực hiện việc đếm như thế chính là thiết lập một song ánh từ A vào tập hợp $\{1, 2, ..., n\}$. Từ đó ta có thể định nghĩa phép đếm như sau:

Định nghĩa 3.7. Cho A là một tập hợp khác rỗng. Nếu tồn tại một số nguyên dương n và một song ánh f từ A vào $\{1, 2, ..., n\}$ thì ta nói A là một *tập hợp hữu hạn* và A có n phần tử. Khi đó song ánh

$$f : A \to \{1, 2, ..., n\}$$

là sẽ được xem là một *phép đếm* tập hợp A.

Tập hợp rỗng có số phần tử là 0, và cũng được xem là tập hữu hạn, $|A| = 0$.

Nếu tập hợp A không hữu hạn, ta nói A là một *tập vô hạn* và viết $|A| = \infty$.

Ghi chú: Để khái quát hóa khái niệm số phần tử đối với các tập hợp tùy ý và so sánh lực lượng của các tập hợp người ta đưa ra định nghĩa về quan hệ đồng lực lượng, và các quan hệ so sánh lực lượng khác dựa vào khái niệm ánh xạ. Chẳng hạn, hai tập hợp A và B được nói là đồng lực lượng khi tồn tại một song ánh f từ A vào B.

Tính chất. Cho A và B là các tập hợp hữu hạn. Giả sử tồn tại đơn ánh từ A vào B. Khi ấy ta có $|A| \leq |B|$.

3.2.2. Nguyên lý cộng

Cơ sở của nguyên lý cộng là mối liên hệ giữa số phần tử của một tập hợp với số phần tử của các tập hợp con tạo thành phân hoạch của tập hợp đã cho, được phát biểu trong mệnh đề sau đây.

Mệnh đề 3.1. *Cho A và B là 2 tập hợp hữu hạn rời nhau, nghĩa là $A \cap B = \emptyset$. Khi ấy ta có $|A \cup B| = |A| + |B|$.*

Một cách tổng quát: Nếu $A_1, A_2, ..., A_n$ là các tập hợp hữu hạn rời nhau, nghĩa là phần giao của hai tập hợp bất kỳ trong n tập hợp là rỗng, thì số phần tử của phần hội của các tập hợp trên bằng tổng của các số lượng phần tử trong mỗi tập hợp:

$$|A_1 \cup A_2 \cup ... \cup A_n| = |A_1| + |A_2| + \cdots + |A_n|$$

Để chứng minh mệnh đề trên cho trường hợp 2 tập hợp A và B ta có thể gọi m là số phần tử của tập hợp A và n là số phần tử của tập hợp B. Sau đó, từ việc giả sử có các song ánh

$f : A \to \{1, 2, ..., m\}$ và $g : B \to \{1, 2, ..., n\}$

ta có thể lập dễ dàng một song ánh

$h : A \cup B \to \{1, 2, ..., m + n\}$.

để đi đến kết luận $|A \cup B| = m + n$.

Trong trường hợp tổng quát ta có thể sử dụng nguyên lý qui nạp, với bước cơ sở là việc chứng minh cho trương hợp 2 tập hợp vừa trình bày ở trên.

Ghi chú: Trong trường hợp đối với hai tập hợp hữu hạn A và B tùy ý thì ta có

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Tính chất này có thể mở rộng cho trường hợp đối với n tập hợp tùy ý $A_1, A_2, ..., A_n$ như sau

$$|A_1 \cup A_2 \cup ... \cup A_n| = \sum_{1 \le r \le n} |A_r| - \sum_{1 \le r < s \le n} |A_r \cap A_s|$$
$$+ \sum_{1 \le r < s < t \le n} |A_r \cap A_s \cap A_t| - ... + (-1)^n |A_1 \cap A_2 \cap ... \cap A_n|.$$

Nguyên lý cộng: *Giả sử ta phải thực hiện công việc và để thực hiện công việc này ta có thể chọn một trong hai biện pháp khác nhau theo nghĩa là cách thực hiện biện pháp thứ nhất luôn luôn khác cách thực hiện biện pháp thứ hai. Biện pháp thứ nhất có n cách thực hiện, và đối với biện pháp thứ hai ta có m cách thực hiện. Vậy ta có $n + m$ cách thực hiện công việc.*

Ví dụ 3.11. Chúng ta cần chọn một sinh viên toán năm thứ 3 hay năm thứ 4 đi dự một hội nghị. Hỏi có bao nhiêu cách chọn lựa một sinh viên như thế biết rằng có 100 sinh viên toán học năm thứ 3 và 85 sinh viên toán học năm thứ tư?

Lời giải. Ta có thể thực hiện một trong 2 việc chọn lựa khác nhau: chọn một sinh viên toán năm 3, hoặc chọn một sinh viên toán năm 4. Để thực hiện công việc thứ nhất ta có 100 cách, và để thực hiện công việc thứ 2 ta có 85 cách. Vậy để chọn một sinh viên toán theo yêu cầu ta có 100+85 = 185 cách. ☺

Chúng ta có thể mở rộng nguyên lý cộng cho trường hợp nhiều sự chọn lựa hơn như sau: Giả sử ta phải thực hiện một công việc bằng cách chọn một trong m sự chọn lựa các biện pháp khác nhau $T_1, T_2, ..., T_m$. Để thực hiện $T_i, 1 \le i \le m$, ta có n_i cách. Vậy ta số cách thực hiện công việc trên là $n_1 + n_2 + ... + n_m$. Nguyên lý cộng dạng tổng quát này có thể được chứng minh bằng qui nạp.

Ví dụ 3.12. Một sinh viên có thể chọn một đề tài từ một trong 3 danh sách các đề tài. Số đề tài trong các danh sách đề tài lần lượt là $23, 15, 19$. Hỏi sinh viên có bao nhiêu cách chọn một đề tài.

Lời giải. Sinh viên có thể chọn một đề tài trong danh sách thứ thứ nhất theo 23 cách, trong danh sách thứ hai theo 15 cách, và trong danh sách thứ ba theo 19 cách. Do đó số cách chọn đề tài là $23 + 15 + 19 = 57$. ☺

Ví dụ 3.13. Xác định giá trị của k sau khi đoạn chương trình sau đây được thực hiện xong

Đoạn chương trình

```
k := 0
for i_1 := 1 to n_1 do
   k := k + 1;
for i_2 := 1 to n_2 do
    k := k + 1;
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
for i_m := 1 to n_m do
    k := k + 1;
```

Lời giải. Giá trị của k ban đầu là 0. Sau đó là m vòng lặp khác nhau. Mỗi thao tác lặp trong một vòng lặp là cộng thêm 1 vào k. Vòng lặp thứ $i$có n_i thao tác, và tất cả m vòng lặp không thể thực hiện 2 vòng lặp nào một cách đồng thời. Do đó số thao tác để thực hiện xong đoạn chương trình trên là $n_1 + n_2 + ... + n_m$. Đây cũng chính là giá trị cuối cùng của k.

3.2.3. Nguyên lý nhân

Cơ sở của nguyên lý nhân là mối liên hệ giữa số phần tử của một tập hợp tích Descartes với số phần tử của các tập hợp thành phần tạo nên tập hợp tích, được phát biểu trong mệnh đề sau đây

Mệnh đề 3.2. *Cho A và B là 2 tập hợp hữu hạn rời nhau. Khi ấy ta có:*

$$|A \times B| = |A|.|B|$$

Một cách tổng quát: Nếu $A_1, A_2, ..., A_n$ là các tập hợp hữu hạn thì số phần tử của tích Descartes của các tập hợp trên bằng tích của các số lượng phần tử của các tập hợp trên

$$|A_1 \times A_2 \times ... \times A_n| = |A_1|.|A_2|.....|A_n|$$

Để chứng minh mệnh đề trên cho trường hợp 2 tập hợp A và B ta có thể gọi m là số phần tử của tập hợp A và n là số phần tử của tập hợp B. Sau đó, từ việc giả sử có các song ánh

$f : A \to \{1, 2, ..., m\}$ và $g : B \to \{1, 2, ..., n\}$

ta có thể lập dễ dàng một song ánh

$h : A \cup B \to \{1, 2, ..., mn\}$.

dễ đi đến kết luận $|A \times B| = mn$.

Trong trường hợp tổng quát ta có thể sử dụng nguyên lý qui nạp, với bước cơ sở là việc chứng minh cho trương hợp 2 tập hợp vừa trình bày ở trên.

Nguyên lý nhân: *Giả sử ta phải thực hiện một thủ tục bao gồm hai công việc kế tiếp nhau. Để thực hiện công việc thứ nhất ta có n_1 cách, và ứng với mỗi cách chọn thực hiện công việc thứ nhất ta có n_2 cách thực hiện công việc thứ hai. Vậy ta có số cách thực hiện thủ tục là $n_1 \times n_2$.*

Nguyên lý nhân trên có thể được mở rộng và có dạng tổng quát như sau: Giả sử một thủ tục bao gồm m công việc kế tiếp nhau $T_1, T_2, ..., T_m$. Nếu công việc T_1 có thể được thực hiện theo n_1 cách, và sau khi chọn cách thực hiện cho T_1 ta có n_2 cách thực hiện T_2, v.v... cho đến cuối cùng, sau khi chọn cách thực hiện các công việc $T_1, T_2, ..., T_{m-1}$ ta có nm cách thực hiện T_m. Vậy ta có $n_1.n_2.....nm$ cách để thực hiện thủ tục. Nguyên lý nhân ở dạng tổng quát này có thể được chứng minh bằng qui nạp từ qui tắc nhân cho trường hợp thủ tục gồm 2 công việc.

Ví dụ 3.14. Các ghế ngồi trong một hội trường sẽ được ghi nhãn gồm một mẫu tự và một số nguyên dương không lớn hơn 100. Hỏi số ghế tối đa có thể được ghi nhãn khác nhau là bao nhiêu?

Lời giải. Thủ tục ghi nhãn cho một ghế gồm 2 việc : ghi một trong 26 mẫu tự và kế tiếp là ghi một trong 100 số nguyên dương. Qui tắc nhân cho thấy có $26 \times 100 = 2600$ cách khác nhau để ghi nhãn cho một ghế ngồi. Do đó số ghế lớn nhất có thể được ghi nhãn khác nhau là 2600. ☺

Ví dụ 3.15. Giả sử ta phải đi từ một địa điểm A đến một địa điểm C, ngang qua một địa điểm B. Để đi từ A đến B ta có 8 cách đi khác nhau, và có 6 cách đi từ B đến C. Hỏi có bao nhiêu cách để đi từ A đến C ?

Lời giải. Một cách đi từ A đến C gồm 2 việc: đi từ A đến B, rồi đi từ B đến C. Việc thứ nhất (đi từ A đến B) có 8 cách thực hiện, việc thứ hai có 6 cách thực hiện. vậy, theo nguyên lý nhân, số cách đi từ A đến C là $8 \times 6 = 48$. ☺

Ví dụ 3.16. Hỏi có bao nhiêu chuỗi bit khác nhau có độ dài 8 (tức là gồm 8 bits) ?

Lời giải. Mỗi bit có thể được chọn theo 2 cách, vì mỗi bit là 0 hoặc 1. Do đó, qui tắc nhân cho phép ta kết luận rằng có $2^8 = 256$ chuỗi bit có độ dài 8. ☺

Ví dụ 3.17. Một mã bao gồm 6 ký tự, trong đó gồm 3 mẫu tự rồi đến 3 ký số thập phân. Hỏi có bao nhiêu mã khác nhau?

Lời giải. Có 26 cách chọn cho mỗi mẫu tự và có 10 cách chọn cho mỗi ký số thập phân. Do đó, theo qui tắc nhân, có tất cả $26.26.26.10.10.10 = 17576000$ mã khác nhau. ☺

Ví dụ 3.18. Có bao nhiêu ánh xạ đi từ một tập hợp gồm m phần tử vào một tập hợp gồm n phần tử?

Lời giải. Một ánh xạ đi từ tập A gồm m phần tử vào một tập hợp B gồm n phần tử tương ứng với việc chọn lựa một trong n phần tử của B cho mỗi phần tử của A. Do đó, theo qui tắc nhân, có $n.n.....n = nm$ ánh xạ từ A vào B. ☺

Ví dụ 3.19. Có bao nhiêu đơn ánh đi từ một tập hợp gồm m phần tử vào một tập hợp gồm n phần tử?

Lời giải. Trước hết ta nhận xét rằng khi $m > n$ thì không có một đơn ánh nào đi từ một tập hợp gồm m phần tử vào một tập hợp gồm n phần tử. Vậy, cho $m \leq n$. Giả sử các phần tử trong miền xác định của ánh xạ là $a_1, a_2, ..., a_m$. Có n cách chọn ảnh qua ánh xạ cho phần tử a_1 . Vì ánh xạ là đơn ánh nên đối với phần tử a_2 ta chỉ có $n-1$ cách chọn ảnh tương ứng (do giá trị ảnh được chọn cho a_1 không thể được chọn lại cho a_2). Tổng quát, giá trị ảnh của phần tử a_k chỉ có thể được chọn theo $n-k+1$ cách. Theo qui tắc nhân, có $n.(n-1).....(n-m+1)$ đơn ánh đi từ một tập hợp gồm m phần tử vào một tập hợp gồm n phần tử. ☺

Ví dụ 3.20. Phương án đánh số điện thoại.

Giả sử một số điện thoại gồm 10 ký số được chia thành 3 nhóm: 2 nhóm gồm 3 ký số và một nhóm 4 ký số. Do một số lý do nào đó, có một số hạn chế trên các ký số của số điện thoại. Để xác định dạng hợp lệ của một số điện thoại. ta dùng ký hiệu X để chỉ một ký số có thể lấy giá trị từ 0 đến 9, N để chỉ một ký số từ 2 đến 9, và Y chỉ một ký số là 0 hoặc 1.

Chúng ta có 2 phương án để đánh số điện thoại : một phương án cũ và một phương án mới. Theo phương án cũ, số điện thoại có dạng NYX NNX XXXX; và theo phương án mới thì số điện thoại có dạng NXX NXX XXXX.

Hỏi số lượng số điện thoại khác nhau của mỗi phương án là bao nhiêu?

Lời giải. Do qui tắc nhân, đối với phương án đánh số điện thoại cũ, số trường hợp khác nhau của mỗi nhóm ký số trong 3 nhóm lần lượt là : $8.2.10 = 160$ (ứng với dạng NYX), $8.8.10 = 640$ (ứng với dạng NNX), và $10.10.10.10 = 10000$ (ứng với dạng XXXX).

Vậy, trong phương án đánh số điện thoại cũ, số lượng số điện thoại là $160.640.10000 = 1024000000$.

Tương tự Số lượng số điện thoại trong phương án đánh số mới là

$(8.10.10).(8.10.10).(10.10.10.10) = 800.800.10000 = 6400000000$. ☺

Ví dụ 3.21. Sử dụng qui tắc nhân để chứng minh rằng một tập hợp S hữu hạn có tất cả $2^{|S|}$ tập hợp con khác nhau.

Lời giải. Cho S là một tập hợp hữu hạn, $|S| = n$. Liệt kê các phần tử của S theo một thứ tự bất kỳ. Ta có thể thấy rằng có sự tương ứng một-một trên (song ánh) giữa các tập hợp con của S và tập hợp các chuỗi bit gồm n bits. Một tập con của S được cho tương ứng với một chuỗi bits có bit thứ i là 1 nếu phần tử thứ 4 trong danh sách liệt kê thuộc tập hợp con, và bit thứ i là 0 trong trường hợp ngược lại. Bởi qui tắc nhân, có 2^n chuỗi bit gồm n bits. Do đó, S có 2^n tập hợp con. ☺

Dưới đây chúng ta sẽ xem xét một số bài toán về phép đếm phức tạp hơn. Nó đòi hỏi chúng ta phải sử dụng cả nguyên lý cộng lẫn nguyên lý nhân.

Ví dụ 3.22. Trong một version của ngôn ngữ BASIC tên của một biến là một chuỗi gồm 1 hoặc 2 ký tự, mỗi ký tự là mẫu tự hoặc ký số thập lục phân và không phân biệt giữa chữ in hoa và chữ thường. Hơn nữa, một tên biến phải bắt đầu bởi một mẫu tự và tên biến phải khác với 5 chuỗi gồm 2 ký tự đã được dành riêng cho ngôn ngữ. Hỏi có bao nhiêu tên biến khác nhau trong version này của BASIC.

Lời giải. Đặt V là số tên biến khác nhau trong version này của BASIC, V_1 là số biến gồm một ký tự, và V_2 là số biến gồm hai ký tự. Theo qui tắc cộng ta có $V = V_1 + V_2$.

Vì biến gồm một ký tự phải là một mẫu tự nên $V_1 = 26$. Ngoài ra, theo qui tắc nhân ta có 26.36 chuỗi có độ dài 2 với ký tự đi đầu là mẫu tự và ký tự kế là mẫu tự hoặc ký số thập phân. Tuy nhiên, có 5 chuỗi bị loại ra nên $V_2 = 26.36 - 5 = 931$. Vậy có $V = V_1 + V_2 = 26 + 931 = 957$ tên khác nhau cho các biến của version này của BASIC. ☺

Ví dụ 3.23. Mỗi người sử dụng trên một hệ thống máy tính có một "password" dài từ 6 đến 8 ký tự, trong đó mỗi ký tự là một chữ in hoa hoặc là một ký số thập phân. Mỗi "password" phải có ít nhất một ký số. Hỏi có bao nhiêu password khác nhau?

Lời giải. Đặt P là số lượng tất cả các "password", và P_6, P_7, P_8 lần lượt là số các "password" có độ dài $6, 7, 8$. Do qui tắc cộng ta có $P = P_6 + P_7 + P_8$. Chúng ta sẽ tính P_6, P_7, và P_8. Tính trực tiếp P_6 tương đối khó. Để tính P_6 cho dễ, ta tính số chuỗi có độ dài 6 gồm các chữ in hoa hay ký số thập phân, kể cả các chuỗi không có ký số thập phân, và trừ cho số chuỗi (với độ dài 6) không có ký số thập phân. Theo qui tắc nhân, số chuỗi gồm 6 ký tự là 36^6 và số chuỗi không có ký số là 26^6. Suy ra

$$\begin{aligned} P_6 = 366 - 266 &= 2176782336 - 308915776 \\ &= 1867866560. \end{aligned}$$

Tương tự, ta có thể tính ra được

$$\begin{aligned} P_7 = 367 - 267 &= 78364164096 - 8031810176 \\ &= 70332353920. \end{aligned}$$

và

$$\begin{aligned} P_8 = 368 - 268 &= 2821109907456 - 208827064576 \\ &= 2612282842880. \end{aligned}$$

Từ đó ta tính được $P = P_6 + P_7 + P_8 = 2684483063360.$ ☺

3.2.4. Nguyên lý bù trừ

Khi hai công việc có thể được làm đồng thời, ta không thể dùng quy tắc cộng để tính số cách thực hiện nhiệm vụ gồm cả hai việc. Để tính đúng số cách thực hiện nhiệm vụ này ta cộng số cách làm mỗi một trong hai việc rồi trừ đi số cách làm đồng thời cả hai việc. Ta có thể phát biểu nguyên lý đếm này bằng ngôn ngữ tập hợp. Cho A_1, A_2 là hai tập hữu hạn, khi đó

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|.$$

Từ đó với ba tập hợp hữu hạn A_1, A_2, A_3, ta có:

$$|A_1 \cup A_2 \cup A_3| = |A_1| + |A_2| + |A_3| - |A_1 \cap A_2| - |A_2 \cap A_3| - |A_3 \cap A_1| + |A$$

và bằng quy nạp, với k tập hữu hạn $A_1, A_2, ..., A_k$ ta có:

$$|A_1 \cup A_2 \cup ... \cup A_k| = N_1 - N_2 + N_3 - ... + (-1)^{k-1} N_k,$$

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

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

Bây giờ 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 sao cho không thỏa mãn bất kỳ một tính chất A_m nào. Gọi $\overline{N}$ là số cần đếm, N là số phần tử của U. Ta có

$$\overline{N} = N - |A_1 \cup A_2 \cup ... \cup A_k| = N - N_1 + N_2 - ... + (-1)^k N_k,$$

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 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 3.24. Có n lá thư và n phong bì ghi sẵn địa chỉ. Bỏ ngẫu nhiên các lá thư vào các phong bì. Hỏi xác suất để xảy ra không một lá thư nào đúng địa chỉ.

Lời giải. Mỗi phong bì có n cách bỏ thư vào, nên có tất cả $n!$ cách bỏ thư. Vấn đề còn lại là đếm số cách bỏ thư sao cho không lá thư nào đúng địa chỉ. Gọi U là tập hợp các cách bỏ thư và A_m là tính chất lá thư thứ m bỏ đúng địa chỉ. Khi đó theo công thức về nguyên lý bù trừ ta có

$$\overline{N} = n! - N_1 + N_2 - ... + (-1)^n N_n,$$

trong đó $N_m (1 \leq m \leq n)$ là số tất cả các cách bỏ thư sao cho có m lá thư đúng địa chỉ. Nhận xét rằng, N_m là tổng theo mọi cách lấy m lá thư từ n lá, với mỗi cách lấy m lá thư, có $(n-m)!$ cách bỏ để m lá thư này đúng địa chỉ, ta nhận được

$$N_m = C_n^m (n-m)! = \frac{n!}{k!} \text{ và } \overline{N} = n! \left(1 - \frac{1}{1!} + \frac{1}{2!} + \cdots + (-1)^n \frac{1}{n!}\right),$$

trong đó $C_n^m = \dfrac{n!}{m!(n-m)!}$ là tổ hợp chập m của tập n phần tử (số cách chọn m đối tượng trong n đối tượng được cho). Từ đó xác suất cần tìm là $1 - \frac{1}{1!} + \frac{1}{2!} + \cdots + (-1)^n \frac{1}{n!}$. điều lý thú là xác suất này dần đến e^{-1} (nghĩa là còn $> \frac{1}{3}$) khi n khá lớn.

Số $\overline{N}$ trong bài toán này được gọi là số mất thứ tự và được ký hiệu là D_n. Dưới đây là một vài giá trị của D_n, cho ta thấy D_n tăng nhanh như thế nào so với n

n	2	3	4	5	6	7	8	9	10	11
D_n	1	2	9	44	265	1851	14833	133496	1334961	14684570

Ví dụ 3.25. Mỗi sinh viên học lớp toán rời rạc hoặc là giỏi Toán, hoặc là giỏi Tin, hoặc là giỏi cả hai môn này. Trong lớp có bao nhiêu sinh viên nếu có 38 người giỏi Tin, 23 người giỏi Toán và 7 giỏi cả 3 môn.

Lời giải. Đặt

$A =$"Số sinh viên giỏi tin". Suy ra $|A| = 38$.

$B =$"Số sinh viên giỏi toán". Suy ra $|B| = 23$.

$A \cap B =$"Số sinh viên giỏi tin và toán". Suy ra $|A \cap B| = 7$.

Ta có $|A \cup B| = |A| + |B| - |A \cap B| = 38 + 23 - 7 = 54$.

Vậy số sinh viên của lớp là 54.

Ví dụ 3.26. Biết rằng có 1232 sinh viên học tiếng Tây Ban Nha; 879 sinh viên học tiếng Pháp và 114 sinh viên học tiếng Nga; 103 sinh viên học cả Tây Ban Nha và Pháp; 23 sinh viên học cả Tây Ban Nha và Nga; 14 sinh viên học cả Pháp và Nga; Nếu tất cả 2092 sinh viên đều theo hcọ ít nhất một ngoại ngữ thì có bao nhiêu sinh viên học cả ba thứ tiếng?

Lời giải. Đặt

$S =$"sinh viên học tiếng Tây Ban Nha". Suy ra $|S| = 1232$;

$F =$"sinh viên học tiếng Pháp". Suy ra $|F| = 879$;

$R =$"sinh viên học tiếng Pháp". Suy ra $|R| = 114$;

Khi đó

$S \cap F$ ="sinh viên học cả tiếng Tây Ban Nha và Pháp". Suy ra $|S \cap F| = 103$;

$S \cap R$ ="sinh viên học cả tiếng Tây Ban Nha và Nga". Suy ra $|S \cap R| = 23$;

$F \cap R$ ="sinh viên học cả tiếng Pháp và Nga". Suy ra $|S \cap R| = 14$;

$S \cap F \cap R$ ="sinh viên học tất cả ba thứ tiếng". Suy ra $|F \cap S \cap R| = x$.

Ta có

$|S \cup F \cup R| = |S| + |F| + |R| - |S \cap F| - |S \cap R| - |F \cap R| + |S \cap F \cap R|$

$= 1232 + 879 + 114 - 103 - 23 - 14 + x = 2092$

Hay $2085 + x = 2092$, suy ra $x = 7$. Vậy có 7 sinh viên học cả ba thứ tiếng.

3.3 Nguyên lý Dirichlet

3.3.1. Giới thiệu nguyên lý

Giả sử có một đàn chim bồ câu bay vào chuồng. Nếu số chim nhiều hơn số ngăn chuồng thì ít nhất trong một ngăn có nhiều hơn một con chim. Nguyên lý này dĩ nhiên là có thể áp dụng cho các đối tượng không phải là chim bồ câu và chuồng chim.

Mệnh đề 3.3. *(**Nguyên lý**) Nếu có $k+1$ (hoặc nhiều hơn) đồ vật được đặt vào trong k hộp thì tồn tại một hộp có ít nhất hai đồ vật.*

Chứng minh. Giả sử không có hộp nào trong k hộp chứa nhiều hơn một đồ vật. Khi đó tổng số vật được chứa trong các hộp nhiều nhất là bằng k. Điều này trái giả thiết là có ít nhất $k+1$ vật.

Nguyên lý này thường được gọi là nguyên lý Dirichlet, mang tên nhà toán học người Đức ở thế kỷ 19. Ông thường xuyên sử dụng nguyên lý này trong công việc của mình.

Ví dụ 3.27. Trong bất kỳ một nhóm 367 người thế nào cũng có ít nhất hai người có ngày sinh nhật giống nhau.

Lời giải. Bởi vì chỉ có tất cả 366 ngày sinh nhật khác nhau. ☺

Ví dụ 3.28. Trong kỳ thi học sinh giỏi, điểm bài thi được đánh giá bởi một số nguyên trong khoảng từ 0 đến 100. Hỏi rằng ít nhất có bao nhiêu học sinh dự thi để cho chắc chắn tìm được hai học sinh có kết quả thi như nhau?

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

Ví dụ 3.29. Trong số những người có mặt trên trái đất, phải tìm được hai người có hàm răng giống nhau.

Lời giải. Nếu xem mỗi hàm răng gồm 32 cái như là một xâu nhị phân có chiều dài 32, trong đó răng còn ứng với bit 1 và răng mất ứng với bit 0, thì có tất cả 2 32 = 4.294.967.296 hàm răng khác nhau. Trong khi đó số người trên hành tinh này là vượt quá 5 tỉ, nên theo nguyên lý Dirichlet ta có điều cần tìm. ☺

3.3.2. Nguyên lý Dirichlet tổng quát

Ta dùng ký hiệu $\lceil x \rceil$ là giá trị của hàm trần tại số thực x, đó là số nguyên nhỏ nhất có giá trị lớn hơn hoặc bằng x. Khái niệm này đối ngẫu với $[x]$ (hoặc $\lfloor x \rfloor$) – giá trị của hàm sàn hay hàm phần nguyên tại x – là số nguyên lớn nhất có giá trị nhỏ hơn hoặc bằng x.

Mệnh đề 3.4. *Nếu có N đồ vật được đặt vào trong k hộp thì sẽ tồn tại một hộp chứa ít nhất* $\left\lceil \frac{N}{k} \right\rceil$ *đồ vật.*

Chứng minh. Giả sử mọi hộp đều chứa ít hơn $\left\lceil \frac{N}{k} \right\rceil$ vật. Khi đó tổng số đồ vật là

$$\leq k\left(\left\lceil \frac{N}{k} \right\rceil - 1\right) < k\frac{N}{k} = N$$

Điều này mâu thuẩn với giả thiết là có N đồ vật cần xếp.

Ví dụ 3.30. Trong 100 người, có ít nhất 9 người sinh cùng một tháng.

Lời giải. Xếp những người sinh cùng tháng vào một nhóm. Có 12 tháng tất cả. Vậy theo nguyên lý Dirichlet, tồn tại một nhóm có ít nhất $\left\lceil \dfrac{100}{12} \right\rceil = 9$ người. ☺

Ví dụ 3.31. Có năm loại học bổng khác nhau. Hỏi rằng phải có ít nhất bao nhiêu sinh viên để chắc chắn rằng có ít ra là 6 người cùng nhận học bổng như nhau.

Lời giải. Gọi N là số sinh viên, khi đó $\left\lceil \dfrac{N}{5} \right\rceil = 6$ khi và chỉ khi $5 < \dfrac{N}{5} \leq 6$ hay $25 < N \leq 30$. Vậy số N cần tìm là 26. ☺

Ví dụ 3.32. Số mã vùng cần thiết nhỏ nhất phải là bao nhiêu để đảm bảo 25 triệu máy điện thoại trong nước có số điện thoại khác nhau, mỗi số có 9 chữ số (giả sử số điện thoại có dạng 0XX - 8XXXXX với X nhận các giá trị từ 0 đến 9).

Lời giải. Có $10^7 = 10.000.000$ số điện thoại khác nhau có dạng 0XX - 8XXXXX. Vì vậy theo nguyên lý Dirichlet tổng quát, trong số 25 triệu máy điện thoại ít nhất có

$$\left\lceil \frac{25.000.000}{10.000.000} \right\rceil = 3$$

có cùng một số. Để đảm bảo mỗi máy có một số cần có ít nhất 3 mã vùng. ☺

3.3.3. Một số ứng dụng của nguyên lý Dirichlet

Trong nhiều ứng dụng thú vị của nguyên lý Dirichlet, khái niệm đồ vật và hộp cần phải được lựa chọn một cách khôn khéo. Trong phần nay có vài thí dụ như vậy.

Ví dụ 3.33. Trong một phòng họp có n người, bao giờ cũng tìm được 2 người có số người quen trong số những người dự họp là như nhau.

Lời giải. Số người quen của mỗi người trong phòng họp nhận các giá trị từ 0 đến $n - 1$. Rõ ràng trong phòng 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). Vì vậy theo số lượng người quen, ta chỉ có thể phân n người ra thành $n-1$ nhóm. Vậy theo nguyên lý Dirichlet tồn tai một nhóm có ít nhất 2 người, tức là luôn tìm được ít nhất 2 người có số người quen là như nhau. ☺

Ví dụ 3.34. Trong một tháng gồm 30 ngày, một đội bóng chuyền thi đấu mỗi ngày ít nhất 1 trận nhưng chơi không quá 45 trận. Chứng minh rằng tìm được một giai đoạn gồm một số ngày liên tục nào đó trong tháng sao cho trong giai đoạn đó đội chơi đúng 14 trận.

Lời giải. Gọi a_j là số trận mà đội đã chơi từ ngày đầu tháng đến hết ngày j. Khi đó

$$1 \leq a_1 < a_2 < ... < a_{30} < 45$$

$$15 \leq a_1 + 14 < a_2 + 14 < ... < a_{30} + 14 < 59.$$

Sáu mươi số nguyên $a_1, a_2, ..., a_{30}, a_1 + 14, a_2 + 14, ..., a_{30} + 14$ nằm giữa 1 và 59. Do đó theo nguyên lý Dirichlet có ít nhất 2 trong 60 số này bằng nhau. Vì vậy tồn tại i và j sao cho $a_i = a_j + 14 (j < i)$. Điều này có nghĩa là từ ngày $j+1$ đến hết ngày i đội đã chơi đúng 14 trận. ☺

Ví dụ 3.35. Chứng tỏ rằng trong $n+1$ số nguyên dương không vượt quá 2n, tồn tại ít nhất một số chia hết cho số khác.

Lời giải. Ta viết mỗi số nguyên $a_1, a_2, ..., a_{n+1}$ dưới dạng $a_j = 2^{k_j} q_j$ trong đó k_j là số nguyên không âm còn q_j là số dương lẻ nhỏ hơn $2n$. Vì chỉ có n số nguyên dương lẻ nhỏ hơn $2n$ nên theo nguyên lý Dirichlet tồn tại i và j sao cho $q_i = q_j = q$. Khi đó $a_i = 2^{k_i} q$ và $a_j = 2^{k_j} q$. Vì vậy, nếu $k_i \leq k_j$ thì a_j chia hết cho a_i còn trong trường hợp ngược lại ta có a_i chia hết cho a_j . ☺

Thí dụ cuối cùng trình bày cách áp dụng nguyên lý Dirichlet vào lý thuyết tổ hợp mà vẫn quen gọi là *lý thuyết Ramsey*, tên của nhà toán học người Anh. Nói chung, lý thuyết Ramsey giải quyết những bài toán phân chia các tập con của một tập các phần tử.

Ví dụ 3.36. Giả sử trong một nhóm 6 người mỗi cặp hai hoặc là bạn hoặc là thù. Chứng tỏ rằng trong nhóm có ba người là bạn lẫn nhau hoặc có ba người là kẻ thù lẫn nhau.

Lời giải. Gọi A là một trong 6 người. Trong số 5 người của nhóm hoặc là có ít nhất ba người là bạn của A hoặc có ít nhất ba người là kẻ thù của A, điều này suy ra từ nguyên lý Dirichlet tổng quát, vì $\left\lceil \frac{5}{2} \right\rceil = 3$. Trong trường hợp đầu ta gọi B, C, D là bạn của A. nếu trong ba người này có hai người là bạn thì họ cùng với A lập thành một bộ ba người bạn lẫn nhau, ngược lại, tức là nếu trong ba người B, C, D không có ai là bạn ai cả thì chứng tỏ họ là bộ ba người thù lẫn nhau. Tương tự có thể chứng minh trong trường hợp có ít nhất ba người là kẻ thù của A. ☺

3.4 Hoán vị, chỉnh hợp và tổ hợp

3.4.1. Hoán vị và chỉnh hợp

Định nghĩa 3.8. *Hoán vị* của một tập các đối tượng khác nhau là một cách sắp xếp có thứ tự các đối tượng này.

Một cách sắp xếp có thứ tự r phần tử của một tập hợp n phần tử được gọi là *chỉnh hợp chập r của n* phần tử. Ký hiệu P_n^r là số chỉnh hợp chập r của tập n phần tử.

Định lý 3.1. *Số chỉnh hợp chập r của n phần tử là*

$$P_n^r = n(n-1)(n-2)...(n-r+1) = \frac{n!}{(n-r)!}$$

Ví dụ 3.37. Cho $A = \{1, 2, 3\}$ có bao nhiêu cách sắp xếp thứ tự hai phần tử trong tệp A?

Lời giải. $P_3^2 = 3.2 = 6$. Đó là các cặp

$$\{1, 2\}, \{2, 1\}, \{1, 3\}, \{3, 1\}, \{2, 3\}, \{3, 2\}.$$

Ví dụ 3.38. Có bao nhiêu cách chọn 4 cầu thủ khác nhau trong 10 cầu thủ của đội bóng?

Lời giải. $P_{10}^4 = 10(10-1)(10-2)(10-3) = 10.9.8.7 = 5040.$

Chú ý. Số chỉnh hợp chập n của n phần tử là $P_n^n = n!$.

Nghĩa là số hoán vị của tập gồm n phần tử là $P_n^n = n!$,

Ví dụ 3.39. Giả sử có 8 vận động viên chạy thi tốc độ cự ly 2000m. Người đến đích đầu tiên được trao huy chương vàng, người đến đích thứ hai được trao huy chương bạc và người đến đích thứ ba được trao huy chương đồng. Hỏi có bao nhiêu cách trao huy chương Vàng, Bạc và Đồng cho 8 vận động viên trên.

Lời giải. Số cách trao huy chương Vàng, Bạc và Đồng cho 8 vận động viên chính là chỉnh hợp chập 3 của 8. Nghĩa là $P_8^3 = 8.7.6 = 336$ cách.

3.4.2. Tổ hợp và định lý nhị thức

Định nghĩa 3.9. Một tổ hợp chập r của một tập hợp là một cách chọn không có thứ tự r phần tử của tập hợp đã cho.

- Một tổ hợp chập r chính là một tập hợp con r của tập hợp đã cho.
- Số tổ hợp chập r của tập hợp n ký hiệu là C_n^r, $0 \le r \le n$. Ta có kết quả sau

Định lý 3.2. *Số tổ hợp chập r của n phần tử là*

$$C_n^r = \frac{n!}{r!(n-r)!}.$$

Chứng minh. $P(n,r) = C_n^r P(r,r)$ hay

$$C_n^r = \frac{P(n,r)}{r!} = \frac{n!}{r!(n-r)!}.$$

Hệ quả 3.1. *Cho n, r là các số nguyên không âm sao cho $r \le n$. Khi đó ta có $C_n^r = C_n^{n-r}$.*

Chứng minh. Theo Định lý 3.2 ta có $C_n^r = \dfrac{n!}{r!(n-r)!}$. Vậy

$$C_n^{n-r} = \frac{n!}{(n-r)!(n-(n-r))!} = \frac{n!}{r!(n-r)!} = C_n^r.$$

Hệ quả 3.2. *Cho n, r là các số nguyên không âm sao cho $n \geq k$. Khi đó ta có*
$$C_{n+1}^{k} = C_n^{k-1} + C_n^k.$$

Hệ quả 3.3. $\sum\limits_{k=0}^{n} C_n^k = 2^n$ *(n nguyên dương).*

Chứng minh. Số các tập con của tập gồm n phần tử là
$$2^n = C_n^0 + C_n^1 + \cdots + C_n^n = \sum_{k=0}^{n} C_n^k.$$

Ví dụ 3.40. Cho $A = \{a, b, c, d\}$ Tìm C_4^2, C_4^3.

Lời giải. Ta có $C_4^2 = \dfrac{4!}{2!(4-2)!} = \dfrac{1.2.3.4}{1.2.1.2} = 6.$

Cụ thể là các cặp $\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}$.

Ta có $C_4^3 = \dfrac{4!}{3!(4-3)!} = 4.$

Cụ thể là các cặp $\{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}$.

Ví dụ 3.41. Có bao nhiêu cách tuyển 5 trong số 10 cầu thủ của một đội quần vợt để đi thi đấu?

Lời giải. Đó chính là $C_{10}^5 = \dfrac{10!}{5!(10-5)!} = 252.$

Ví dụ 3.42. Cho $A = \{a, b, c\}$ và $B = \{a, b, c, d\}$. Hãy liệt kê tất cả các hoán vị của A và của B.

Lời giải. Đối với A có $3! = 6$ hoán vị là $abc, acb, cab, cba, bac, bca$.

Đối với B có $4! = 24$ hoán vị là $abcd, abcd, adbc, \ldots.$

Ví dụ 3.43. Giả sử $A = \{1, 2, 3, 4\}$.

a) Tìm các chỉnh hợp chập 3 của A.

b) Tìm các tổ hợp chập 3 của A.

Lời giải. a) $P_5^3 = 5.4.3 = 60.$

b) $C_5^3 = \dfrac{5!}{3!2!} = \dfrac{4.5}{2} = 10.$

Ví dụ 3.44. Có bao nhiêu khả năng có thể xảy ra đối với các vị trí thứ nhất, thứ hai, thứ ba trong cuộc đua có 12 con ngựa, nếu mọi thứ tự tới đích đều có thể xẩy ra?

Lời giải. Số khả năng tìm chính là $P_{12}^3 = 1320$.

Ví dụ 3.45. Có bao nhiêu cách chọn một tập hợp 5 chữ từ bảng chữ cái tiếng Anh?

Lời giải. Số cách chọn chính là

$$C_{26}^5 = \frac{26!}{5!21!} = \frac{22.23.24.25.26}{5!} = \frac{7893600}{120} = 65780.$$

Ví dụ 3.46. Một tập hợp có 100 phần tử có bao nhiêu tập con có nhiều hơn hai phần tử?

Lời giải. Theo Hệ quả 3.3 ta có $\sum\limits_{i=0}^{100} C_n^i = 2^{100}$, hay

Số tập con của tập 100 phần tử có nhiều hơn hai phần tử là

$$\begin{aligned} 2^{100} - \left(\frac{100!}{0!100!} + \frac{100!}{1!99!} + \frac{100}{2!98!}\right) &= 2^{100} - \left(1 + 100 + \frac{99.100}{2}\right) \\ &= 2^{100} - (101 + 4950) \\ &= 2^{100} - 5051. \end{aligned}$$

Định lý 3.3. *Cho x, y là hai biến và n là một số nguyên dương. Khi đó ta có*

$$\begin{aligned} (x+y)^n &= \sum_{i=1}^{n} C_n^i x^{n-i} y^i \\ &= C_n^0 x^n + C_n^1 x^{n-1} y + \cdots + C_n^{n-1} x y^{n-1} + C_n^n y^n. \end{aligned}$$

Ví dụ 3.47. Tìm khai triển $(x+y)^5$.

Lời giải. Ta có

$$\begin{aligned} (x+y)^5 = C_5^0 x^5 + C_5^1 x^4 y + C_5^2 x^3 y^2 + C_5^3 x^2 y^3 \\ + + C_5^4 x y^4 + C_5^5 y^5, \end{aligned}$$

với $C_5^0 = 1, C_5^1 = 5, C_5^2 = 10$.

$$(x+y)^5 = x^5 + 5x^4 y + 10x^3 y^2 + 10x^2 y^3 + 5xy^4 + y^5.$$

Ví dụ 3.48. Trong khai triển $(x+y)^{100}$ có bao nhiêu số hạng?

Lời giải. Có 101 số hạng là $C_{100}^{0}, C_{100}^{1}, ..., C_{100}^{100}$.

Ví dụ 3.49. Tìm hệ số x^9 trong khai triển $(2-x)^{19}$.

Lời giải.

$$(2-x)^{19} = C_{19}^{0}2^{19} + C_{19}^{1}2^{18}(-x) + C_{19}^{2}2^{17}(-x)^2 + \cdots + C_{19}^{9}2^{10}(-x)^9 + \cdots \quad (3.1)$$

Vậy số hạng của x^9 là $-C_{19}^{9}2^{10} = -94595072$.

3.5 Chỉnh hợp và tổ hợp suy rộng

3.5.1. Chỉnh hợp có lặp

Định nghĩa 3.10. Một cách sắp xếp có thứ tự k phần tử có thể lặp lại của một tập n phần tử được gọi là *một chỉnh hợp lặp* chập k từ tập n phần tử.

Nếu A là tập gồm n phần tử đó thì mỗi chỉnh hợp như thế là một phần tử của tập A^k . Ngoài ra, mỗi chỉnh hợp lặp chập k từ tập n phần tử là một hàm từ tập k phần tử vào tập n phần tử. Vì vậy số chỉnh hợp lặp chập k từ tập n phần tử là n^k.

3.5.2. Tổ hợp lặp

Định nghĩa 3.11. *Một tổ hợp lặp chập* k của một tập hợp là một cách chọn không có thứ tự k phần tử có thể lặp lại của tập đã cho.

Như vậy một tổ hợp lặp kiểu này là một dãy không kể thứ tự gồm k thành phần lấy từ tập n phần tử. Do đó có thể là $k > n$.

Mệnh đề 3.5. *Số tổ hợp lặp chập k từ tập n phần tử bằng* C_{n+k-1}^{k}

Chứng minh. Mỗi tổ hợp lặp chập k từ tập n phần tử có thể biểu diễn bằng một dãy $n-1$ thanh đứng và k ngôi sao. Ta dùng $n-1$

thanh đứng để phân cách các ngăn. Ngăn thứ i chứa thêm một ngôi sao mỗi lần khi phần tử thứ i của tập xuất hiện trong tổ hợp. Chẳng hạn, tổ hợp lặp chập 6 của 4 phần tử được biểu thị bởi

$$* * | * || * **$$

mô tả tổ hợp chứa đúng 2 phần tử thứ nhất, 1 phần tử thứ hai, không có phần tử thứ 3 và 3 phần tử thứ tư của tập hợp.

Mỗi dãy $n-1$ thanh và k ngôi sao ứng với một xâu nhị phân độ dài $n+k-1$ với k số 1. Do đó số các dãy $n-1$ thanh đứng và k ngôi sao chính là số tổ hợp chập k từ tập $n+k-1$ phần tử. Đó là điều cần chứng minh.

Ví dụ 3.50. Có bao nhiêu cách chọn 5 tờ giấy bạc từ một két đựng tiền gồm những tờ 1000đ, 2000đ, 5000đ, 10.000đ, 20.000đ, 50.000đ, 100.000đ. Giả sử thứ tự mà các tờ tiền được chọn là không quan trọng, các tờ tiền cùng loại là không phân biệt và mỗi loại có ít nhất 5 tờ.

Lời giải. Vì ta không kể tới thứ tự chọn tờ tiền và vì ta chọn đúng 5 lần, mỗi lần lấy một từ 1 trong 7 loại tiền nên mỗi cách chọn 5 tờ giấy bạc này chính là một tổ hợp lặp chập 5 từ 7 phần tử. Do đó số cần tìm là $C^5_{7+5-1} = 462$.

Ví dụ 3.51. Phương trình $x_1 + x_2 + x_3 = 15$ có bao nhiêu nghiệm nguyên không âm?

Lời giải. Chúng ta nhận thấy mỗi nghiệm của phương trình ứng với một cách chọn 15 phần tử từ một tập có 3 loại, sao cho có x_1 phần tử loại 1, x_2 phần tử loại 2 và x_3 phần tử loại 3 được chọn. Vì vậy số nghiệm bằng số tổ hợp lặp chập 15 từ tập có 3 phần tử và bằng $C^{15}_{3+15-1} = 136$.

3.5.3. Hoán vị của tập hợp có lặp

Trong bài toán đếm, một số phần tử có thể giống nhau. Khi đó cần phải cẩn thận, tránh đếm chúng hơn một lần. Ta xét thí dụ sau.

Ví dụ 3.52. Có thể nhận được bao nhiêu xâu khác nhau bằng cách sắp xếp lại các chữ cái của từ SUCCESS?

Lời giải. Vì một số chữ cái của từ SUCCESS là như nhau nên câu trả lời không phải là số hoán vị của 7 chữ cái được. Từ này chứa 3 chữ S, 2 chữ C, 1 chữ U và 1 chữ E. Để xác định số xâu khác nhau có thể tạo ra được ta nhận thấy có C_7^3 cách chọn 3 chỗ cho 3 chữ S, còn lại 4 chỗ trống. Có C_4^2 cách chọn 2 chỗ cho 2 chữ C, còn lại 2 chỗ trống. Có thể đặt chữ U bằng C_2^1 cách và C_1^1 cách đặt chữ E vào xâu. Theo nguyên lý nhân, số các xâu khác nhau có thể tạo được là:

$$C_7^3.C_4^2.C_2^1.C_1^1 = \frac{71412!1!}{3!4!2!2!1!1!1!0!} = \frac{7!}{3!2!1!1!} = 420.$$

Mệnh đề 3.6. *Số hoán vị của n phần tử trong đó có n_1 phần tử như nhau thuộc loại 1, n_2 phần tử như nhau thuộc loại 2, ..., và n_k phần tử như nhau thuộc loại k, bằng*

$$\frac{n!}{n_1!n_2!...n_k!}.$$

Chứng minh. Để xác định số hoán vị trước tiên chúng ta nhận thấy có $C_n^{n_1}$ cách giữ n_1 chỗ cho n_1 phần tử loại 1, còn lại $n - n_1$ chỗ trống. Sau đó có $C_{n-n_1}^{n_2}$ cách đặt n_2 phần tử loại 2 vào hoán vị, còn lại $n - n + 1 - n_2$ chỗ trống. Tiếp tục đặt các phần tử loại 3, loại 4,..., loại $k - 1$vào chỗ trống trong hoán vị. Cuối cùng có $C_{n-n_1-...-n_{k-1}}^{n_k}$ cách đặt n_k phần tử loại k vào hoán vị. Theo quy tắc nhân tất cả các hoán vị có thể là

$$C_n^{n_1}.C_{n-n_1}^{n_2}.....C_n - n_1 - ... - n_{k-1} = \frac{n!}{n_1!n_2!...n_k!}.$$

3.5.4. Sự phân bố các đồ vật vào trong hộp.

Ví dụ 3.53. Có bao nhiêu cách chia những xấp bài 5 quân cho mỗi một trong 4 người chơi từ một cỗ bài chuẩn 52 quân?

Lời giải. Người đầu tiên có thể nhận được 5 quân bài bằng C_{52}^5 cách. Người thứ hai có thể được chia 5 quân bài bằng C_{47}^5 cách, vì chỉ còn 47 quân bài. Người thứ ba có thể nhận được 5 quân bài bằng C_{42}^5 cách. Cuối cùng, người thứ tư nhận được 5 quân bài bằng C_{37}^5 cách. Vì vậy, theo nguyên lý nhân tổng cộng có

$$C_{52}^5.C_{47}^5.C_{42}^5.C_{37}^5$$

cách chia cho 4 người mỗi người một xấp 5 quân bài.

Thí dụ trên là một bài toán điển hình về việc phân bố các đồ vật khác nhau vào các hộp khác nhau. Các đồ vật là 52 quân bài, còn 4 hộp là 4 người chơi và số còn lại để trên bàn. Số cách sắp xếp các đồ vật vào trong hộp được cho bởi mệnh đề sau

Mệnh đề 3.7. *Số cách phân chia n đồ vật khác nhau vào trong k hộp khác nhau sao cho có n_i vật được đặt vào trong hộp thứ i, với $i = 1, 2, ..., k$ bằng*

$$\frac{n!}{n_1!n_2!....n_k!.(n - n_1 - ... - n_k)!}$$

3.6 Sinh các hoán vị và tổ hợp

3.6.1. Sinh các hoán vị

Có nhiều thuật toán đã được phát triển để sinh ra $n!$ hoán vị của tập $\{1, 2, ..., n\}$. Ta sẽ mô tả một trong các phương pháp đó, phương pháp liệt kê các hoán vị của tập $\{1, 2, ..., n\}$ theo thứ tự từ điển. Khi đó, hoán vị $a_1a_2...a_n$ được gọi là đi trước hoán vị $b_1b_2...b_n$ nếu tồn tại $k(1 \le k \le n), a_1 = b_1, a_2 = b_2, ..., a_{k-1} = b_{k-1}$ và $a_k < b_k$.

Thuật toán sinh các hoán vị của tập $\{1, 2, ..., n\}$ dựa trên thủ tục xây dựng hoán vị kế tiếp, theo thứ tự từ điển, từ hoán vị cho trước $a_1a_2...a_n$. Đầu tiên nếu $a_{n-1} < a_n$ thì rõ ràng đổi chỗ a_{n-1} và a_n cho nhau thì sẽ nhận được hoán vị mới đi liền sau hoán vị đã cho.

Nếu tồn tại các số nguyên a_j và a_{j+1} sao cho $aj < a_{j+1}$ và $a_{j+1} > a_{j+2} > ... > a_n$, tức là tìm cặp số nguyên liền kề đầu tiên tính từ bên phải sang bên trái của hoán vị mà số đầu nhỏ hơn số sau. Sau đó, để nhận được hoán vị liền sau ta đặt vào vị trí thứ j số nguyên nhỏ nhất trong các số lớn hơn a_j của tập $a_{j+1}, a_{j+2}, ..., a_n$, rồi liệt kê theo thứ tự tăng dần của các số còn lại của $a_j, a_{j+1}, a_{j+2}, ..., a_n$ vào các vị trí $j+1, ..., n$. Dễ thấy không có hoán vị nào đi sau hoán vị xuất phát và đi trước hoán vị vừa tạo ra.

Ví dụ 3.54. Tìm hoán vị liền sau theo thứ tự từ điển của hoán vị 4736521.

Lời giải. Cặp số nguyên đầu tiên tính từ phải qua trái có số trước nhỏ hơn số sau là $a_3 = 3$ và $a_4 = 6$. Số nhỏ nhất trong các số bên phải của số 3 mà lại lớn hơn 3 là số 5. Đặt số 5 vào vị trí thứ 3. Sau đó đặt các số 3, 6, 1, 2 theo thứ tự tăng dần vào bốn vị trí còn lại. Hoán vị liền sau hoán vị đã cho là 4751236.

Thuật toán 3.1. Hoán vị liền sau

procedure hoan_vi_lien_sau $(a_1, a_2, ..., a_n)$
//*(hoán vị của* $\{1, 2, ..., n\}$ *khác* $\{n, n-1, ..., 2, 1\}$*)*
  $j := n - 1$
  while $a_j > a_{j+1}$
    $j := j - 1$
//*j là chỉ số lớn nhất mà* $a_j < a_{j+1}$
  $k := n$
   while $a_j > a_k$
     $k := k - 1$
//*a_k là số nguyên nhỏ nhất trong các số lớn hơn a_j và bên phải a_j*
   doi_cho (a_j, a_k)
   $r := n$
   $s := j + 1$
   while $r > s$
     doi_cho (a_r, a_s)
    $r := r - 1; s := s + 1$
//*Điều này sẽ xếp phần đuôi của hoán vị ở sau vị trí thứ j theo thứ tự tăng dần.*

3.6.2. Sinh các tổ hợp

Làm thế nào để tạo ra tất cả các tổ hợp các phần tử của một tập hữu hạn? Vì tổ hợp chính là một tập con, nên ta có thể dùng phép tương ứng 1-1 giữa các tập con của $\{a_1, a_2, ..., a_n\}$ và xâu nhị phân độ dài n.

Ta thấy một xâu nhị phân độ dài n cũng là khai triển nhị phân của một số nguyên nằm giữa 0 và $2^n - 1$. Khi đó 2^n xâu nhị phân có thể liệt kê theo thứ tự tăng dần của số nguyên trong biểu diễn nhị phân của chúng. Chúng ta sẽ bắt đầu từ xâu nhị phân nhỏ nhất 00...00 (n số 0). Mỗi bước để tìm xâu liền sau ta tìm vị trí đầu tiên tính từ phải qua trái mà ở đó là số 0, sau đó thay tất cả số 1 ở bên phải số này bằng 0 và đặt số 1 vào chính vị trí này.

Thuật toán 3.2. Xâu nhị phân liền sau

```
procedure xau_nhi_phan_lien_sau (b_{n-1}b_{n-2}...b_1b_0)
// xâu nhị phân khác (11...11)
    i := 0
    while b_i = 1
    begin
        b_i := 0
        i := i + 1
    end
    b_i := 1
```

Tiếp theo chúng ta sẽ trình bày thuật toán tạo các tổ hợp chập k từ n phần tử $\{1, 2, ..., n\}$. Mỗi tổ hợp chập k có thể biểu diễn bằng một xâu tăng. Khi đó có thể liệt kê các tổ hợp theo thứ tự từ điển. Có thể xây dựng tổ hợp liền sau tổ hợp $a_1a_2...a_k$ bằng cách sau. Trước hết, tìm phần tử đầu tiên a_i trong dãy đã cho kể từ phải qua trái sao cho $a_i \neq n - k + i$. Sau đó thay a_i bằng $a_i + 1$ và a_j bằng $a_i + j - i + 1$ với $j = i + 1, i + 2, ..., k$.

Ví dụ 3.55. Tìm tổ hợp chập 4 từ tập $\{1, 2, 3, 4, 5, 6\}$ đi liền sau tổ hợp $\{1, 2, 5, 6\}$.

Lời giải. Ta thấy từ phải qua trái $a_2 = 2$ là số hạng đầu tiên của tổ hợp đã cho thỏa mãn điều kiện $a_i \neq 6 - 4 + i$. Để nhận được tổ hợp tiếp sau ta tăng a_i lên một đơn vị, tức $a_2 = 3$, sau đó đặt $a_3 = 3 + 1 = 4$ và $a_4 = 3 + 2 = 5$. Vậy tổ hợp liền sau tổ hợp đã cho là $\{1, 3, 4, 5\}$. Thủ tục này được cho dưới dạng thuật toán như sau.

Thuật toán 3.3. Tổ hợp liền sau

procedure to_hop_lien_sau ($\{a_1, a_2, ..., a_k\}$: tập con thực sự của tập $\{1, 2, ..., n\}$ không bằng $\{n - k + 1, ..., n\}$ với $a_1 < a_2 < ... < a_k$)

$i := k$

while $a_i = n - k + i$

  $i := i - 1$

  $a_i := a_i + 1$

for $j := i + 1$ **to** k

  $a_j := a_i + j - i$

3.7 Hệ thức truy hồi

3.7.1. Khái niệm mở đầu hệ thức truy hồi

Đôi khi ta rất khó định nghĩa một đối tượng một cách tường minh. Nhưng có thể dễ dàng định nghĩa đối tượng này qua chính nó. Kỹ thuật này được gọi là đệ quy. Định nghĩa đệ quy của một dãy số định rõ giá trị của một hay nhiều hơn các số hạng đầu tiên và quy tắc xác định các số hạng tiếp theo từ các số hạng đi trước. Định nghĩa đệ quy có thể dùng để giải các bài toán đếm. Khi đó quy tắc tìm các số hạng từ các số hạng đi trước được gọi là các hệ thức truy hồi.

Định nghĩa 3.12. *Hệ thức truy hồi* (hay công thức truy hồi) đối với dãy số $\{a_n\}$ là công thức biểu diễn a_n qua một hay nhiều số hạng đi trước của dãy.

Dãy số được gọi là lời giải hay *nghiệm của hệ thức truy hồi* nếu các số hạng của nó thỏa mãn hệ thức truy hồi này.

Ví dụ 3.56 (Lãi kép)**.** Giả sử một người gửi 10.000 đô la vào tài khoản của mình tại một ngân hàng với lãi suất kép 11% mỗi năm.

Sau 30 năm anh ta có bao nhiêu tiền trong tài khoản của mình?

Lời giải. Gọi T_n là tổng số tiền có trong tài khoản sau n năm. Vì số tiền có trong tài khoản sau n năm bằng số có sau $n-1$ năm cộng lãi suất của năm thứ n, nên ta thấy dãy $\{T_n\}$ thoả mãn hệ thức truy hồi sau:

$$T_n = T_{n-1} + 0,11T_{n-1} = (1,11)T_{n-1}$$

với điều kiện đầu $T_0 = 10.000$ đô la. Từ đó suy ra $T_n = (1,11)^n.10.000$. Thay $n = 30$ cho ta $T_{30} = 228922,97$ đô la.

Ví dụ 3.57. Tìm hệ thức truy hồi và cho điều kiện đầu để tính số các xâu nhị phân độ dài n và không có hai số 0 liên tiếp. Có bao nhiêu xâu nhị phân như thế có độ dài bằng 5?

Lời giải. Gọi a_n là số các xâu nhị phân độ dài n và không có hai số 0 liên tiếp. Để nhận được hệ thức truy hồi cho $\{a_n\}$, ta thấy rằng theo quy tắc cộng, số các xâu nhị phân độ dài n và không có hai số 0 liên tiếp bằng số các xâu nhị phân như thế kết thúc bằng số 1 cộng với số các xâu như thế kết thúc bằng số 0. Giả sử $n \geq 3$.

Các xâu nhị phân độ dài n, không có hai số 0 liên tiếp kết thúc bằng số 1 chính là xâu nhị phân như thế, độ dài $n-1$ và thêm số 1 vào cuối của chúng. Vậy chúng có tất cả là a_{n-1} . Các xâu nhị phân độ dài n, không có hai số 0 liên tiếp và kết thúc bằng số 0, cần phải có bit thứ $n-1$ bằng 1, nếu không thì chúng có hai số 0 ở hai bit cuối cùng. Trong trường hợp này chúng có tất cả là a_{n-2} . Cuối cùng ta có được

$$a_n = a_{n-1} + a_{n-2} \text{ với } n \geq 3.$$

Điều kiện đầu là $a_1 = 2$ và $a_2 = 3$. Khi đó
$a_5 = a_4 + a_3 = a_3 + a_2 + a_3 = 2(a_2 + a_1) + a_2 = 13.$

3.7.2. Giải các hệ thức truy hồi

Định nghĩa 3.13. Một *hệ thức truy hồi tuyến tính thuần nhất bậc k* với hệ số hằng số là hệ thức truy hồi có dạng

$$a_n = c_1a_{n-1} + c_2a_{n-2} + ... + c_ka_{n-k},$$

trong đó $c_1, c_2, ..., c_k$ là các số thực và $c_k \neq 0$.

Theo nguyên lý của quy nạp toán học thì dãy số thỏa mãn hệ thức truy hồi nêu trong định nghĩa được xác định duy nhất bằng hệ thức truy hồi này và k điều kiện đầu:

$$a_0 = C_0, a_1 = C_1, ..., a_{k-1} = C_{k-1}.$$

Phương pháp cơ bản để giải hệ thức truy hồi tuyến tính thuần nhất là tìm nghiệm dưới dạng $a_n = r^n$, trong đó r là hằng số. Chú ý rằng $a_n = r^n$ là nghiệm của hệ thức truy hồi $a_n = c_1 a_{n-1} + c_2 a_{n-2} + ... + c_k a_{n-k}$ nếu và chỉ nếu

$$r^n = c_1 r^{n-1} + c_2 r^{n-2} + ... + c_k r^{n-k}$$

hay

$$r^k - c_1 r^{k-1} - c_2 r^{k-2} - ... - c_{k-1} r - c_k = 0.$$

Phương trình này được gọi là phương trình đặc trưng của hệ thức truy hồi, nghiệm của nó gọi là nghiệm đặc trưng của hệ thức truy hồi.

Mệnh đề 3.8. *Cho $c_1, c_2, ..., c_k$ là các số thực. Giả sử rằng phương trình đặc trưng*

$$r^k - c_1 r^{k-1} - c_2 r^{k-2} - ... - c_{k-1} r - c_k = 0.$$

có k nghiệm phân biệt $r_1, r_2, ..., r_k$. Khi đó dãy $\{a_n\}$ là nghiệm của hệ thức truy hồi $a_n = c_1 a_{n-1} + c_2 a_{n-2} + ... + c_k a_{n-k}$ nếu và chỉ nếu $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + ... + \alpha_k r_k^n$, với $n = 1, 2, ...$ trong đó $\alpha_1, \alpha_2, ..., \alpha_k$ là các hằng số.

Ví dụ 3.58. Tìm công thức hiển của các số Fibonacci.

Lời giải. Dãy các số Fibonacci thỏa mãn hệ thức $f_n = f_{n-1} + f_{n-2}$ và các điều kiện đầu $f_0 = 0$ và $f_1 = 1$. Các nghiệm đặc trưng là $r_1 = \dfrac{1+\sqrt{5}}{2}$ và $r_2 = \dfrac{1-\sqrt{5}}{2}$

Do đó các số Fibonacci được cho bởi công thức

$$f_n = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Các điều kiện ban đầu $f0 = 0 = \alpha_1 + \alpha_2$ và $f_1 = 1 = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right) + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right)$. Từ hai phương trình này cho ta $\alpha_1 = \frac{1}{\sqrt{5}}, \alpha_2 = -\frac{1}{\sqrt{5}}$. Do đó các số Fibonacci được cho bởi công thức hiển sau

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Ví dụ 3.59. Hãy tìm nghiệm của hệ thức truy hồi $a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$ với điều kiện ban đầu $a_0 = 2, a_1 = 5$ và $a_2 = 15$.

Lời giải. Đa thức đặc trưng của hệ thức truy hồi này là $r^3 - 6r^2 + 11r - 6$. Các nghiệm đặc trưng là $r = 1, r = 2, r = 3$. Do vậy nghiệm của hệ thức truy hồi có dạng

$$a_n = \alpha_1 1^n + \alpha_2 2^n + \alpha_3 3^n.$$

Các điều kiện ban đầu

$$\begin{aligned} a_0 &= 2 = \alpha_1 + \alpha_2 + \alpha_3 \\ a_1 &= 5 = \alpha_1 + \alpha_2 2 + \alpha_3 3 \\ a_1 &= 15 = \alpha_1 + \alpha_2 4 + \alpha_3 9. \end{aligned}$$

Giải hệ các phương trình này ta nhận được $\alpha_1 = 1, \alpha_2 = -1, \alpha_3 = 2$. Vì thế, nghiệm duy nhất của hệ thức truy hồi này và các điều kiện ban đầu đã cho là dãy $\{a_n\}$ với

$$a_n = 1 - 2^n + 2.3^n.$$

3.8 Quan hệ chia để trị

3.8.1. Giới thiệu

Nhiều thuật toán đệ quy chia bài toán với các thông tin vào đã cho thành một hay nhiều bài toán nhỏ hơn. Sự phân chia này được áp dụng liên tiếp cho tới khi có thể tìm được lời giải của bài toán nhỏ một cách dễ dàng. Chẳng hạn, ta tiến hành việc tìm kiếm nhị phân bằng cách rút gọn việc tìm kiếm một phần tử trong một danh sách tới việc tìm phần tử đó trong một danh sách có độ dài giảm đi một nửa. Ta rút gọn liên tiếp như vậy cho tới khi còn lại một phần tử. Một ví dụ khác là thủ tục nhân các số nguyên. Thủ tục này rút gọn bài toán nhân hai số nguyên tới ba phép nhân hai số nguyên với số bit giảm đi một nửa. Phép rút gọn này được dùng liên tiếp cho tới khi nhận được các số nguyên có một bit. Các thủ tục này gọi là các thuật toán chia để trị.

3.8.2. Hệ thức chia để trị

Giả sử rằng một thuật toán phân chia một bài toán cỡ $n$thành a bài toán nhỏ, trong đó mỗi bài toán nhỏ có cỡ $\frac{n}{b}$ (để đơn giản giả sử rằng n chia hết cho b; trong thực tế các bài toán nhỏ thường có cỡ $\lfloor\frac{n}{b}\rfloor$ hoặc $\lceil\frac{n}{b}\rceil$). Giả sử rằng tổng các phép toán thêm vào khi thực hiện phân chia bài toán cỡ n thành các bài toán có cỡ nhỏ hơn là $g(n)$. Khi đó, nếu $f(n)$ là số các phép toán cần thiết để giải bài toán đã cho thì f thỏa mãn hệ thức truy hồi sau

$$f(n) = a.f\left(\frac{n}{b}\right) + g(n)$$

Hệ thức này có tên là hệ thức truy hồi chia để trị.

Ví dụ 3.60. Thuật toán tìm kiếm nhị phân đưa bài toán tìm kiếm cỡ n về bài toán tìm kiếm phần tử này trong dãy tìm kiếm cỡ $\frac{n}{2}$, khi n chẵn. Khi thực hiện việc rút gọn cần hai phép so sánh. Vì thế, nếu $f(n)$ là số phép so sánh cần phải làm khi tìm kiếm một phần tử trong danh sách tìm kiếm cỡ n ta có $f(n) = f(\frac{n}{2}) + 2$, nếu n là số chẵn.

Ví dụ 3.61. Có các thuật toán hiệu quả hơn thuật toán thông thường để nhân hai số nguyên. Ở đây ta sẽ có một trong các thuật toán như vậy. Đó là thuật toán phân nhanh, có dùng kỹ thuật chia để trị. Trước tiên ta phân chia mỗi một trong hai số nguyên $2n$ bit thành hai khối mỗi khối n bit. Sau đó phép nhân hai số nguyên $2n$ bit ban đầu được thu về ba phép nhân các số nguyên n bit cộng với các phép dịch chuyển và các phép cộng.

Giả sửa và b là các số nguyên có các biểu diễn nhị phân độ dài $2n$ là $a = (a_{2n-1}a_{2n-2}...a_1a_0)_2$ và $b = (b_{2n-1}b_{2n-2}...b_1b_0)_2$.

Giả sử $a = 2^nA_1 + A_0, b = 2^nB_1 + B_0$, trong đó

$$A_1 = (a_{2n-1}a_{2n-2}...a_{n+1}a_n)_2, A_0 = (a_{n-1}...a_1a_0)_2$$
$$B_1 = (b_{2n-1}b_{2n-2}...b_{n+1}b_n)_2, B_0 = (b_{n-1}...b_1b_0)_2.$$

Thuật toán nhân nhánh các số nguyên dựa trên đẳng thức:

$$ab = (2^{2n} + 2^n)A_1B_1 + 2^n(A_1 - A_0)(B_0 - B_1) + (2^n + 1)A_0B_0.$$

Đẳng thức này chỉ ra rằng phép nhân hai số nguyên 2n bit có thể thực hiện bằng cách dùng ba phép nhân các số nguyên n bit và các phép cộng, trừ và phép dịch chuyển. Điều đó có nghĩa là nếu $f(n)$ là tổng các phép toán nhị phân cần thiết để nhân hai số nguyên n bit thì

$$f(2n) = 3f(n) + C_n.$$

Ba phép nhân các số nguyên n bit cần $3f(n)$ phép toán nhị phân. Mỗi một trong các phép cộng, trừ hay dịch chuyển dùng một hằng số nhân với n lần các phép toán nhị phân và C_n là tổng các phép toán nhị phân được dùng khi làm các phép toán này.

Mệnh đề 3.9. *Giả sử f là một hàm tăng thoả mãn hệ thức truy hồi $f(n) = a.f\left(\frac{n}{b}\right) + c$ với mọi n chia hết cho $b, a \geq 1, b$ là số nguyên lớn hơn 1, còn c là số thực dương. Khi đó*

$$f(n) = \begin{cases} O(n^{\log_b a}), & a > 1 \\ O(\log n), & a = 1. \end{cases}$$

Mệnh đề 3.10. *Giả sử f là hàm tăng thoả mãn hệ thức truy hồi $f(n) = a.f\left(\frac{n}{b}\right) + cn^d$ với mọi $n = b^k$, trong đó k là số nguyên dương, $a \geq 1, b$ là số nguyên lớn hơn 1, còn c và d là các số thực dương. Khi đó*

$$f(n) = \begin{cases} O(n^{\log_b a}), & a > b^d \\ O(n^d \log n), & a = b^d. \\ O(n^d), & a < b^d. \end{cases}$$

Ví dụ 3.62. Hãy ước lượng số phép toán nhị phân cần dùng khi nhân hai số nguyên n bit bằng thuật toán nhân nhanh.

Lời giải. Ví dụ trước đã chỉ ra rằng $f(n) = 3f(n/2) + C_n$, khi n chẵn. Vì thế, từ Mệnh đề 3.10 ta suy ra $f(n) = O(n^{\log_2 3})$. Chú ý là $\log_2 3 \approx 1,6$. Vì thuật toán nhân thông thường dùng $O(n^2)$ phép toán nhị phân, thuật toán nhân nhanh sẽ thực sự tốt hơn thuật toán nhân thông thường khi các số nguyên là đủ lớn.

3.9 Bài tập

▷ **3.1.** Có 100 vé đánh số từ 1 đến 100 được bán cho 100 người khác nhau. Người ta sẽ trao 4 giải thưởng kể cả giải độc đắc. Hỏi

a) Có bao nhiêu cách trao thưởng?

b) Có bao nhiêu cách trao thưởng nếu người giữ vé 47 trúng giải độc đắc?

▷ **3.2.** Một câu lạc bộ có 25 thành viên.

a) Có bao nhiêu cách chọn 4 thành viên vào ủy ban thường trực?

b) Có bao nhiêu cách chọn Chủ tích, Phó chủ tích, Thư ký và Thủ quỹ?

▷ **3.3.** Giả sử một tổ bộ môn có 10 nam và 15 nữ. Có bao nhiêu cách chọn một hội đồng gồm 6 ủy viên trong đó ủy viên nam bằng số ủy viên nữ.

▷ **3.4.** Tìm hệ số của $x^{101}y^{99}$ trong khai triển của $(2x - 3y)^{200}$.

▷ **3.5.** Để chuẩn bị cho giai đoạn 2, coa 150 sinh viên ghi tên học môn Lôgic toán; 120 sinh viên ghi tên học môn Lý thuyết đồ thị và 200 sinh viên ghi tên học môn Văn phạm và Ôtômat. Hỏi có bao nhiêu sinh viên ghi tên học một trong ba môn, biết rằng không có sinh viên nào ghi tên học đồng thời 2 môn hoặc cả 3 môn.

▷ **3.6.** Ghi nhãn cho chiếc ghế trong hội trường bằng một chữ cái và một số nguyên dương không vượt quá 100. Hỏi có bao nhiêu chiếc ghế được ghi nhãn?

▷ **3.7.** Trong một trung tâm máy tính có 50 máy tinh. Mỗi máy có 24 cổng. Hỏi có bao nhiêu cổng khác nhau trong trung tâm này.

▷ **3.8.** Có bao nhiêu xâu nhị phân độ dài:

a) Bằng n?

b) Nhỏ hơn hoặc bằng n?

▷ **3.9.** Một dãy XXXYYY độ dài 6. X có thể gán bởi một chữ cái, Y có thể gán một chữ số. Có bao nhiêu dãy được thành lập theo cách trên?

▷ **3.10.** Có thể tạo bao nhiêu hàm số từ tập A có m phần tử vào tập B có n phần tử?

▷ **3.11.** Trong tương lai số điện thoại có 10 chữ số, trong đó 3 chữ số đầu NXX là mã vùng, ba chữ số tiếp theo là mã chi nhánh có dạng NXX, 4 chữ còn lại là XXXX là mã máy. Biết N có thể nhẫn chữ số từ 2 đến 9, còn X nhận các số từ 0 đến 9. Hỏi có bao nhiêu số điện thoại khác nhau theo cách trên?

▷ **3.12.** Mỗi người sử dụng máy tính đều có mật khẩu dài từ 6 đến 8 ký tự. Trong đó mỗi ký tự là một chữ hoa hay một chữ số. Mỗi mật khẩu phải chứa ít nhất một chữ số. Hỏi có bao nhiêu mật khẩu?

▷ **3.13.** Giả sử trong Kho Công nghệ có 1807 sinh viên năm thứ nhất, trong số này có 453 sinh viên chọn môn Tin học; 567 chọn môn toán và 299 chọn cả hai môn Toán và Tin. Hỏi có bao nhiêu sinh viên không học Toán cũng không học Tin học?

▷ **3.14.** Tập $A \cup B$ có bao nhiêu phần tử nếu A có 12 phần tử, B có 18 phần tử và

a) $A \cap B = \emptyset$?

b) $|A \cap B| = 1$?

c) $|A \cap B| = 6$?

d) $A \subset B$?

▷ **3.15.** Tìm số phần tử của tệp $A \cup B \cup C$ nếu mỗi tập có 100 phần tử và nếu

a) Các tập hợp là từng cặp rời nhau;

b) Có 50 phần tử chung của mỗi cặp tập hợp và không có phần tử chung của ba tập hợp;

c) Có 50 phần tử chung của mỗi cặp tập hợp và 25 phần tử chung của ba tập hợp;

▷ **3.16.** Giả sử Khoa Toán - Cơ -Tin học có tổng số sinh viên. Trong đó có 1876 sinh viên học Basic; 999 sinh viên học Java; 345 sinh viên học C. Ngoài ra, 876 sinh viên học Basic và Java; 232 sinh viên học Java và C; 290 sinh viên học Basic và C. Nếu có 189 sinh viên học cả ba môn thì trong khoa có bao nhiêu sinh viên không học môn nào trong cả ba môn kể trên?

▷ **3.17.** Sau cuộc phỏng vấn sinh viên tại nhà ăn ĐHKHTN, người ta thấy 64 sinh viên thích ăn cải xanh; 94 sinh viên thích ăn bắp cải; 58 sinh viên thích ăn súp lơ; 26 sinh viên thích ăn cải xanh và bắp cải; 28 sinh viên thích ăn cải xanh và sup lơ; 22 sinh viên thích ăn bắp cải và sup lơ; 11 sinh viên thích ăn tất cả các loại. Hỏi trong số 270 sinh viên này có bao nhiêu sinh viên không thích ăn cả ba loại rau trên?

▷ **3.18.** Một cuộc họp gồm 12 người tham dự để bàn về 3 vấn đề. Có 8 người phát biểu về vấn đề I, 5 người phát biểu về vấn đề II và 7 người phát biểu về vấn đề III. Ngoài ra, có đúng 1 người không phát biểu vấn đề nào. Hỏi nhiều lắm là có bao nhiêu người phát biểu cả 3 vấn đề.

▷ **3.19.** Chỉ ra rằng có ít nhất 4 người trong số 25 triệu người có cùng tên họ viết tắt bằng 3 chữ cái sinh cùng ngày trong năm (không nhất thiết trong cùng một năm).

▷ **3.20.** Một tay đô vật tham gia thi đấu giành chức vô địch trong 75 giờ. Mỗi giờ anh ta có ít nhất một trận đấu, nhưng toàn bộ anh ta có không quá 125 trận. Chứng tỏ rằng có những giờ liên tiếp anh ta đã đấu đúng 24 trận.

▷ **3.21.** Cho n là số nguyên dương bất kỳ. Chứng minh rằng luôn lấy ra được từ n số đã cho một số số hạng thích hợp sao cho tổng của chúng chia hết cho n.

▷ **3.22.** Trong một cuộc lấy ý kiến về 7 vấn đề, người được hỏi ghi vào một phiếu trả lời sẵn bằng cách để nguyên hoặc phủ định các câu trả lời tương ứng với 7 vấn đề đã nêu. Chứng minh rằng với 1153 người được hỏi luôn tìm được 10 người trả lời giống hệt nhau.

▷ **3.23.** Có 17 nhà bác học viết thư cho nhau trao đổi 3 vấn đề. Chứng minh rằng luôn tìm được 3 người cùng trao đổi một vấn đề.

▷ **3.24.** 10. Có bao nhiêu xâu khác nhau có thể lập được từ các chữ cái trong từ MISSISSIPI, yêu cầu phải dùng tất cả các chữ?

▷ **3.25.** Một giáo sư cất bộ sưu tập gồm 40 số báo toán học vào 4 chiếc ngăn tủ, mỗi ngăn đựng 10 số. Có bao nhiêu cách có thể cất các tờ báo vào các ngăn nếu

1) Mỗi ngăn được đánh số sao cho có thể phân biệt được;

2) Các ngăn là giống hệt nhau?

▷ **3.26.** 1) Tìm hệ thức truy hồi mà R_n thoả mãn, trong đó R_n là số miền của mặt phẳng bị phân chia bởi n đường thẳng nếu không có hai đường nào song song và không có 3 đường nào cùng đi qua một điểm.

2) Tính R_n bằng phương pháp lặp.

▷ **3.27.** 16. Tìm nghiệm của hệ thức truy hồi $a_n = 2a_{n-1} + 5a_{n-2} - 6a_{n-3}$ với $a_0 = 7, a_1 = -4, a_2 = 8$.

Chương 4

QUAN HỆ

4.1 Quan hệ hai ngôi

4.1.1. Định nghĩa quan hệ và ví dụ

Giữa các phần tử trong một tập hợp nào đó mà chúng ta đang quan tâm thường có những mối liên hệ hay những quan hệ. Ví dụ quan hệ lớn hơn giữa các số thực, quan hệ "anh em" giữa người với người, quan hệ đồng dạng giữa các tam giác, v.v.... Mỗi quan hệ trong một tập hợp được đặc trưng bằng một hay một số tiêu chuẩn nào đó thể hiện ngữ nghĩa của quan hệ. Ở đây chúng ta chỉ đề cập đến những quan hệ, được gọi là những quan hệ 2 ngôi, nói lên sự liên hệ giữa mỗi phần tử với các phần tử khác trong tập hợp. Khi ta đang xem xét một quan hệ như thế, thì với hai phần tử x, y tùy ý trong tập hợp chúng sẽ có hoặc là x có quan hệ với y, hoặc là x không có quan hệ với y. Nói như vậy cũng có nghĩa là tập hợp các cặp (x, y) gồm 2 phần tử có quan hệ có thể xác định được quan hệ đang xét trên tập hợp. Về mặt toán học, một quan hệ 2 ngôi được định nghĩa như sau.

Định nghĩa 4.1. Cho một tập hợp X khác rỗng. Một *quan hệ 2 ngôi* trên X là một tập hợp con R của X^2. Cho 2 phần tử x và y của X, ta nói x có quan hệ R với y khi và chỉ khi $(x, y) \in R$, và viết là xRy.

Như vậy,

$$xRy \Leftrightarrow (x, y) \in R.$$

Khi x không có quan hệ R với y, ta viết: $x\overline{R}y$.

Ví dụ 4.1.

1. Trên tập hợp $X = \{1, 2, 3, 4\}$, xét quan hệ 2 ngôi R được định nghĩa bởi
   $R = \{(1, 1), (1, 3), (2, 2), (2, 4), (3, 1), (3, 3), (4, 2), (4, 4)\}$
   Với quan hệ này ta có: $2R4$, nhưng $2\overline{R}3$.
2. Trên tập hợp các số nguyên Z ta định nghĩa một quan hệ 2 ngôi R như sau
   xRy nếu và chỉ nếu $x - y$ là số chẵn.
   hay nói cách khác
   $R = \{(x, y) \in Z^2 | x - y = 2k \text{ với } k \in Z\}$
   Quan hệ R này chính là quan hệ đồng dư modulo 2.

Ví dụ 4.2.

1. Cho n là một số nguyên dương. Nhắc lại rằng quan hệ đồng dư modulo n trên tập hợp các số nguyên Z, ký hiệu bởi $=$ (mod n), được định nghĩa như sau

$$a = b \pmod{n} \Leftrightarrow \exists k \in \mathrm{Z} : (a - b) = k.n$$

   Quan hệ này là một quan hệ 2 ngôi trên Z.

2. Quan hệ $\leq$ trên tập hợp các số thực R cũng là một quan hệ 2 ngôi.

Ví dụ 4.3. Cho E là một tập hợp, đặt $X = P(E)$. Mỗi phần tử thuộc X là một tập hợp con của E. Trên E có các quan hệ quen thuộc sau đây

- quan hệ bao hàm, ký hiệu bởi $\subset$;

- quan hệ chứa, ký hiệu bởi $\supset$;

- quan hệ bằng nhau, ký hiệu bởi $=$

Chú ý.

1. Người ta còn định nghĩa một quan hệ (2 ngôi) giữa một tập hợp A và một tập hợp B là một tập hợp con của $A \times B$. Ví dụ: $A = \{1,2,3,4,5\}, B = \{0,1\}$. Ta có $R = \{(1,1),(2,0),(3,1),(4,0),(5,0)\}$ là một quan hệ giữa A và B.

2. Tổng quát hơn, ta có thể định nghĩa một *quan hệ* giữa các tập hợp $A_1, A_2, ..., A_n$ là một tập hợp con của $A_1 \times A_2 \times ... \times A_n$ (tích Descartes của các tập hợp $A_1, A_2, ..., A_n$). Như vậy, khi R là một quan hệ giữa các tập $A_1, A_2, ..., A_n$ thì mỗi phần tử của R là một bộ $n(a_1, a_2, ..., a_n)$ với $a_i \in A_i (i = 1, \ldots, n)$.

Cách xác định một quan hệ: Dựa vào các phương pháp xác định một tập hợp, ta có thể xác định một quan hệ bằng các phương pháp sau đây

1. Liệt kê: liệt kê tất cả các cặp hay bộ phần tử có quan hệ R (tức là thuộc R). Trong ví dụ 1 ở trên, quan hệ R được cho theo cách liệt kê.

2. Nêu tính chất đặc trưng cho quan hệ R, tức là tính chất hay tiêu chuẩn để xác định các phần tử thuộc R hay không. Trong

các ví dụ 2 và 3 ở trên, quan hệ R được cho bằng cách nêu lên tính chất xác định quan hệ.

4.1.2. Biểu diễn quan hệ

Các phương pháp biểu diễn một quan hệ là theo định nghĩa như tập hợp con của tích Descartes, phương pháp ma trận và phương pháp đồ thị.

a) Phương pháp định nghĩa. Các ví dụ ở phần trước thể hiện phương pháp này. Phương pháp định nghĩa là phương pháp cho quan hệ R một cách trực tiếp hoặc gián tiếp vào tính chất các phần tử của R.

b) Phương pháp ma trận. Ngoài phương pháp biểu diễn một quan hệ 2 ngôi dưới dạng tập hợp các cặp phần tử người ta còn có thể sử dụng ma trận để biểu diễn cho quan hệ trong trường hợp các tập hợp là hữu hạn. Khái niệm ma trận được định nghĩa và khảo sát chi tiết hơn trong "Đại số Tuyến tính". Ở đây chúng ta chỉ cần hiểu ma trận một cách đơn giản là một bảng liệt kê các phần tử thành các dòng và các cột.

Ví dụ, bảng liệt kê 6 số nguyên thành 2 dòng và 3 cột sau đây là một ma trận:

$$M = \begin{pmatrix} 3 & 2 & 5 \\ 0 & 1 & 2 \\ 0 & 5 & 4 \end{pmatrix}$$

Một ma trận M gồm m dòng, n cột sẽ được gọi là một ma trận có cấp $m \times n$. Nếu $m = n$ thì ta nói M là một ma trận vuông cấp n.

Định nghĩa 4.2. Giả sử R là một quan hệ 2 ngôi giữa một tập hợp hữu hạn $A = \{a_1, a_2, ..., a_m\}$ và một tập hữu hạn $B = \{b_1, b_2, ..., b_m\}$. Quan hệ R có thể được biểu diễn bởi ma trận $MR_{m\times n} = (m_{ij})_{m\times n}$ gồm m dòng và n cột (tức là ma trận cấp $m \times n$), trong đó

$$m_{ij} = \begin{cases} 1 & \text{nếu } (a_i, b_j) \in R \\ 0 & \text{nếu } (a_i, b_j) \notin R \end{cases}$$

Ta gọi ma trận MR là *ma trận biểu diễn* của quan hệ R.

Ví dụ 4.4. Với $A = \{1,2,3\}$ và $B = \{a,b,c\}$, thì các quan hệ sau đây:
$R = \{(1,a),(1,b),(1,c)\}$
$S = \{(1,a),(1,b),(1,c),(2,b),(2,c),(3,c)\}$
có các ma trận biểu diễn là

$$MR_{3\times 3} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad MS_{3\times 3} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Trong trường hợp R là một quan hệ 2 ngôi trên một tập X hữu hạn và có n phần tử thì ma trận biểu diễn của R là một ma trận có n dòng và n cột (tức là ma trận vuông cấp n).

b) Phương pháp đồ thị. Cho tập hữu hạn khác trống $A = \{a_1, a_2, ..., a_n\}$.

Định nghĩa 4.3. Một quan hệ $R \subseteq A \times A$ có thể biểu diễn bằng một đồ thị có hướng như sau: Mỗi một $a \in A$ được xem là một đỉnh của đồ thị, do đó số đỉnh của đồ thị bằng số phần tử của tập A. Với $(a,b) \in R$ thì có một cung đi từ đỉnh a tới đỉnh b. Như vậy,

- nếu $a, b \in A$ mà $(a,b) \in R$ hay aRb thì a đến b có một cung.

- Nếu $(a,a) \in R$ thì ta có một khuyên tại đỉnh a.

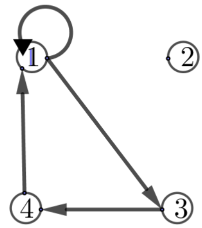
- Nếu $(a,b) \notin R$ thì a đến b không có một cung nào.

Với mỗi một quan hệ $R \subseteq A \times A$ ta có thể biểu diễn bằng đồ thị có hướng. Ngược lại, ứng với mỗi một đồ thị có hướng mà các đỉnh thuộc tập A có thể xác định được một quan hệ R trên A sao cho nó nhận đồ thị đó là dạng biểu diễn của nó.

Ví dụ 4.5. Cho $A = \{1, 2, 3, 4\}$ và quan hệ $R \subseteq A \times A$ với

$$R = \{(1,1), (1,3), (3,4), (4,1)\}$$

Khi đó biểu diễn đồ thị của quan hệ R sẽ là



Ví dụ 4.6. Cho $A = \{1, 2, 3, 4, 5, 6\}$ và quan hệ $R \subseteq A \times A$ được xác định như sau: Với mọi $a, b \in A$, aRb khi và chỉ khi hiệu $a - b$ là một số chẵn. Khi đó ta có thể biểu diễn quan hệ theo cả ba phương pháp như sau:

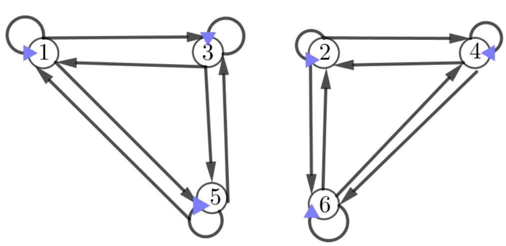
a) Biểu diễn quan hệ R theo định nghĩa:

Quan hệ R là tập hợp $\{(1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (1,3), (3,1), (1,5), (5,1), (2,4), (4,2), (2,6), (6,2), (3,5), (5,3), (4,6), (6,4)\}$

b) Biểu diễn quan hệ R theo ma trận

$$MR_{6\times 6} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

c) Biểu diễn quan hệ R theo đồ thị



4.1.3. Các tính chất của quan hệ

Một quan hệ 2 ngôi trên một tập hợp có thể có một số tính chất nào đó làm cho tập hợp có một cấu trúc nhất định. Dưới đây là định nghĩa một số tính chất thường được xét đối với một quan hệ 2 ngôi.

Định nghĩa 4.4. Giả sử R là một quan hệ 2 ngôi trên một tập hợp X.

- Ta nói quan hệ R có tính *phản xạ* (reflexive) nếu và chỉ nếu xRx với mọi $x \in X$.
- Ta nói quan hệ R có tính *đối xứng* (symmetric) nếu và chỉ nếu $xRy \Rightarrow yRx$ với mọi $x, y \in X$.
- Ta nói quan hệ R có tính *phản xứng* (antisymmetric) nếu và chỉ nếu $(xRy$ và $yRx) \Rightarrow x = y$ với mọi $x, y \in X$.
- Ta nói quan hệ R có tính *truyền* hay *bắc cầu* (transitive) nếu và chỉ nếu $(xRy$ và $yRz) \Rightarrow xRz$ với mọi $x, y, z \in X$.

Ví dụ 4.7. Trong ví dụ này chúng ta đề cập đến một số quan hệ đã được nêu lên trong các ví dụ của mục ở trên, và phát biểu các tính chất của chúng. Việc kiểm chứng các tính chất này khá dễ dàng.

1. Quan hệ đồng dư modulo n trên Z có 3 tính chất: phản xạ, đối xứng, truyền.
2. Quan hệ $\leq$ trên tập hợp các số thực có 3 tính chất: phản xạ, phản xứng, truyền.
3. Cho E là một tập hợp. Quan hệ $\subseteq$ trên $P(E)$ có 3 tính chất: phản xạ, phản xứng, truyền.

Bằng phương pháp biểu diễn quan hệ theo ma trận và đồ thị ta dễ dàng nhận biết được một số tính chất của quan hệ.

- Một quan hệ có tính chất phản xạ khi và chỉ khi ma trận biểu diễn nó có tất cả các phần tử trên đường chéo chính bằng 1.
- Một quan hệ có tính chất phản xạ khi và chỉ khi đồ thị biểu diễn nó tại mỗi đỉnh đều có khuyên.

- Một quan hệ có tính đối xứng khi và chỉ khi ma trận biểu diễn nó là ma trận đối xứng qua đường chéo chính.
- Một quan hệ có tính đối xứng khi và chỉ khi đồ thị biểu diễn nó có tính chất: Với A, B là hai đỉnh bất kỳ và nếu có một cung đi từ A đến B thì cũng có cung đi từ B đến A.

Ví dụ 4.8. Cho $A = \{1, 2, 3, 4\}$. Tập $R \subseteq A \times A$ được định nghĩa như sau Với mọi $a, b \in A$ thì aRb khi và chỉ khi tổng $a + b$ là số lẻ.

1. Biểu diễn R bằng 3 phương pháp đã nêu.
2. R có tính chất gì?
3. Có nhận xét gì về ma trận và đồ thị của R?

Lời giải.

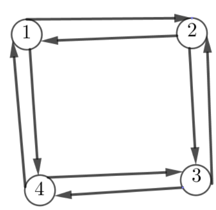
1. Dạng định nghĩa của R là

$$R = \{(1, 2), (1, 4), (2, 3), (3, 4), (2, 1), (4, 1), (3, 2), (4, 3)\}$$

Dạng ma trận của R là

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

Dạng đồ thị là



2. Ta có nhận xét về một số tính chất của quan hệ R như sau:

- Quan hệ R có tính đối xứng, vì $a + b = b + a$ hay $(a, b) \in R$ thì $(b, a) \in R$.
- R không phản xạ vì $(1, 1) \notin R$.

- R không phản đối xứng, vì $(1,2) \in R, (2,1) \in R$ nhưng $2 \neq 1$.
- R không bắc cầu, vì $(1,2) \in R, (2,3) \in R$ nhưng $(1,3) \notin R$.

3. Nhận xét tính chất của R từ ma trận quan hệ R.

Trong ma trận trên phần tử trên đường chéo chính bẳng 0, nên R là không phản xạ. Do các phần tử đối xứng qua đường chéo chính đều bằng nhau nên R có tính đối xứng

Nhận xét tính chất của R từ đồ thị quan hệ R. Vì trong đồ thị tồn tại đỉnh không có khuyên nên quan hệ R là không phản xạ. Từ đồ thị ta thấy cứ có một cung đi từ đỉnh i đến j thì chiều ngược lại cũng có, nên R có tính đối xứng.

Ví dụ 4.9. Cho $A = \{1,2,3,4\}$. Quan hệ $R \subseteq A \times A$ được định nghĩa như sau với mọi $a, b \in A$ thì $aRb \Leftrightarrow a + b = 2k$ (k là số nguyên). Hãy chỉ ra quan hệ R thỏa mãn 3 tính chất: Phản xạ, đối xứng, bắc cầu.

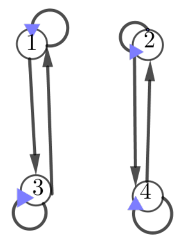
Lời giải. Thật vậy,
- R có tính phản xạ vì với mọi $a \in A$ ta có $a + a = 2a$ là một số chẵn, do đó $(a,a) \in R$.
- R có tính đối xứng, vì với mọi $a, b \in A$ mà $a + b = 2k$ thì $b + a = 2k$ do đó nếu $(a,b) \in R$ thì $(b,a) \in R$.
- R có tính bắc cầu, vì với mọi $a, b, c \in A$ nếu $a + b = 2k, b + c = 2k'$ thì $a + c = (2k - b) + (2k' - b) = 2(k + k') - 2b = 2k''$ với k, k', k'' là số nguyên nào đó. Nghĩa là $(a,b) \in R$ và $(b,c) \in R$ thì $(a,c) \in R$.

Nhận xét về đồ thị và quan hệ R.

- Ma trân của R

$$MR_{4\times4} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

- Đồ thị của R

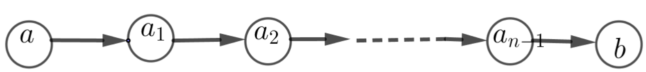


4.2 Cung và đường trong đồ thị quan hệ

4.2.1. Định nghĩa

Giả sử $R \subseteq A \times A$. Nếu $a, b \in A$ mà $(a, b) \in R$ hay aRb thì từ a đến b có một cung hướng đi từ a đến b.

Định nghĩa 4.5. Nếu giữa đỉnh a và b có tồn tại một dãy các đỉnh $a = a_0, a_1, a_2, ..., a_{n-1}a_n = b$ sao cho a_iRa_{i+1} $(i = 1, 2, ..., n-1)$ thì ta nói từ a đến b có một đường đi trong đồ thị quan hệ R.



Độ dài của đường đi là số các cung có mặt trong đường đó. Trong đường trên có độ dài n. Nếu đường đi có độ dài bằng 1 thì đườn đi đó là một cung, hay cung có thể xem như đường đi có độ dài bằng 1.

4.2.2. Tính chất

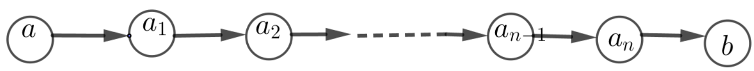
Định lý 4.1. *Cho quan hệ $R \subseteq A \times A$ và R có tính chất bắc cầu. Khi đó nếu trong đồ thị của R có một đường đi độ dài n $(n \geq 1)$ từ đỉnh a tới đỉnh b thì cũng có một cung từ a đến b.*

Chứng minh. chứng minh Quy nạp theo $n \geq 1$.

- Trường hợp $n = 1$. Cung và đường trùng nhau nên định lý đúng.

- Giả sử định lý đúng với n, nghĩa là từ a đến b có một đường đi có độ dài n thì cũng có một cung đi từ a đến b.

- Xét trường hợp bất kỳ từ a đến b có độ dài $n+1$ trong đồ thị của R:



Ở đây $aRa_1, a_1Ra_2, ..., a_{n-1}Ra_n, a_nRb$. Đường đi từ a đến a_n có độ dài n nên theo giả thiết quy nạp thì từ a đếna_n có cung, tức là aRa_n. Do R có tính bắc cầu nên từ aRa_n và a_nRb ta có aRb hay trong đường độ dài $n+1$ từ a đến b coa một cung đi từ a đến b. Điều phải chứng minh.

4.3 Quan hệ ngược và hợp thành

4.3.1. Quan hệ ngược

Định nghĩa 4.6. Giả sử R là quan hệ từ tập A vào tập B. *Quan hệ ngược* của quan hệ R được ký hiệu là R^{-1} đó là quan hệ từ tập B vào tập A được định nghĩa như sau

$$R^{-1} = \{(b, a) : a \in A, b \in B \text{ và } aRb\}$$

Ma trận quan hệ ngược : Giả sử $A = \{a_1, a_2, ..., a_n\}, B = \{b_1, b_2, ..., b_m\}$ Ma trận quan hệ ngược là ma trận cấp $m \times n$, là ma trận m hàng n cột mà phần tử α_{ji} được xác định như sau

$$\begin{aligned}\alpha_{ji} &= \begin{cases} 1 & \text{nếu } b_jR^{-1}a_i \\ 0 & \text{trong trường hợp ngược lại,} \end{cases} \\ &= \begin{cases} 1 & \text{nếu } a_iRb_j \\ 0 & \text{trong trường hợp ngược lại,} \end{cases} \\ &= \delta_{ij} \text{ là phần tử ở hàng } i, \text{ cột } j \text{ trong ma trân của quan hệ } R\ .\end{aligned}$$

Như vậy, nếu quan hệ R có ma trận $MR_{n\times m} = (\delta_{ij})_{m\times n}$ thì ma trận của quan hệ ngược $R^{-1} \subseteq B \times A$ mà ký hiệu là $MR^{-1} = (\alpha_{ji})_{m\times n}$ sẽ nhận được từ $MR_{n\times m}$ bằng cách đổi hàng thành cột, đổi cột thành hàng.

Ví dụ 4.10. Cho $A = \{1,2,3,4\}, B = \{5,6,7\}$. R là quan hệ từ A vào B được cho bởi $R = \{(1,5),(1,7),(2,6),(3,6),(3,5)\}$. Tìm quan hệ ngược và ma trận của nó?

Lời giải. Theo định nghĩa ta có
$R^{-1} = \{(5,1),(7,1),(6,2),(6,3),(5,3)\}$.

$$MR_{4\times 3} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad MR^{-1}_{3\times 4} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Rõ ràng ma trận của quan hệ R^{-1} nhận được từ ma trận của quan hệ R bằng cách chuyển hang thành cột , cột thành hàng.

4.3.2. Quan hệ hợp thành

Định nghĩa 4.7. Giả sử R là quan hệ từ tập A vào tập B, S là quan hệ từ tập B vào tập C. Khi đó một quan hệ Q từ A vào C được định nghĩa

$$Q = \{(a,c) : a \in A, c \in C \text{ sao cho tồn tại } b \in B \text{ mà } aRb \text{ và } bSc\}$$

Q gọi là quan hệ hợp thành giữa hai quan hệ R và S.

Ta ký hiệu quan hệ hợp thành giữa R và S là $Q = R.S$.

Ma trận của quan hệ hợp thành.

Giả sử Q là quan hệ hợp thành từ 2 quan hệ R và S như $Q = R.S$.

Giả sử $A = \{a_1, a_2, ..., a_n\}, B = \{b_1, b_2, ..., b_m\}$ và $C = \{c_1, c_2, ..., c_r\}$.

Ma trận quan hệ $R \subseteq A \times B$ là $MR_{n\times m}$ và phần tử hàng i, cột j là

$$\delta_{ij} = \begin{cases} 1 & \text{nếu } a_i R b_j \\ 0 & \text{trong trường hợp ngược lại.} \end{cases}$$

Ma trận quan hệ $S \subseteq B \times C$ là $MR_{m\times r}$ và phần tử hàng j, cột k là

$$\beta_{jk} = \begin{cases} 1 & \text{nếu } b_j R c_k \\ 0 & \text{trong trường hợp ngược lại.} \end{cases}$$

Ma trận quan hệ $Q \subseteq A \times C$ là $MR_{n\times r}$ và phần tử hàng i, cột k là

$$\gamma_{ik} = \begin{cases} 1 & \text{nếu } a_i R c_k \\ 0 & \text{trong trường hợp ngược lại.} \end{cases}$$

Theo định nghĩa của quan hệ hợp thành

$$\gamma_{ik} = \begin{cases} 1 & \text{nếu tồn tại } b_j \in B \text{ sao cho } a_i R b_j \text{ và } b_j R c_k \\ 0 & \text{trong trường hợp ngược lại.} \end{cases}$$

Từ đó suy ra $\gamma_{ik} = 1$ khi và chỉ khi $\delta_{ij} = \beta_{jk} = 1$. Hay ma trận của quan hệ hợp thành từ R, S là nhân hai ma trận của hai quan hệ trên.

$$MQ_{n\times r} = MR_{n\times m}.MS_{m\times r}.$$

Ví dụ 4.11. Cho $A = \{1,2,3,4,5\}, B = \{6,7,8,9\}$ và $c = \{a,b\}$. Cho $R \subseteq A \times B$ bởi

$$R = \{(1,6),(1,8),(2,7),(2,9),(3,9),(4,6),(4,8),(5,7)\}$$

Cho $S \subseteq B \times C$ bởi

$$S = \{(6,a),(7,a),(8,b)\}.$$

Xác định quan hệ hợp thành $Q = R.S$ và chỉ ra rằng

$$MQ_{5\times 2} = MR_{5\times 4}.MS_{4\times 2}$$

Lời giải. Xét quab hệ $Q \subseteq A \times C$ dựa vào R, S ta có

$$Q = R.S = \{(1,a),(1,b),(2,a),(4,a),(4,b),(5,a)\}.$$

$$MR_{5\times 4} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad MS_{4\times 2} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad MQ_{5\times 2} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

4.4 Quan hệ tương đương

4.4.1. Định nghĩa quan hệ tương đương

Định nghĩa 4.8. Một quan hệ 2 ngôi R trên một tập hợp X được gọi là một quan hệ tương đương nếu và chỉ nếu nó thỏa 3 tính chất:

1) Phản xạ: Với mọi $a \in A$ ta có aRa;

2) đối xứng: Với mọi $a, b \in A$ nếu aRb thì bRa;

3) Bắc cầu: Với mọi $a, b, c \in A$ nếu aRb và bRc thì aRc;

Ví dụ 4.12. Cho $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Trên X ta định nghĩa quan hệ R như sau: Với mọi $a, b \in A$ thì aRb khi và chỉ khi $a + b = 2k$, với k là một số nguyên dương nào đó. Hãy chỉ ra R là quan hệ tương đương?

Lời giải. Ta kiểm tra ba tính chất của quan hệ. Thật vậy,

- Tính phản xạ đúng, vì $\forall a \in A$ ta có $a + a = 2a$ nên aRa.
- Tính đối xứng đúng, vì giả sử $a, b \in A$ và aRb khi đó $a + b = 2k$, do đó $b + a = 2k$ nên bRa.
- Tính bắc cầu đúng, vì giả sử $a, b, c \in A$ và aRb, bRc tức là $a + b = 2k_1$ và $b + c = 2k_2$.

  Xét
  $$\begin{aligned} a + c &= (2k_1 - b) + (2k_2 - b) \\ &= 2(k_1 + k_2) - 2b = 2(k_1 + k_2 - b). \end{aligned}$$

  Từ đó suy ra $a + c = 2k_3$, với k_3 là một số nguyên dương nào đó, nên aRc. Vậy R là quan hệ tương đương.

Ví dụ 4.13. 1. Một ví dụ quan trọng về quan hệ tương đương là quan hệ đồng dư modulo n trên $\mathbb{Z}$. Ta đã biết quan hệ này có 3 tính chất phản xạ, đối xứng, truyền.

2. Quan hệ $\leq$ trên $\mathbb{Z}$ không phải là một quan hệ tương đương vì nó không có tính chất đối xứng.

4.4.2. Lớp tương đương và tập hợp tương đương

Định nghĩa 4.9. Với mỗi phần tử $x \in A$, ta định nghĩa *lớp tương đương* chứa x, ký hiệu $\overline{x}_R$ là tập hợp tất cả những phần tử (thuộc x) có quan hệ R với x:

$$\overline{x}_R = \{y : y \in X \text{ và } yRx\}.$$

Như vậy mỗi lớp tương đương là một tập hợp con của A. Nhiều khi ta biết quan hệ tương đương rồi thì người tha viết $\overline{x}$ thay vì $\overline{x}_R$. Người ta đã chứng minh rằng tập hợp các lớp tương đương của quan hệ tương đương R trên A tạo thành một "phân hoạch" của tập hợp A, tức là sưu tập các lớp tương đương khác nhau cho ta một họ các tập con của A rời nhau đôi một và có phần hội bằng A.

Định nghĩa 4.10. Tập hợp các lớp tương đương của quan hệ tương đương R trên A này (là một tập con của $P(A)$) được gọi là *tập hợp thương* (của quan hệ tương đương R trên A).

Quan hệ đồng dư modulo n trên $\mathbb{Z}$ có tập hợp thương tương ứng, được ký hiệu là $\mathbb{Z}_n$, gồm n phần tử

$$\mathbb{Z}_n = \{\overline{0}, \overline{2}, ..., \overline{n-1}\}$$

trong đó ($k \in \mathbb{Z}$) là tập hợp tất cả những số nguyên đồng dư với k modulo n.

Định lý 4.2. *Cho R là quan hệ tương đương trên A. Ba điều kiện sau là tương đương*

1) aRb;

2) $\overline{a}_R = \overline{b}_R$;

3) $\overline{a}_R \cap \overline{b}_R \neq \emptyset$;

Người ta còn chứng minh được rằng việc xác định một quan hệ tương đương trên một tập hợp A tương đương với việc xác định một phân hoạch của tập hợp A, tức là có một song ánh giữa tập hợp tất cả các quan hệ tương đương trên A và tập hợp tất cả các phân hoạch của tập A.

4.4.3. Phân hoạch tương đương

Định nghĩa 4.11. Cho tập $A \neq \emptyset$ và hữu hạn phần tử. Ta chia tập A thành n tập con $A_1, A_2, ..., A_n$ sao cho $A = \bigcup_{k=1}^{n} A_k$ và $A_i \cap A_j = \emptyset$ với $i \neq j$. Khi đó ta nói các tập $A_1, A_2, ..., A_n$ là một *phân hoạch tương đương* trên tập A.

Ngược lại, ứng với mỗi phân hoạch tương đương $A_1, A_2, ..., A_n$ trên A, ta xác định quan hệ tương đương R trên A sinh ra phân hoạch đó như sau

$$\forall a, b \in A, aRb \Leftrightarrow \exists A_i \text{ sao cho } a, b \in A_i (i \in \{1, 2, ..., n\}).$$

Ta sẽ chỉ ra quan hệ R định nghĩa như trên là quan hệ tương đương trên A và R sinh ra phân hoạch đã cho. Thật vậy, ta sẽ chỉ ra R thỏa mãn ba tính chất:

- Phản xạ: với mọi $a \in A$ có duy nhất A_i để $a \in A_i$ hay aRa.

- Đối xứng: Nếu $a, b \in A$ mà aRb, suy ra có ít nhất A_i sao cho $a, b \in A_i$, $i \in \{1, 2, ..., n\}$, từ đó $b, a \in A$ nên bRa.

- Bắc cầu: Giả sử $a, b, c \in A$ và aRb, bRc, ta cần chỉ ra aRc. Thật vậy, từ aRb suy ra tồn tại tập A_i để $a, b \in A_i$; từ bRc suy ra tồn tại A_j để $b, c \in A_j$. Do tính chất $A_i \cap A_j = \emptyset$ với $i \neq j$ nên suy ra $A_i = A_j$ vì chúng đều chứ b. Điều đó chứng tỏ $a, c \in A_i$, hay aRc.

Ví dụ 4.14. Cho $A = \{2, 3, 4, 5, 6\}$ và phân hoạch của A là $A_1 = \{2, 3\}, A_2 = \{4\}, A_3 = \{5, 6\}$. Quan hệ tương đương R sinh ra phân hoạch trên là
$R = \{(2, 2), (3, 3), (2, 3), (3, 2), (4, 4), (5, 5), (6, 6), (5, 6), (6, 5)\}$.
Hãy chỉ ra R sinh ra phân hoạch trên A.

Lời giải. Để chỉ ra R sinh ra A_1, A_2, A_3 ở trên ta lập ma trận từ quan hệ R

$$MR_{5\times5} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Hàng thứ nhất và thứ hai của ma trận sinh ra $A_1 = \{2, 3\}$.

Hàng thứ ba của ma trận sinh ra $A_2 = \{4\}$.

Hàng thứ tư và thứ năm của ma trận sinh ra $A_3 = \{4, 6\}$.

Ví dụ 4.15. Cho $A = \{1, 2, 3, 4, 5, 6, 7\}$. Trên A ta định nghĩa một quan hệ: $\forall a, b \in A : aRb \Leftrightarrow a - b = 3k$ (với k là một số nguyên). Hãy chỉ ra quan hệ R trên là quan hệ tương đương và tìm phân hoạch tương đương do R sinh ra.

Lời giải. Từ định nghĩa ta có
$R = \{(1,1), (2,2), (3,3), (4,4), (5,5), (6,6), (7,7), (1,4), (4,1),$
$(1,7), (7,1), (2,5), (5,2), (3,6), (6,3), (4,7), (7,4)\}$.

Trước hết chỉ ra R thỏa mãn 3 tính chất

1. Phản xạ: Với $\forall a \in A$ rõ ràng $a - a = 0 = 3.0$ nên aRa.
2. Đối xứng: Giả sử $a, b \in A$ và aRb ta có $a - b = 3k$.

   Dĩ nhiên $b - a = 3.(-k)$, do đó bRa.
3. Bắc cầu: Giả sử $a, b, c \in A$: aRb và bRc. theo định nghĩa ta có $a - b = 3k_1, b - c = 3k_2$.

   Xét $a - c = (3k_1 + b) - (b - 3k_2) = 3(k_1 + k_2) = 3k_3$. Vậy aRc.

Để tìm phân hoạch tương đương trên A do R sinh ra, ta lập ma trận

$$MR_{7\times7} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Các hàng 1,4 và 7 sinh ra tập $A_1 = \{1, 4, 7\}$.

Các hàng 2 và 5 sinh ra tập $A_2 = \{2, 5\}$.

Các hàng 3 và 6 sinh ra tập $A_1 = \{3, 6\}$.

Khi đó A_1, A_2, A_3 sẽ tạo ra một phân hoạch tương đương trên A.

Định lý 4.3. • *Nếu $R \subseteq A \times A$ là một quan hệ tương đương trên $A (A \neq \emptyset)$ thì R sẽ tạo ra một phân hoạch tương đương trên A.*

- *Giả sử $A_1, A_2, ..., A_n$ là một phân hoạch tương đương nào đó trên A. Khi đó sẽ có một quan hệ tương đương $S \subseteq A \times A$ sinh ra phân hoặc tương đương trên.*

Chứng minh. Dựa vào định nghĩa quan hệ tương đương và phân hoạch tương đương trên A.

4.5 Quan hệ thứ tự

4.5.1. Các định nghĩa

Định nghĩa 4.12. Một quan hệ 2 ngôi R trên một tập hợp X (khác rỗng) được gọi là một *quan hệ thứ tự* (hay vắn tắt, là một thứ tự) nếu và chỉ nếu nó có 3 tính chất: *phản xạ, phản xứng, bắc cầu*. Khi đó ta cũng nói tập hợp X là một tập có thứ tự. Nếu có thêm tính chất: với mọi $x, y \in X$ ta có xRy hay yRx thì ta nói R là một *quan hệ thứ tự toàn phần* trên X.

Chú ý. Trong trường hợp trên X có nhiều quan hệ thứ tự thì khi xét đến thứ tự trên X ta phải nói rõ thứ tự nào, và ta thường viết tập hợp X có thứ tự dưới dạng một cặp (X, R); trong đó R là quan hệ thứ tự đang xét trên X.

Với 2 tập hợp có thứ tự X và Y ta có thể định ra một thứ tự trên tích Descartes$X \times Y$ dựa vào các thứ tự trênX và trên Y. Từ đó ta $X \times Y$ trở thành một tập hợp thứ tự (xem phần bài tập).

Ví dụ 4.16. 1. Quan hệ $\leq$ trên tập hợp các số thực $\mathbb{R}$ là một quan hệ thứ tự toàn phần.

2. Cho E là một tập hợp. Quan hệ $\subset$ trên $P(E)$ là một quan hệ thứ tự. Nếu E có nhiều hơn 2 phần tử thì thứ tự này không phải là thứ tự toàn phần. Việc kiểm chứng điều này được dành cho người đọc.

Ví dụ 4.17. Trên tập hợp các số nguyên Z, xét qna hệ "chia hết" hay "ước số của", ký hiệu là $|$, được định nghĩa như sau:

$$a|b \Leftrightarrow \exists k \in \mathbb{Z} : a = k.b$$

Dễ dàng kiểm chứng rằng quan hệ $|$ có 3 tính chất: phản xạ, phản xứng, truyền. Từ đó ta có $(\mathbb{Z}, |)$ là một tập hợp có thứ tự.

Ta có 2 số nguyên 2 và 3, không có quan hệ với nhau theo quan hệ $|$. Do đó $|$ không phải là thứ tự toàn phần trên $\mathbb{Z}$.

Nhận xét. Nếu (X, R) là một tập hợp có thứ tự và $A \subset X$ thì quan hệ thứ R thu hẹp trên tập A, cũng được ký hiệu là R (nếu không gây ra nhầm lẫn), là một quan hệ thứ tự trên A. Nói một cách khác, ta có

(X, R) thứ tự và $A \subset X \Rightarrow (A, R)$ thứ tự.

Đối với một tập hợp có thứ tự thì việc đề cập đến các khái niệm như "phần tử nhỏ nhất", "phần tử lớn nhất", ... là điều rất tự nhiên. Dưới đây, chúng ta sẽ giới thiệu một số khái niệm quan trọng khi xét một tập hợp có thứ tự.

Định nghĩa 4.13. Cho $(X, \leq)$ là một tập hợp có thứ tự, và $A \subset X$.

Ta gọi một phần tử $a \in A$ là một *phần tử nhỏ nhất* của tập hợp A nếu và chỉ nếu với mọi $x \in A$ ta có : $a \leq x$.

Ta gọi một phần tử $a \in A$ là một *phần tử lớn nhất* của tập hợp A nếu và chỉ nếu với mọi $x \in A$ ta có : $x \leq a$.

Định nghĩa 4.14. Ta gọi một phần tử $a \in A$ là một *phần tử tối tiểu* của tập hợp A nếu và chỉ nếu không tồn tại $x \in A$ sao cho $x \neq a$ và $x \leq a$.

Ta gọi một phần tử $a \in A$ là một *phần tử tối đại* của tập hợp A nếu và chỉ nếu không tồn tại $x \in A$ sao cho $x \neq a$ và $a \leq x$.

Nhận xét.

1. Phần tử nhỏ nhất (lớn nhất) của một tập hợp, nếu có, là duy nhất.Ta ký hiệu phần tử nhỏ nhất của một tập hợpA là $\min A$ hay $\min(A)$, và ký hiệu phần tử lớn nhất của A là $\max A$ hay $\max(A)$.
2. Phần tử tối tiểu (tối đại) của một tập hợp có thứ tự không nhất thiết là duy nhất.

Ví dụ 4.18. Ví dụ: xét tập hợp $X = \{1, 2, 3\}$ với quan hệ 2 ngôi ρ được cho bởi $\rho = \{(1, 1), (2, 2), (3, 3), (1, 2), (3, 2)\}$. Chúng ta có thể kiểm chứng rằng (X, ρ) là một tập hợp có thứ tự. Với thứ tự ρ này, X có 2 phần tử tối tiểu là 1 và 3.

3. Phần tử lớn nhất (nhỏ nhất) của một tập hợp, nếu có, là phần tử tối đại (tối tiểu) duy nhất của tập hợp đó.

Ví dụ 4.19. 1. Trong tập hợp có thứ tự $(\mathbb{Z}, \leq)$, tập hợp

$$A = \{m \in \mathbb{Z} | m^2 < 100\}$$

có phần tử nhỏ nhất là -9, và phần tử lớn nhất là 9. Ta có thể viết: $\min(A) = -9; \max(A) = 9$.

2. Trong tập hợp có thứ tự $(R, \leq)$,

$$A = \{x \in R | x^2 < 100\}$$

không có phần tử nhỏ nhất và cũng không có phần tử lớn nhất.

3. Cho E là một tập hợp. Ta đã biết $(P(E), \subset)$ là một tập hợp có thứ tự. Với thứ tự này $P(E)$ có phần tử nhỏ nhất là $\emptyset$, phần tử lớn nhất là E.

Định nghĩa 4.15. Cho $(X, \leq)$ là một tập hợp có thứ tự, và $A \subset X$. Ta gọi một phần tử $x \in X$ là một *chặn dưới* của tập hợp A nếu và chỉ nếu với mọi $a \in A$ ta có : $x \leq a$. *Chặn dưới lớn nhất* (nếu có), tức là phần tử lớn nhất trong tập hợp tất cả những chặn dưới của A được ký hiệu là $\inf(A)$.

Định nghĩa 4.16. Ta gọi một phần tử $x \in X$ là một *chặn trên* của tập hợp A nếu và chỉ nếu với mọi $a \in A$ ta có : $a \leq x$. *Chặn trên nhỏ nhất* (nếu có), tức là phần tử nhỏ nhất trong tập hợp tất cả những chặn trên, của A được ký hiệu là $\sup(A)$.

Ví dụ 4.20. Trong $(\mathbb{R}, \leq)$, $A = \{x \in \mathbb{R} | x^2 < 100|\}$. Ta cớ $\sup(A) = 10$ và $\inf(A) = -10$.

Định nghĩa 4.17 (Thứ tự tốt). Một tập hợp có thứ tự được gọi là có *thứ tự tốt* (hay được sắp tốt) nếu và chỉ nếu mọi tập con khác rỗng đều có phần tử nhỏ nhất.

Ví dụ 4.21. 1. Tập hợp thứ tự $(\mathbb{N}, \leq)$ là một tập hợp được sắp tốt.

2. Tập hợp có thứ tự $(\mathbb{Z}, \leq)$ không phải là một tập hợp được sắp tốt vì $\mathbb{Z}$ không có phần tử nhỏ nhất.

4.5.2. Biểu diễn quan hệ thứ tự

Giả sử R là một quan hệ 2 ngôi giữa một tập hợp hữu hạn $A = \{a_1, a_2, ..., a_m\}$ và một tập hữu hạn $B = \{b_1, b_2, ..., b_m\}$. Quan hệ R có thể được biểu diễn bởi ma trận $MR = (mij)$ gồm m dòng và n cột (tức là ma trận cấp $m \times n$), trong đó

$$m_{ij} = \begin{cases} 1 & \text{nếu } (a_i, b_j) \in R \\ 0 & \text{nếu } (a_i, b_j) \notin R \end{cases}$$

Ta gọi ma trận MR là *ma trận biểu diễn* của quan hệ R

Ví dụ 4.22. Với $A = \{1, 2, 3\}$ và $B = \{a, b, c\}$, thì các quan hệ sau đây
$R = \{(1, a), (1, b), (1, c)\}$
$S = \{(1, a), (1, b), (1, c), (2, b), (2, c), (3, c)\}$
có các ma trận biểu diễn là

$$MR = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad MS = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Trong trường hợp R là một quan hệ 2 ngôi trên một tập X hữu hạn và có n phần tử thì ma trận biểu diễn của R là một ma trận có n dòng và n cột (tức là ma trận vuông cấp n).

Chú ý. Ngoài cách biểu diễn quan hệ dưới dạng ma trận ta còn biểu đồ (dạng đồ thị) để biểu diễn quan hệ. Cách biểu diễn này sẽ được xét đến trong phần sau, khi nói về biểu đồ Hasse của một cấu trúc thứ tự.

4.5.3. Sắp xếp topo

Sắp xếp topo là một vấn đề quan trọng trong việc khảo sát các cấu trúc thứ tự hữu hạn và phương pháp sắp xếp topo thường được sử dụng để giải nhiều bài toán thực tế. Chúng ta thử xem bài toán đặt ra như sau:

Giả sử có một đề tài gồm 20 công việc khác nhau. Một số công việc chỉ có thể được thực hiện sau khi một số công việc khác được thực hiện hoàn tất. Chúng ta phải thực hiện các công việc theo thứ tự nào?

Để mô hình cho vấn đề, chúng ta đặt X là tập hợp 20 công việc của đề tài; trên X ta xét một thứ tự (hay quan hệ thứ tự) $\leq$ sao cho $a \leq b$ nếu và chỉ nếu a và b là 2 công việc trong đó công việc b chỉ có thể được bắt đầu khi công việc a đã được hoàn thành.

Muốn có một kế hoạch thực hiện các công việc cho đề tài chúng ta phải tìm ra một thứ tự cho tất cả 20 công việc "tương thích" với thứ tự $\leq$ nêu trên.

Trước hết chúng ta nêu ra định nghĩa khái niệm “tương thích" như sau

Định nghĩa 4.18. Cho (X, ρ) là một tập hợp có thứ tự. Một thứ tự toàn phần $\leq$ trên X được gọi là *tương thích* với thứ tự ρ nếu và chỉ nếu

$$a\rho b \Rightarrow a \leq b, \forall a, b \in X.$$

Việc xây dựng thứ tự toàn phần tương thích với một thứ tự cho trước được gọi là *sắp xếp topo*.

Thuật toán sắp xếp topo cho một tập hợp hữu hạn có thứ tự dựa vào kết quả nêu trong các định lý trên.

Để định nghĩa một thứ tự toàn phần trên (X, ρ), trước hết ta chọn ra một phần tử tối tiểu a_1 của X; phần tử này tồn tại do định lý 1.

Kế đó, chú ý rằng Nếu tập hợp $X \setminus \{a_1\} \neq \emptyset$ thì $(X \setminus \{a_1\}, \rho)$ cũng là một tập hợp hữu hạn (khác rỗng) có thứ tự. Ta lại chọn ra phần tử tối tiểu a_2 trong $X \setminus \{a_1\}$, rồi loại a_2 khỏi việc xem xét ở bước tiếp theo để chọn phần tử tối tiểu trong $X \setminus \{a_1, a_2\}$ nếu tập hợp $X \setminus \{a_1, a_2\}$ khác rỗng.

Tiếp tục quá trình này bằng cách chọn phần tử tối tiểu a_{k+1} trong $X \setminus \{a_1, a_2, ..., a_k\}$ khi tập hợp còn phần tử.

Vì X là một tập hợp hữu hạn nên quá trình chọn trên phải dừng. Cuối c ng ta đã sắp các phần tử của tập hợp X thành một dãy $a_1, a_2, ..., a_n$ thỏa điều kiện : với mọi i, j sao cho $i < j$ ta có $a_i \rho a_j$ hoặc a_i và a_j không có quan hệ trong thứ tự ρ. Như vậy chọn thứ tự toàn phần $\leq$ trên X được xác định bởi dây chuyền

$$a_1 \leq a_2 \leq ... \leq a_n$$

thì ta được một thứ tự toàn phần trên X tương thích với thứ tự ρ. Thuật toán sắp xếp topo có thể được viết dưới dạng mã giả như sau:

Thuật toán. Sắp xếp topo

Đầu vào : (X, ρ) là một cấu trúc thứ tự hữu hạn.

Đầu ra : Dãy $S = \{a_1, a_2, ..., a_n\}$ là một sự sắp xếp topo của (X, ρ).

1. $k := 1$; $S := \emptyset$;
2. while $X \neq \emptyset$ do

   begin

      $a_k :=$ một phần tử tối tiểu của X // (xem định lý 1 và thuật toán 1)

      $S := S \cup \{a_k\}$

      $X := X \setminus \{a_k\}$

      $k := k + 1$

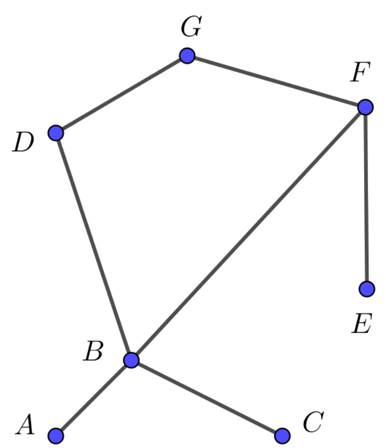
   end
3. Xuất S.

Ví dụ 4.23. Tìm một thứ tự toàn phần tương thích với tập hợp có thứ tự $(\{1, 5, 2, 4, 12, 20\}, |)$.

Lời giải. Đầu tiên ta chọn một phần tử tối tiểu. Phần tử này phải là 1 vì đó là phần tử tối tiểu duy nhất (hay phần tử nhỏ nhất). Kế đó ta chọn phần tử tối tiểu của $(\{5, 2, 4, 20, 12\}, |)$. Có 2 phần tử tối tiểu trong tập hợp có thứ tự này là 2 và 5. Ta chọn 5. Những phần tử còn lại là $\{2, 4, 20, 12\}$. Trong tập hợp này ta chọn một phần tử tối tiểu, ví dụ là 2. Tiếp tục quá trình này ta lần lượt chọn ra được các phần tử 4, 20, và 12. Cuối cùng ta được một thứ tự toàn phần cho bởi:

$$1 \leq 5 \leq 2 \leq 4 \leq 20 \leq 12.$$

Ví dụ 4.24. Giả sử có một đề tài ở một công ty máy tính đòi hỏi phải hoàn thành 7 công việc A, B, C, D, E, F, G. Một số công việc chỉ có thể bắt đầu sau khi đã hoàn thành một số công việc khác. Nói một c ách khác, có một thứ tự ρ được định ra trên tập hợp các công việc như sau : (công việc X) ρ (công việc Y) nếu và chỉ nếu công việc Y chỉ có thể bắt đầu khi công việc X đã được thực hiện hoàn tất. Biểu đồ Hasse cho 7 công việc ứng với thứ tự này là : Hãy tìm một thứ tự thực hiện cho 7 công việc để có thể hoàn



Hình 4.1. Biểu đồ 7 công việc

thành đề tài.

Lời giải. Áp dụng phương pháp sắp xếp topo ta đạt được một thứ tự có thể thực hiện cho các công việc là

$$A \leq C \leq B \leq D \leq E \leq F \leq G.$$

4.6 Dàn (lattice - tập bị chặn)

Định nghĩa 4.19. Cho $(L, \leq)$ là một tập hợp có thứ tự. Ta nói $(L, \leq 4)$ là một *dàn* nếu và chỉ nếu với mọi $a, b \in L$, tập hợp $\{a, b\}$ có chặn dưới lớn nhất và có chặn trên nhỏ nhất; tức là tồn tại $\sup(a, b)$ và $\inf(a, b)$. Ta sẽ dùng ký hiệu $a \vee b$ và $a \wedge b$ để chỉ $\sup(a, b)$ và $\inf(a, b)$

$$a \vee b = \sup(a, b) \qquad a \wedge b = \inf(a, b).$$

1. Tập hợp có thứ tự toàn phần là một dàn, với $a \vee b = \max(a, b)$ và $a \wedge b = \min(a, b)$.
2. Trong dàn $(L, \leq)$, phần tử $\sup(a, b) = a \vee b$ được đặc trưng bởi 2 tính chất sau:

   (1) $a \leq a \vee b$ và $b \leq a \vee b$,

   (2) $\forall c \in L : (a \leq c$ và $b \leq c) \Rightarrow (a \vee b \leq c)$.
3. Trong dàn $(L, \leq)$, phần tử $\inf(a, b) = a \wedge b$ được đặc trưng bởi 2 tính chất sau:

   (1) $a \wedge b \leq a$ và $a \wedge b \leq b$,

   (2) $\forall c \in L : (c \leq a$ và $c \leq b) \Rightarrow (c \leq a \wedge b)$.

Ví dụ 4.25. Cho E là một tập hợp; Tập hợp $(P(E), \subseteq)$ là một dàn. Với mọi $A, B \in P(E)$, ta thấy $A \cup B$ và $A \cap B$ lần lượt chính là chặn trên nhỏ nhất và chặn dưới lớn nhất theo thứ tự $\subseteq$. Nói cách khác, ta có $A \vee B = A \cup B, \quad A \wedge B = A \cap B$

Ví dụ 4.26. Ta đã biết rằng $(N, |)$ là một tập hợp có thứ tự. Theo thứ tự $|$, thứ tự "chia hết", với 2 số tự nhiên a và b ta có chặn trên nhỏ nhất chính là bội số chung nhỏ nhất của chúng, chặn dưới lớn nhất chính là ước số chung lớn nhất của chúng. Vậy $(N, |)$ là một dàn, và ta có

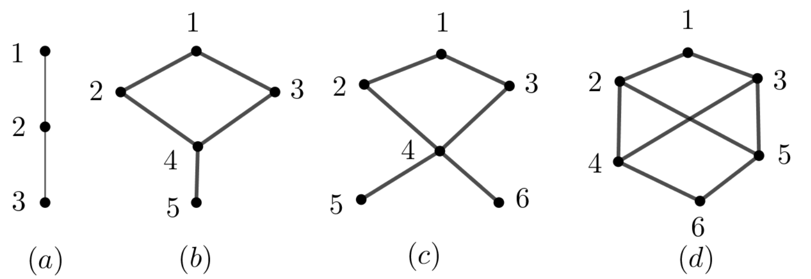
$a \vee b = [a, b]$ (bội số chung nhỏ nhất của a và b)

$a \wedge b = (a, b)$ (ước số chung lớn nhất của a và b)

Ví dụ 4.27. Cho n là một số tự nhiên. Đặt D_n là tập hợp tất cả các ước số không âm của n. Ta có $(D_n, |)$ là một tập hợp có thứ tự. Cho a và b là 2 ước số không âm của n, ta có n là một bội số chung của a và b. Do đó bội số chung nhỏ nhất $[a,b]$ của a và b cũng là một ước số của n. Vậy $[a,b]$ chính là chặn trên nhỏ nhất của a và b trong D_n. Ước số chung lớn nhất (a,b) của a và b cũng là một ước số của n, nên ta có (a,b) chính là chặn dưới lớn nhất của a và b trong D_n.

Tóm lại, ta có thể kết luận rằng $(D_n, |)$ là một dàn.

Ví dụ 4.28. Các tập hợp có thứ tự được biểu diễn bởi các biểu đồ Hasse trong hình dưới đây có phải là dàn hay không?



Hình 4.2. Các tập hợp có thứ tự được biểu diễn bởi (a) và (b) là các dàn

Tập hợp có thứ tự được biểu diễn bởi (c) không phải là một dàn vì $5 \wedge 6$ không tồn tại.

Tập hợp có thứ tự được biểu diễn bởi (d) không phải là một dàn vì $4 \vee 5$ không tồn tại.

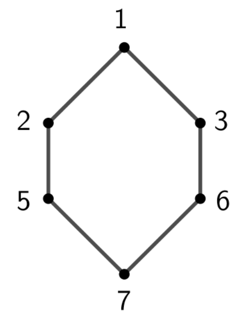
Chú ý. Với 2 dàn X và Y ta có thể định ra một thứ tự trên tích Descartes $X \times Y$ dựa vào các thứ tự trên X và trên Y để $X \times Y$ trở thành một dàn (xem phần bài tập).

Sau đây chúng ta sẽ giới thiệu tiếp khái niệm "dàn con" và "đồng cấu dàn".

Định nghĩa 4.20. Cho$(L, \leq)$ là một dàn và B là một tập hợp con của L. Ta nói B là một *dàn con* của L khi và chỉ khi với mọi $a, b \in B$ ta có $a \vee b \in B$ và $a \wedge b \in B$.

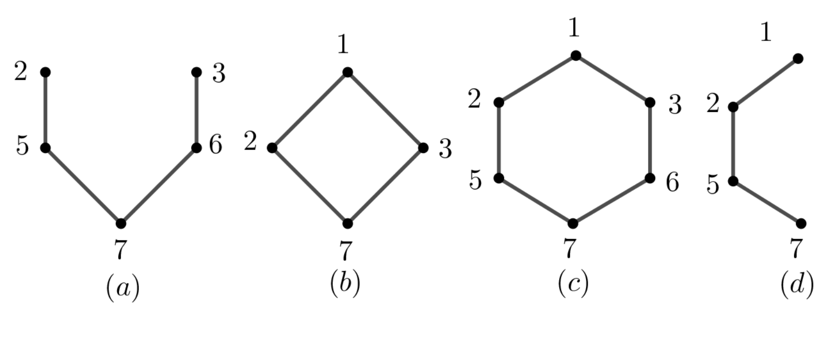
Ví dụ 4.29. Cho một số tự nhiên n, ta có D_n là một dàn con của dàn $(N, |)$.

Ví dụ 4.30. Xem dàn L có biểu đồ Hasse như sau



Hình 4.3. Dàn có biểu đồ Hasse

Trong cấu trúc thứ tự có các biểu đồ Hasse như dưới đây, cấu trúc nào là một dàn con của dàn L? Xem dàn L có biểu đồ Hasse như sau Tập hợp có thứ tự (a) không phải là một dàn con của L



Hình 4.4. Dàn con

vì $2 \vee 3$ không thuộc tập hợp đó.

Tập hợp có thứ tự (c) cũng không phải là một dàn con của L vì $2 \wedge 3$ không thuộc tập hợp đó.

Các tập hợp có thứ tự (b) và (d) đúng là các dàn con của L.

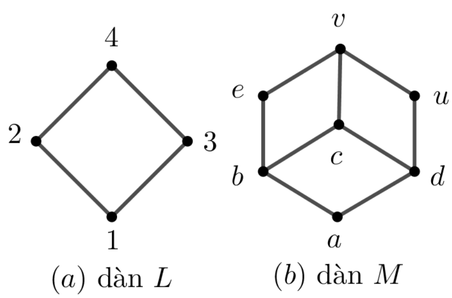
Ví dụ 4.31. Cho $(L, \leq)$ là một dàn và a, b là các phần tử thuộc L. Đặt
$$[a, b] = \{x \in L | a \leq x \text{ và } x \leq b\}$$
Chứng minh rằng $[a, b]$ là một dàn con của L với mọi a,b thỏa $a \leq b$.

Việc chứng minh tính chất này khá đơn giản và được dành cho phần bài tập.

Định nghĩa 4.21. Cho $(L, \leq)$ và $(M, \leq)$ là các dàn. Một ánh xạ $f : L \to M$ được gọi là một *đồng cấu dàn* nếu và chỉ nếu
$$\forall x, y \in L : x \leq y \Rightarrow f(x) \leq f(y)$$
Trường hợp f có thêm tính chất song ánh thì ta nói f là một *đẳng cấu dàn*.

Ghi chú. Ta có thể chứng minh được rằng nếu $f : L \to M$ là một đẳng cấu dàn thì với mọi $x, y \in L$ ta có
$$f(x \vee y) = f(x) \vee f(y)$$
$$f(x \wedge y) = f(x) \wedge f(y)$$

Ví dụ 4.32. Xem hai dàn L và M có biểu đồ Hasse như dưới đây



Hình 4.5. Biểu đồ Hasse

Ánh xạ $f : L \to M$ được định nghĩa bởi
$$f(1) = b, f(2) = e, f(3) = c, f(4) = v$$

là một đồng cấu dàn.

Ánh xạ $g : L \to M$ được định nghĩa bởi :

$$g(1) = a, g(2) = b, g(3) = d, g(4) = v$$

không phải là một đồng cấu dàn vì $g(2) \vee g(3) = b \vee d = c$, nhưng $g(2 \vee 3) = g(4) = v \neq c$.

Ví dụ 4.33. Cho $E = \{a, b\}$. Hai dàn $(P(E), \subset)$ và $(D_{10}, |)$ đẳng cấu với nhau. Thật vậy, xét ánh xạ $f : P(E) \to D_{10}$ được định nghĩa bởi

$$f(\emptyset) = 1, f(\{a\}) = 2, f(\{b\}) = 5, f(\{a, b\}) = 10$$

ta có thể kiểm tra dễ dàng f là một đẳng cấu dàn. Điều này có thể thấy rõ khi ta quan sát các biểu đồ Hasse của 2 dàn trên.

Bây giờ chúng ta sẽ phát biểu một số tính chất của dàn. Việc chứng minh các tính chất này được xem như bài tập.

Định lý 4.4. *Với mọi phần tử x, y, z thuộc dàn $(L, \leq)$ ta có*

1. $x \vee x = x, x \wedge x = x$ *(tính lũy đẳng)*
2. $x \vee y = y \vee x, x \wedge y = y \wedge x$ *(tính giao hoán)*
3. $x \vee (y \vee z) = (x \vee y) \vee z$ *(tính kết hợp)*
   $x \wedge (y \wedge z) = (x \wedge y) \wedge z$
4. $(x \leq y) \Leftrightarrow (x \vee y = y) \Leftrightarrow (x \wedge y = x)$
5. $x \wedge (x \vee y) = x = x \vee (x \wedge y)$.

Định lý 4.5. *Với mọi phần tử a, b, c, d thuộc dàn $(L, \leq)$ ta có*

1. $(a \leq b) \Rightarrow (a \vee c \leq b \vee c$ *và* $a \wedge c \leq b \wedge c)$
2. $(a \leq b$ *và* $c \leq d) \Rightarrow (a \vee c \leq b \vee d$ *và* $a \wedge c \leq b \wedge d)$.

Định lý 4.6. *Với mọi phần tử x, y, z thuộc dàn $(L, \leq)$ ta có*

1. $x \wedge (y \vee z) \geq (x \wedge y) \vee (x \wedge z)$.
2. $x \vee (y \wedge z) \leq (x \vee y) \wedge (x \vee z)$.
3. $(x \leq z) \Rightarrow (x \vee (y \wedge z) \leq (x \vee y) \wedge z)$.

4.7 Bài tập

▷ **4.1.** Cho 2 tập hợp $X = \{a, b, c\}$ và $Y = \{b, c, d, e\}$

a. Tính $|X \times Y|$.

b. Tìm số quan hệ 2 ngôi trên Y.

c. Tìm số quan hệ giữa X và Y chứa $(b, c), (b, d)$.

d. Hãy tìm 1 quan hệ trên X có tính phản xạ và bắc cầu nhưng không đối xứng.

e. Hãy tìm 1 quan hệ trên Y có tính phản xạ và đối xứng nhưng không bắc cầu.

▷ **4.2.** Cho $R \subseteq A \times A$. Ta định nghĩa $R^n (n = 1, 2, ...)$ bằng quy nạp như sau

$$R^1 = R, R^{n+1} = R^n.R.$$

a) cho $R = \{(1, 1), (2, 1), (3, 2), (4, 3)\}$. Tìm $R^n (n = 1, 2, ...)$

b) Chứng minh tính chất: Quan hệ R trên tập A là bắc cầu khi và chỉ khi $R^n \subseteq R$.

▷ **4.3.** Cho $R \subseteq A \times B$. Quan hệ ngược của R là

$$R^{-1} = \{(b, a) : b \in B, a \in A \text{ và } (a, b) \in R\}.$$

Còn quan hệ bù của R là $\overline{R} = \{(a, b) : (a, b) \notin R\}$.

a) Cho $R = \{(a, b) : a < b\}$ trên tập hợp các số nguyên. Tìm R^{-1} và $\overline{R}$.

b) Cho R là quan hệ trên tập tất cả các tỉnh, thành phố của Việt Nam và xác định $R = \{(a, b) : a \text{ là tỉnh giáp với tỉnh } b\}$. Tìm R^{-1} và $\overline{R}$.

▷ **4.4.** Chứng minh rằng

a) nếu quan hệ R trên tập A có tính đối xứng và bắc cầu thì R có tính phản xạ.

b) nếu quan hệ R trên tập A là phản xạ khi và chỉ khi quan hệ R^{-1} là phản xạ.

c) nếu quan hệ R trên tập A là phản xạ khi và chỉ khi quan hệ $\overline{R}$ là không phản xạ.

▷ **4.5.** R là một quan hệ trên $A = 1, 2, 3, 4, 5$ với
$R = \{(1,1), (1,2), (1,3), (1,4), (2,1), (2,2), (2,4), (2,5), (3,1), (3,3), (4,1), (4,2), (4,4), (5,2), (5,5)\}$
R có phải là 1 quan hệ tương đương hay không?

▷ **4.6.** Cho R là 1 quan hệ trên tập hợp các số tự nhiên với

$$R = \{(x, y) : x + y \text{ chẵn }\}.$$

Chứng minh R là 1 quan hệ tương đương.

▷ **4.7.** Cho R là 1 quan hệ trên $A \times A$ với $A = \{1, 3, 5, 7, 8, 9\}$ sao cho

$$(a, b)R(c, d) \Leftrightarrow b = d$$

a. Chứng minh R là 1 quan hệ tương đương.

b. Tìm lớp tương đương chứa $(1, 3)$.

c. Phân hoạch $A \times A$ thành các lớp tương đương tách biệt phân hoạch trên R.

▷ **4.8.** Giả sử R là quan hệ trên các tập các xâu chữ cái tiếng Anh sao cho aRb khi và chỉ khi $l(a) = l(b)$, ở đây $l(x)$ là độ dài của xâu x. R có phải quan hệ tương đương không?

▷ **4.9.** Chứng minh rằng quan hệ R trên các tập số nguyên được xác định aRb khi và chỉ khi $a = b$ hoặc $a = -b$ là quan hệ tương đương.

▷ **4.10.** Các quan hệ nào trong số các quan hệ trên tập $\{0, 1, 2, 3\}$ cho dưới đây là quan hệ tương đương? Xác định các tính chất của một quan hệ tương đương mà các quan hệ khác không có

a) $\{(0,0), (1,1), (2,2), (3,3)\}$;

b) $\{(0,0), (0,2), (2,0), (2,2), (2,3), (3,2), (3,3)\}$;

c) $\{(0,0), (1,1), (1,2), (2,1), (2,2), (3,3)\}$;

d) $\{(0,0), (1,1), (1,3), (2,2), (2,3), (3,1), (3,2), (3,3)\}$;

e) $\{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,2), (3,3)\}$;

▷ **4.11.** Giả sử A là một tệp khác rỗng và f là một hàm số có A là miền xác định. Trên A ta định nghĩa quan hệ R như sau

$$\forall x, y \in A, xRy \Leftrightarrow f(x) = f(y).$$

a) Chứng minh rằng R là quan hệ tương đương.

b) Xác định lớp tương đương của R.

▷ **4.12.** Xác định xem các quan hệ được biểu diễn bởi các ma trận dưới đây có phải là quan hệ tương đương hay không?

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

▷ **4.13.** Giả sử R_1 và R_2 là hai quan hệ tương đương trên tập A. Xác định xem các quan hệ $R_1 \cup R_2$, $R_1 \cap R_2$ có nhất thiết là quan hệ tương đương hay không?

▷ **4.14.** Xác định số các quan hệ tương đương khác nhau trên tập 3 phần tử bằng cách liệt kê ra các quan hệ đó.

▷ **4.15.** Cho $A = \{1, 2, 3, 4, 5\}$. Trên A xác định quan hệ R như sau

$$\forall a, b \in A : aRb \Leftrightarrow a + b = 2k, (k = 1, 2, ...).$$

a) Biểu diễn R bằng các phương pháp liệt kê, ma trận và đồ thị có hướng.

b) Chứng minh rằng R là quan hệ tương đương trên A.

c) Tìm phân hoạch tương đương trên A do R sinh ra.

d) Cho $A_1 = \{1\}, A_2 = \{2, 3\}, A_3 = \{4, 5\}$. Tìm quan hệ tương đương S trên A mà S sinh ra phân hoạch A_1, A_2, A_3.

e) Chứng minh rằng $F = R \cup S$ là một quan hệ trên A mà $MF = RR \vee MS$.

▷ **4.16.** Cho $A = \{1, 2, 3, 4, 5, 6\}$. Trên A xác định quan hệ R như sau

$$\forall a, b \in A : aRb \Leftrightarrow a - b = 3k, (k = 0, \pm 1, \pm 2, ...).$$

a) Biểu diễn R bằng các phương pháp liệt kê, ma trận và đồ thị có hướng.

b) Chứng minh rằng R là quan hệ tương đương trên A.

c) Tìm phân hoạch tương đương trên A do R sinh ra.

d) Tìm quan hệ tương đương S trên A sinh ra các tập $A_1 = \{1\}, A_2 = \{2,3\}, A_3 = \{4,5,6\}$.

e) Chứng minh rằng $R^{-1} = R$.

▷ **4.17.** R là một quan hệ trên $A \times A$ với $A = \{1,2,3,4,5\}$ $R = \{(1,1),(1,2),(1,3),(1,4),(2,2),(2,4),(2,5),$
$(3,3),(3,5),(4,4),(4,5),(5,5)\}$

R có phải là 1 quan hệ thứ tự hay không?

▷ **4.18.** Cho là 1 quan hệ trên $X = \{2,3,4,5,12,15,60\}$ xác định bởi

$$\forall x,y \in X, x \prec y \Leftrightarrow y = kx, k \in \mathbb{Z}^+.$$

a. Chứng minh $\prec$ là 1 quan hệ thứ tự.

b. Tìm các phần tử tối đại, tối tiểu, lớn nhất, nhỏ nhất xác định bởi quan hệ trên

c. Vẽ biểu đồ Hasse tương ứng

▷ **4.19.** Cho là 1 quan hệ trên $A \times A$ với $A = \{2,4,6,12,24\}$ xác định bởi

$$\forall (x,y),(z,t) \in A, (x,y) \prec (z,t) \Leftrightarrow x + y = z + t.$$

$\prec$ có phải là 1 quan hệ thứ tự hay không? Nếu $\prec$ là 1 quan hệ thứ tự thì xác định các phần tử tối đại, tối tiểu, lớn nhất, nhỏ nhất xác định bởi quan hệ trên.

Chương

5 THUẬT TOÁN

5.1 Khái niệm thuật toán

5.1.1. Mở đầu

Có nhiều lớp bài toán tổng quát xuất hiện trong toán học rời rạc. Chẳng hạn, cho một dãy các số nguyên, tìm số lớn nhất; cho một tập hợp, liệt kê các tập con của nó; cho tập hợp các số nguyên, xếp chúng theo thứ tự tăng dần; cho một mạng, tìm đường đi ngắn nhất giữa hai đỉnh của nó. Khi được giao cho một bài toán như vậy thì việc đầu tiên phải làm là xây dựng một mô hình dịch bài toán đó thành ngữ cảnh toán học. Các cấu trúc rời rạc được dùng trong các mô hình này là tập hợp, dãy, hàm, hoán vị, quan hệ, cùng với các cấu trúc khác như đồ thị, cây, mạng - những khái niệm sẽ được nghiên cứu ở các chương sau.

Lập được một mô hình toán học thích hợp chỉ là một phần của quá trình giải. Để hoàn tất quá trình giải, còn cần phải có một phương pháp dùng mô hình để giải bài toán tổng quát. Nói một cách lý tưởng, cái được đòi hỏi là một thủ tục, đó là dãy các bước dẫn tới đáp số mong muốn. Một dãy các bước như vậy, được gọi là một thuật toán.

Khi thiết kế và cài đặt một phần mềm tin học cho một vấn đề nào đó, ta cần phải đưa ra phương pháp giải quyết mà thực chất đó là thuật toán giải quyết vấn đề này. Rõ ràng rằng, nếu không tìm được một phương pháp giải quyết thì không thể lập trình được. Chính vì thế, thuật toán là khái niệm nền tảng của hầu hết các lĩnh vực của tin họ

5.1.2. Định nghĩa

Thuật toán là một bảng liệt kê các chỉ dẫn (hay quy tắc) cần thực hiện theo từng bước xác định nhằm giải một bài toán đã cho.

Thuật ngữ "Algorithm" (thuật toán) là xuất phát từ tên nhà toán học Ả Rập Al- Khowarizmi. Ban đầu, từ algorism được dùng để chỉ các quy tắc thực hiện các phép tính số học trên các số thập phân. Sau đó, algorism chuyển thành algorithm vào thế kỷ 19. Với sự quan tâm ngày càng tăng đối với các máy tính, khái niệm thuật toán đã được cho một ý nghĩa chung hơn, bao hàm cả các thủ tục xác định để giải các bài toán, chứ không phải chỉ là thủ tục để thực hiện các phép tính số học.

Có nhiều cách trình bày thuật toán: dùng ngôn ngữ tự nhiên, ngôn ngữ lưu đồ (sơ đồ khối), ngôn ngữ lập trình. Tuy nhiên, một khi dùng ngôn ngữ lập trình thì chỉ những lệnh được phép trong ngôn ngữ đó mới có thể dùng được và điều này thường làm cho sự mô tả các thuật toán trở nên rối rắm và khó hiểu. Hơn nữa, vì nhiều ngôn ngữ lập trình đều được dùng rộng rãi, nên chọn một ngôn ngữ đặc biệt nào đó là điều người ta không muốn. Vì vậy ở đây các thuật toán ngoài việc được trình bày bằng ngôn ngữ tự nhiên cùng với những ký hiệu toán học quen thuộc còn dùng một dạng giả mã để mô tả thuật toán. Giả mã tạo ra bước trung gian giữa sự mô tả một thuật toán bằng ngôn ngữ thông thường và sự thực hiện thuật toán đó trong ngôn ngữ lập trình. Các bước của thuật toán được chỉ rõ bằng cách dùng các lệnh giống như trong các ngôn ngữ lập trình.

Ví dụ 5.1. Mô tả thuật toán tìm phần tử lớn nhất trong một dãy hữu hạn các số nguyên.

a) *Dùng ngôn ngữ tự nhiên để mô tả các bước cần phải thực hiện.*

1. Đặt giá trị cực đại tạm thời bằng số nguyên đầu tiên trong dãy. (Cực đại tạm thời sẽ là số nguyên lớn nhất đã được kiểm tra ở một giai đoạn nào đó của thủ tục.)
2. So sánh số nguyên tiếp sau với giá trị cực đại tạm thời, nếu nó lớn hơn giá trị cực đại tạm thời thì đặt cực đại tạm thời bằng số nguyên đó.
3. Lặp lại bước trước nếu còn các số nguyên trong dãy.
4. Dừng khi không còn số nguyên nào nữa trong dãy. Cực đại tạm thời ở điểm này chính là số nguyên lớn nhất của dãy.

b) *Dùng đoạn giả mã.*

Thuật toán 5.1. Tìm phần tử lớn nhất trong dãy số nguyên

```
procedure max(a_1, a_2, ..., a_n:integer)
  max := a_1
  for i := 2 to n
    if max < a_i then max := a_i
//max là phần tử lớn nhất
```

Thuật toán này trước hết gán số hạng đầu tiên a 1 của dãy cho biến max. Vòng lặp "for" được dùng để kiểm tra lần lượt các số hạng của dãy. Nếu một số hạng lớn hơn giá trị hiện thời của max thì nó được gán làm giá trị mới của max.

5.1.3. Các đặc trưng của thuật toán

– **Đầu vào (Input):** Một thuật toán có các giá trị đầu vào từ một tập đã được chỉ rõ.

– **Đầu ra (Output):** Từ mỗi tập các giá trị đầu vào, thuật toán sẽ tạo ra các giá trị đầu ra.

Các giá trị đầu ra chính là nghiệm của bài toán.

– **Tính dừng:** Sau một số hữu hạn bước thuật toán phải dừng.

– **Tính xác định:** Ở mỗi bước, các bước thao tác phải hết sức rõ ràng, không gây nên sự nhập nhằng. Nói rõ hơn, trong cùng một điều kiện hai bộ xử lý cùng thực hiện một bước của thuật toán phải cho những kết quả như nhau.

– **Tính hiệu quả:** Trước hết thuật toán cần đúng đắn, nghĩa là sau khi đưa dữ liệu vào thuật toán hoạt động và đưa ra kết quả như ý muốn.

– **Tính phổ dụng:** Thuật toán có thể giải bất kỳ một bài toán nào trong lớp các bài toán. Cụ thể là thuật toán có thể có các đầu vào là các bộ dữ liệu khác nhau trong một miền xác định.

5.2 Thuật toán tìm kiếm

5.2.1. Bài toán tìm kiếm

Bài toán xác định vị trí của một phần tử trong một bảng liệt kê sắp thứ tự thường gặp trong nhiều trường hợp khác nhau. Chẳng hạn chương trình kiểm tra chính tả của các từ, tìm kiếm các từ này trong một cuốn từ điển, mà từ điển chẳng qua cũng là một bảng liệt kê sắp thứ tự của các từ. Các bài toán thuộc loại này được gọi là các bài toán tìm kiếm.

Bài toán tìm kiếm tổng quát được mô tả như sau: xác định vị trí của phần tử x trong một bảng liệt kê các phần tử phân biệt $a_1, a_2, ..., a_n$ hoặc xác định rằng nó không có mặt trong bảng liệt kê đó. Lời giải của bài toán trên là vị trí của số hạng của bảng liệt kê có giá trị bằng x (tức là i sẽ là nghiệm nếu $x = a_i$ và là 0 nếu x không có mặt trong bảng liệt kê).

5.2.2. Thuật toán tìm kiếm tuyến tính

Tìm kiếm tuyến tính hay tìm kiếm tuần tự là bắt đầu bằng việc so sánh x với a_1 ; khi $x = a_1$, nghiệm là vị trí a_1 , tức là 1; khi $x \neq a_1$, so sánh x với a_2. Nếu $x = a_2$, nghiệm là vị trí của a_2, tức là 2. Khi $x \neq a_2$, so sánh x với a_3. Tiếp tục quá trình này bằng cách tuần tự so sánh x với mỗi số hạng của bảng liệt kê cho tới khi tìm được số hạng bằng x, khi đó nghiệm là vị trí của số hạng đó. Nếu toàn bảng liệt kê đã được kiểm tra mà không xác định được vị trí của x, thì nghiệm là 0. Giả mã đối với thuật toán tìm kiếm tuyến tính được cho dưới đây

Thuật toán 5.2. Tìm kiếm tuyến tính

```
procedure tk_tuyen_tinh(x: integer,a_1, a_2, ..., a_n:integer phân biệt)
  i := 1
  while (i ≤ n and x ≠ a_i)
  i := i + 1
  if i ≤ n then location := i
     else location := 0
//location là chỉ số dưới của số hạng bằng x hoặc là 0 nếu không tìm được x
```

5.2.3. Thuật toán tìm kiếm nhị phân

Thuật toán này có thể được dùng khi bảng liệt kê có các số hạng được sắp theo thứ tự tăng dần. Chẳng hạn, nếu các số hạng là các số thì chúng được sắp từ số nhỏ nhất đến số lớn nhất hoặc nếu chúng là các từ hay xâu ký tự thì chúng được sắp theo thứ tự từ điển. Thuật toán thứ hai này gọi là thuật toán tìm kiếm nhị phân. Nó được tiến hành bằng cách so sánh phần tử cần xác định vị trí với số hạng ở giữa bảng liệt kê. Sau đó bảng này được tách làm hai bảng kê con nhỏ hơn có kích thước như nhau, hoặc một trong hai bảng con ít hơn bảng con kia một số hạng. Sự tìm kiếm tiếp tục bằng cách hạn chế tìm kiếm ở một bảng kê con thích hợp dựa trên việc so sánh phần tử cần xác định vị trí với số hạng giữa bảng kê. Ta sẽ thấy rằng thuật toán tìm kiếm nhị phân hiệu quả hơn nhiều so với thuật toán tìm kiếm tuyến tính.

Ví dụ 5.2. Để tìm số 19 trong bảng liệt kê 1,2,3,5,6,7,8,10,12,13,15,16,18,19,20,22 ta tách bảng liệt kê gồm 16 số hạng này thành hai bảng liệt kê nhỏ hơn, mỗi bảng có 8 số hạng, cụ thể là: 1,2,3,5,6,7,8,10 và 12,13,15,16,18,19,20,22. Sau đó ta so sánh 19 với số hạng cuối cùng của bảng con thứ nhất. Vì 10<19, việc tìm kiếm 19 chỉ giới hạn trong bảng liệt kê con thứ 2 từ số hạng thứ 9 đến 16 trong bảng liệt kê ban đầu. Tiếp theo, ta lại tách bảng liệt kê con gồm 8 số hạng này làm hai bảng con, mỗi bảng có 4 số hạng, cụ thể là 12,13,15,16 và 18,19,20,22. Vì 16<19, việc tìm kiếm lại được giới hạn chỉ trong bảng con thứ 2, từ số hạng thứ 13 đến 16 của bảng liệt kê ban đầu. Bảng liệt kê thứ 2 này lại được tách làm hai, cụ thể là: 18,19 và 20,22. Vì 19 không lớn hơn số hạng lớn nhất của bảng con thứ nhất nên việc tìm kiếm giới hạn chỉ ở bảng con thứ nhất gồm các số 18,19, là số hạng thứ 13 và 14 của bảng ban đầu. Tiếp theo bảng con chứa hai số hạng này lại được tách làm hai, mỗi bảng có một số hạng 18 và 19. Vì 18<19, sự tìm kiếm giới hạn chỉ trong bảng con thứ 2, bảng liệt kê chỉ chứa số hạng thứ 14 của bảng liệt kê ban đầu, số hạng đó là số 19. Bây giờ sự tìm kiếm đã thu hẹp về chỉ còn một số hạng, so sánh tiếp cho thấy19 là số hạng thứ 14 của bảng liệt kê ban đầu.

Bây giờ ta có thể chỉ rõ các bước trong thuật toán tìm kiếm nhị phân.

Để tìm số nguyên x trong bảng liệt kê $a_1, a_2, ..., a_n$ với $a_1 < a_2 < ... < a_n$, ta bắt đầu bằng việc so sánh x với số hạng a_m ở giữa của dãy, với $m = [(n+1)/2]$. Nếu $x > a_m$, việc tìm kiếm x giới hạn ở nửa thứ hai của dãy, gồm $a_{m+1}, a_{m+2}, ..., a_n$. Nếu x không lớn hơn a_m, thì sự tìm kiếm giới hạn trong nửa đầu của dãy gồm $a_1, a_2, ..., a_m$.

Bây giờ sự tìm kiếm chỉ giới hạn trong bảng liệt kê có không hơn $[n/2]$ phần tử. Dùng chính thủ tục này, so sánh x với số hạng ở giữa của bảng liệt kê được hạn chế. Sau đó lại hạn chế việc tìm kiếm ở nửa thứ nhất hoặc nửa thứ hai của bảng liệt kê. Lặp lại quá trình này cho tới khi nhận được một bảng liệt kê chỉ có một số hạng. Sau đó, chỉ còn xác định số hạng này có phải là x hay không. Giả mã cho thuật toán tìm kiếm nhị phân được cho dưới đây

Thuật toán 5.3. Tìm kiếm nhị phân

```
procedure tk_nhi_phan(x: integer,a1, a2, ..., an:integer tăng)
  i := 1  //i là điểm mút trái của khoảng tìm kiếm
  j := n    //j là điểm mút phải của khoảng tìm kiếm
  while i < j
    begin
    m := [(i + j)/2]
    if x > am then i := m + 1
      else j := m
    end
    if x = ai then location := i
      else location := 0
//location là chỉ số dưới của số hạng bằng x hoặc 0 nếu không tìm thấy x
```

5.3 Độ phức tạp của thuật toán

5.3.1. Khái niệm về độ phức tạp của một thuật toán

Thước đo hiệu quả của một thuật toán là thời gian mà máy tính sử dụng để giải bài toán theo thuật toán đang xét, khi các giá trị đầu vào có một kích thước xác định.

Một thước đo thứ hai là dung lượng bộ nhớ đòi hỏi để thực hiện thuật toán khi các giá trị đầu vào có kích thước xác định. Các vấn đề như thế liên quan đến độ phức tạp tính toán của một thuật toán. Sự phân tích thời gian cần thiết để giải một bài toán có kích thước đặc biệt nào đó liên quan đến độ phức tạp thời gian của thuật toán. Sự phân tích bộ nhớ cần thiết của máy tính liên quan đến độ phức tạp không gian của thuật toán. Vệc xem xét độ phức tạp thời gian và không gian của một thuật toán là một vấn đề rất thiết yếu khi các thuật toán được thực hiện. Biết một thuật toán sẽ đưa ra đáp số trong một micro giây, trong một phút hoặc trong một tỉ năm, hiển nhiên là hết sức quan trọng. Tương tự như vậy, dung lượng bộ nhớ đòi hỏi phải là khả dụng để giải một bài toán,vì vậy độ phức tạp không gian cũng cần phải tính đến.Vì việc xem xét độ phức tạp không gian gắn liền với các cấu trúc dữ liệu đặc biệt được dùng để thực hiện thuật toán nên ở đây ta sẽ tập trung xem xét độ phức tạp thời gian.

Độ phức tạp thời gian của một thuật toán có thể được biểu diễn qua số các phép toán được dùng bởi thuật toán đó khi các giá trị đầu vào có một kích thước xác định. Sở dĩ độ phức tạp thời gian được mô tả thông qua số các phép toán đòi hỏi thay vì thời gian thực của máy tính là bởi vì các máy tính khác nhau thực hiện các phép tính sơ cấp trong những khoảng thời gian khác nhau. Hơn nữa, phân tích tất cả các phép toán thành các phép tính bit sơ cấp mà máy tính sử dụng là điều rất phức tạp.

Ví dụ 5.3. Xét thuật toán tìm số lớn nhất trong dãy n số $a_1, a_2, ..., a_n$. Có thể coi kích thước của dữ liệu nhập là số lượng phần tử của dãy số, tức là n. Nếu coi mỗi lần so sánh hai số của thuật toán đòi hỏi một đơn vị thời gian (giây chẳng hạn) thì thời gian thực hiện thuật toán trong trường hợp xấu nhất là $n-1$ giây. Với dãy 64 số, thời gian thực hiện thuật toán nhiều lắm là 63 giây.

Ví dụ 5.4 (Thuật toán về trò chơi "Tháp Hà Nội")**.** Trò chơi "Tháp Hà Nội" như sau: Có ba cọc A, B, C và 64 cái đĩa (có lỗ để đặt vào cọc), các đĩa có đường kính đôi một khác nhau. Nguyên tắc đặt đĩa vào cọc là: mỗi đĩa chỉ được chồng lên đĩa lớn hơn nó. Ban đầu, cả 64 đĩa được đặt chồng lên nhau ở cột A; hai cột B, C trống.

Vấn đề là phải chuyển cả 64 đĩa đó sang cột B hay C, mỗi lần chỉ được di chuyển một đĩa.

Xét trò chơi với n đĩa ban đầu ở cọc A (cọc B và C trống). Gọi S n là số lần chuyển đĩa để chơi xong trò chơi với n đĩa.

Nếu $n = 1$ thì rõ ràng là $S_1 = 1$.

Nếu $n > 1$ thì trước hết ta chuyển $n - 1$ đĩa bên trên sang cọc B (giữ yên đĩa thứ n ở dưới cùng của cọc A). Số lần chuyển $n - 1$ đĩa là S_{n-1}. Sau đó ta chuyển đĩa thứ n từ cọc A sang cọc C. Cuối cùng, ta chuyển $n - 1$ đĩa từ cọc B sang cọc C (số lần chuyển là S_{n-1}). Như vậy, số lần chuyển n đĩa từ A sang C là

$$\begin{aligned} S_n &= S_{n-1} + 1 + S_n = 2S_{n-1} + 1 = 2(2S_{n-2} + 1) + 1 \\ &= 2^2 S_{n-2} + 2 + 1 = ... = 2^{n-1} S_1 + 2^{n-2} + ... + 2 + 1 = 2^n - 1. \end{aligned}$$

Thuật toán về trò chơi "Tháp Hà Nội" đòi hỏi $2^{64} - 1$ lần chuyển đĩa (xấp xỉ 18,4 tỉ tỉ lần). Nếu mỗi lần chuyển đĩa mất 1 giây thì thời gian thực hiện thuật toán xấp xỉ 585 tỉ năm!

Hai thí dụ trên cho thấy rằng: một thuật toán phải kết thúc sau một số hữu hạn bước, nhưng nếu số hữu hạn này quá lớn thì thuật toán không thể thực hiện được trong thực tế.

Ta nói: thuật toán trong Thí dụ 3 có độ phức tạp là $n - 1$ và là một thuật toán hữu hiệu (hay thuật toán nhanh); thuật toán trong Thí dụ 4 có độ phức tạp là $2^n - 1$ và đó là một thuật toán không hữu hiệu (hay thuật toán chậm).

5.3.2. So sánh độ phức tạp của các thuật toán

Một bài toán thường có nhiều cách giải, có nhiều thuật toán để giải, các thuật toán đó có độ phức tạp khác nhau.

Xét bài toán: Tính giá trị của đa thức
$P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ tại x_0.

Thuật toán 5.4. Tính giá trị đa thức trực tiếp

procedure `tinh_gia_tri_da_thuc1` ($a_0, a_1, a_2, ..., a_n, x_0$: real)
$sum := a_0$

```
for i := 1 to n
        sum := sum + a_i x_0^i
//sum là giá trị của đa thức P(x) tại x_0
```

Chú ý rằng đa thức $P(x)$ có thể viết dưới dạng

$$P(x) = (...((a_n x + a_{n-1})x + a_{n-2})x...)x + a_0.$$

Ta có thể tính $P(x)$ theo thuật toán sau

Thuật toán 5.5. Tính giá trị đa thức theo nhóm số hạng

```
procedure tinh_gia_tri_da_thuc2 (a_0, a_1, a_2, ..., a_n, x_0: real)
P := a_n
for i := 1 to n
        P := P.x_0 + a_{n-i}
//P là giá trị của đa thức P(x) tại x_0
```

Ta hãy xét độ phức tạp của hai thuật toán trên.

Đối với thuật toán 1: ở bước 2, phải thực hiện 1 phép nhân và 1 phép cộng với $i = 1$; 2 phép nhân và 1 phép cộng với $i = 2, ..., n$ phép nhân và 1 phép cộng với $i = n$. Vậy số phép tính (nhân và cộng) mà thuật toán 1 đòi hỏi là:

$$(1+1) + (2+1) + ... + (n+1) = \frac{n(n+1)}{2} + n = \frac{n(n+3)}{2}.$$

Đối với thuật toán 2: bước 2 phải thực hiện n lần, mỗi lần đòi hỏi 2 phép tính (nhân rồi cộng), do đó số phép tính (nhân và cộng) mà thuật toán 2 đòi hỏi là $2n$.

Nếu coi thời gian thực hiện mỗi phép tính nhân và cộng là như nhau và là một đơn vị thời gian thì với mỗi n cho trước, thời gian thực hiện thuật toán 1 là $\frac{n(n+3)}{2}$, còn thời gian thực hiện thuật toán 2 là $2n$.

Rõ ràng là thời gian thực hiện thuật toán 2 ít hơn so với thời gian thực hiện thuật toán 1. Hàm $f_1(n) = 2n$ là hàm bậc nhất, tăng chậm hơn nhiều so với hàm bậc hai $f_2(n) = \frac{n(n+3)}{2}$.

Ta nói rằng thuật toán 2 (có độ phức tạp là 2n) là thuật toán hữu hiệu hơn (hay nhanh hơn) so với thuật toán 1 (có độ phức tạp là $\frac{n(n+3)}{2}$).

Để so sánh độ phức tạp của các thuật toán, điều tiện lợi là coi độ phức tạp của mỗi thuật toán như là cấp của hàm biểu hiện thời gian thực hiện thuật toán ấy.

Các hàm xét sau đây đều là hàm của biến số tự nhiên $n > 0$.

Định nghĩa 5.1. Ta nói hàm$f(n)$ có cấp thấp hơn hay bằng hàm $g(n)$ nếu tồn tại hằng số $C > 0$ và một số tự nhiên n_0 sao cho

$$|f(n)| \leq C|g(n)| \text{ với mọi } n \geq n_0.$$

Ta viết $f(n) = O(g(n))$ và còn nói $f(n)$ thoả mãn quan hệ big-O đối với $g(n)$.

Theo định nghĩa này, hàm $g(n)$ là một hàm đơn giản nhất có thể được, đại diện cho "sự biến thiên" của $f(n)$.

Khái niệm big-O đã được dùng trong toán học đã gần một thế kỷ nay. Trong tin học, nó được sử dụng rộng rãi để phân tích các thuật toán. Nhà toán học người Đức Paul Bachmann là người đầu tiên đưa ra khái niệm big-O vào năm 1892.

Ví dụ 5.5. Hàm $f(n) = \frac{n(n+3)}{2}$ là hàm bậc hai và hàm bậc hai đơn giản nhất là n^2. Ta có

$$f(n) = \frac{n(n+3)}{2} = O(n^2),$$

vì $\frac{n(n+3)}{2} \leq n^2$ với mọi $n \geq 3 \quad (C = 1, n_0 = 3)$

Ví dụ 5.6. Một cách tổng quát ví dụ trên, nếu $f(n) = a_k n^k + a_{k-1}n^{k-1} + \cdots + a_1 n + a_0$ thì $f(n) = O(n^k)$.

Lời giải. Thật vậy, với $n > 1$,

$$\begin{aligned}|f(n)| &\le |a_k|n^k + |a_{k-1}|n^{k-1} + \cdots + |a_1|n + |a_0| \\ &= n^k\left(|a_k| + \frac{|a_{k-1}|}{n} + \cdots + \frac{|a_1|}{n^{k-1}} + \frac{|a_0|}{n^k}\right) \\ &\le n^k(|a_k| + |a_{k-1}| + \cdots + |a_1| + |a_0|).\end{aligned}$$

Điều này chứng tỏ $|f(n)| \le Cn^k$ với mọi $n > 1$.

Ví dụ 5.7. Hãy chỉ ra nếu $g(n) = 3n + 5n\log_2 n$ thì $g(n) = O(n\log_2 n)$.

Lời giải. Thật vậy,

$$3n + 5n\log_2 n = n(3 + 5\log_2 n) \le n(\log_2 n + 5\log_2 n) = 6n\log_2 n$$

với mọi $n \ge 8 (C = 6, n_0 = 8)$.

Mệnh đề 5.1. *Cho* $f_1(n) = O(g_1(n))$ *và* $f_2(n) = O(g_2(n))$. *Khi đó*

a) $(f_1 + f_2)(n) = O(\max(|g_1(n)|, |g_2(n)|)$,

b) $(f_1 f_2)(n) = O(g_1(n)g_2(n))$.

Chứng minh. Theo giả thiết, tồn tại C_1, C_2, n_1, n_2 sao cho $|f_1(n)| \le C_1|g_1(n)|$ và $|f_2(n)| \le C_2|g_2(n)|$ với mọi $n > n_1$ và mọi $n > n_2$. Do đó

$$\begin{aligned}|(f_1 + f_2)(n)| &= |f_1(n) + f_2(n)| \le |f_1(n)| + |f_2(n)| \\ &\le C_1|g_1(n)| + C_2|g_2(n)| \le (C_1 + C_2)g(n)\end{aligned}$$

với mọi $n > n_0 = \max(n_1, n_2)$, ở đây $C = C_1 + C_2$ và $g(n) = \max(|g_1(n)|, |g_2(n)|)$.

$$\begin{aligned}|(f_1 f_2)(n)| &= |f_1(n)||f_2(n)| \le C_1|g_1(n)|C_2|g_2(n)| \\ &\le C_1 C_2|(g_1 g_2)(n)|,\end{aligned}$$

với mọi $n > n_0 = \max(n_1, n_2)$.

Định nghĩa 5.2. Nếu một thuật toán có độ phức tạp là $f(n)$ với $f(n) = O(g(n))$ thì ta cũng nói *thuật toán có độ phức tạp* $O(g(n))$.

Nếu có hai thuật toán giải cùng một bài toán, thuật toán 1 có độ phức tạp $O(g_1(n))$, thuật toán 2 có độ phức tạp $O(g_2(n))$, mà $g_1(n)$ có cấp thấp hơn $g_2(n)$, thì ta nói rằng thuật toán 1 hữu hiệu hơn (hay nhanh hơn) thuật toán 2.

5.3.3. Đánh giá độ phức tạp của một thuật toán

A. Thuật toán tìm kiếm tuyến tính.

Số các phép so sánh được dùng trong thuật toán này cũng sẽ được xem như thước đo độ phức tạp thời gian của nó. Ở mỗi một bước của vòng lặp trong thuật toán, có hai phép so sánh được thực hiện: một để xem đã tới cuối bảng chưa và một để so sánh phần tử x với một số hạng của bảng. Cuối cùng còn một phép so sánh nữa làm ở ngoài vòng lặp. Do đó, nếu $x = a_i$, thì đã có $2i + 1$ phép so sánh được sử dụng. Số phép so sánh nhiều nhất, $2n + 2$, đòi hỏi phải được sử dụng khi phần tử x không có mặt trong bảng. Từ đó, thuật toán tìm kiếm tuyến tính có độ phức tạp là $O(n)$.

B. Thuật toán tìm kiếm nhị phân.

Để đơn giản, ta giả sử rằng có $n = 2^k$ phần tử trong bảng liệt kê $a_1, a_2, ..., a_n$, với k là số nguyên không âm (nếu n không phải là lũy thừa của 2, ta có thể xem bảng là một phần của bảng gồm 2^{k+1} phần tử, trong đó k là số nguyên nhỏ nhất sao cho $n < 2^{k+1}$).

Ở mỗi giai đoạn của thuật toán vị trí của số hạng đầu tiên i và số hạng cuối cùng j của bảng con hạn chế tìm kiếm ở giai đoạn đó được so sánh để xem bảng con này còn nhiều hơn một phần tử hay không. Nếu $i < j$, một phép so sánh sẽ được làm để xác định x có lớn hơn số hạng ở giữa của bảng con hạn chế hay không. Như vậy ở mỗi giai đoạn, có sử dụng hai phép so sánh. Khi trong bảng chỉ còn một phần tử, một phép so sánh sẽ cho chúng ta biết rằng không còn một phần tử nào thêm nữa và một phép so sánh nữa cho biết số hạng đó có phải là x hay không. Tóm lại cần phải có nhiều nhất $2k + 2 = 2\log_2 n + 2$ phép so sánh để thực hiện phép tìm kiếm nhị phân (nếu n không phải là lũy thừa của 2, bảng gốc sẽ được mở rộng tới bảng có 2^{k+1} phần tử, với $k = [\log_2 n]$ và sự tìm kiếm đòi hỏi phải thực hiện nhiều nhất $2[\log_2 n] + 2$ phép so sánh). Do đó thuật toán tìm kiếm nhị phân có độ phức tạp là $O(\log_2 n)$. Từ sự phân tích ở trên suy ra rằng thuật toán tìm kiếm nhị phân, ngay cả trong trường hợp xấu nhất, cũng hiệu quả hơn thuật toán tìm kiếm tuyến tính.

C. Chú ý.

Một điều quan trọng cần phải biết là máy tính phải cần bao lâu để giải xong một bài toán. Thí dụ, nếu một thuật toán đòi hỏi 10 giờ, thì có thể còn đáng chi phí thời gian máy tính đòi hỏi để giải bài toán đó. Nhưng nếu một thuật toán đòi hỏi 10 tỉ năm để giải một bài toán, thì thực hiện thuật toán đó sẽ là một điều phi lý. Một trong những hiện tượng lý thú nhất của công nghệ hiện đại là sự tăng ghê gớm của tốc độ và lượng bộ nhớ trong máy tính. Một nhân tố quan trọng khác làm giảm thời gian cần thiết để giải một bài toán là sự xử lý song song - đây là kỹ thuật thực hiện đồng thời các dãy phép tính. Do sự tăng tốc độ tính toán và dung lượng bộ nhớ của máy tính, cũng như nhờ việc dùng các thuật toán lợi dụng được ưu thế của kỹ thuật xử lý song song, các bài toán vài năm trước đây được xem là không thể giải được, thì bây giờ có thể giải bình thường.

D. Các thuật ngữ thường dùng cho độ phức tạp thuật toán.

Độ phức tạp	**Thuật ngữ**
$O(1)$	Độ phức tạp hằng số
$O(\log n)$	Độ phức tạp lôgarit
$O(n)$	Độ phức tạp tuyến tính
$O(n\log n)$	Độ phức tạp $n\log n$
$O(n^b)$	Độ phức tạp đa thức
$O(b^n)(b > 1)$	Độ phức tạp hàm mũ
$O(n!)$	Độ phức tạp giai thừa

E. Thời gian máy tính được dùng bởi một thuật toán.

Cỡ BT	Các phép tính bit được sử dụng					
n	$\log n$	N	$n \log n$	n^2	2^n	$n!$
10	3.10^{-9} s	10^{-8} s	3.10^{-8} s	10^{-7} s	10^{-6} s	3.10^{-3} s
10^2	7.10^{-9} s	10^{-7} s	7.10^{-7} s	10^{-5} s	4.10^{13}y	*
10^3	$1,0.10^{-8}$ s	10^{-6} s	1.10^{-5} s	10^{-3} s	*	*
10^4	$1,3.10^{-8}$ s	10^{-5} s	1.10^{-4} s	10^{-1} s	*	*
10^5	$1,7.10^{-8}$ s	10^{-4} s	2.10^{-3} s	10 s	*	*
10^6	2.10^{-8} s	10^{-3} s	2.10^{-2} s	17 p	*	*

5.4 Số nguyên và thuật toán

5.4.1. Thuật toán Euclide

Phương pháp tính ước chung lớn nhất của hai số bằng cách dùng phân tích các số nguyên đó ra thừa số nguyên tố là không hiệu quả. Lý do là ở chỗ thời gian phải tiêu tốn cho sự phân tích đó. Dưới đây là phương pháp hiệu quả hơn để tìm ước số chung lớn nhất, gọi là *thuật toán Euclide*. Thuật toán này đã biết từ thời cổ đại. Nó mang tên nhà toán học cổ Hy lạp Euclide, người đã mô tả thuật toán này trong cuốn sách "*Những yếu tố*" nổi tiếng của ông. Thuật toán Euclide dựa vào 2 mệnh đề sau đây.

Mệnh đề 5.2 (Thuật toán chia). *Cho a và b là hai số nguyên và $b \neq 0$. Khi đó tồn tại duy nhất hai số nguyên q và r sao cho*

$$a = bq + r, \quad 0 \leq r < |b|$$

. Trong đẳng thức trên, b được gọi là số chia, a được gọi là số bị chia, q được gọi là thương số và r được gọi là số dư.

Khi b là nguyên dương, ta ký hiệu số dư r trong phép chia a cho b là $a \mod b$.

Mệnh đề 5.3. *Cho $a = bq + r$, trong đó a, b, q, r là các số nguyên. Khi đó*

$$UCLN(a, b) = UCLN(b, r).$$

(Ở đây $UCLN(a, b)$ để chỉ ước chung lớn nhất của a và b.)

Giả sử a và b là hai số nguyên dương với $a \geq b$. Đặt $r_0 = a$ và $r_1 = b$. Bằng cách áp dụng liên tiếp thuật toán chia, ta tìm được

$$\begin{aligned}
&r_0 = r_1 q_1 + r_2, && 0 \leq r_2 < r_1\\
&r_1 = r_2 q_2 + r_3, && 0 \leq r_3 < r_2\\
&\dots && \dots\\
&r_{n-2} = r_{n-1} q_{n-1} + r_n, && 0 \leq r_n < r_{n-1}\\
&r_{n-1} = r_n q_n
\end{aligned}$$

Cuối cùng, số dư 0 sẽ xuất hiện trong dãy các phép chia liên tiếp, vì dãy các số dư

$$a = r_0 > r_1 > r_2 > ... \geq 0$$

không thể chứa quá a số hạng được. Hơn nữa, từ Mệnh đề 2 ở trên ta suy ra:

$UCLN(a, b) = UCLN(r_0, r_1) = UCLN(r_1, r_2) = ... = UCLN(r_{n-2}, r_{n-1}) = UCLN(r_{n-1}, r_n) = r_n$.

Do đó, ước chung lớn nhất là số dư khác không cuối cùng trong dãy các phép chia.

Ví dụ 5.8. Dùng thuật toán Euclide tìm UCLN(414, 662).

Lời giải.

$$\begin{aligned}
662 &= 441.1 + 248\\
414 &= 248.1 + 166\\
248 &= 166.1 + 82\\
166 &= 82.2 + 2\\
82 &= 2.41.
\end{aligned}$$

Do đó, $UCLN(414, 662) = 2$.

Thuật toán Euclide được viết dưới dạng giả mã như sau

Thuật toán 5.6. Tìm ước chung lớn nhất

```
procedure UCLN (a, b: positive integer)
   x := a
   y := b
   while y ≠ 0
      begin
         r := x mod y
         x := y
         y := r
      end
//UCLN (a, b) là x
```

Trong thuật toán trên, các giá trị ban đầu của x và y tương ứng là a và b. Ở mỗi giai đoạn của thủ tục, x được thay bằng y và y được thay bằng $x \mod y$. Quá trình này được lặp lại chừng nào $y \neq 0$. Thuật toán sẽ ngừng khi $y = 0$ và giá trị của x ở điểm này, đó là số dư khác không cuối cùng trong thủ tục, cũng chính là ước chung lớn nhất của a và b.

5.4.2. Biểu diễn các số nguyên

Mệnh đề 5.4. *Cho b là một số nguyên dương lớn hơn 1. Khi đó nếu n là một số nguyên dương, nó có thể được biểu diễn một cách duy nhất dưới dạng:*

$$n = a_k b^k + a_{k-1} b^{k-1} + \cdots + a_1 b + a_0.$$

Ở đây k là một số tự nhiên, $a_0, a_1, ..., a_k$ là các số tự nhiên nhỏ hơn b và $a_k \neq 0$.

Biểu diễn của n được cho trong Mệnh đề 5.4 được gọi là khai triển của n theo cơ số b, ký hiệu là $(a_k a_{k-1} ... a_1 a_0)_b$. Bây giờ ta sẽ mô tả thuật toán xây dựng khai triển cơ số b của số nguyên n bất kỳ. Trước hết ta chia n cho b để được thương và số dư, tức là

$$n = bq_0 + a_0, 0 \leq a_0 < b.$$

Số dư a_0 chính là chữ số đứng bên phải cùng trong khai triển cơ số b của n. Tiếp theo chia q_0 cho b, ta được

$$q_0 = bq_1 + a_1, 0 \leq a_1 < b.$$

Số dư a_1 chính là chữ số thứ hai tính từ bên phải trong khai triển cơ số b của n. Tiếp tục quá trình này, bằng cách liên tiếp chia các thương cho b ta sẽ được các chữ số tiếp theo trong khai triển cơ số b của n là các số dư tương ứng. Quá trình này sẽ kết thúc khi ta nhận được một thương bằng 0.

Ví dụ 5.9. Tìm khai triển cơ số 8 của $(12345)_{10}$.

Lời giải.

$$\begin{aligned}12345 &= 8.1543 + 1\\ 1543 &= 8.192 + 7\\ 192 &= 8.24 + 0\\ 24 &= 8.3 + 0\\ 3 &= 8.0 + 3.\end{aligned}$$

Do đó, $(12345)_{10} = (30071)_8$.

Đoạn giả mã sau biểu diễn thuật toán tìm khai triển cơ số b của số nguyên n.

Thuật toán 5.7. khai triển theo cơ số b

procedure khai_trien_theo_co_so_b (n: positive integer)
  $q := n$
  $k := 0$
  while $q \neq 0$
    begin
    $a_k := q \mod b$
    $q := \left[\frac{q}{b}\right]$
    $k := k + 1$
    end

5.4.3. Thuật toán cho các phép tính số nguyên

Các thuật toán thực hiện các phép tính với những số nguyên khi dùng các khai triển nhị phân của chúng là cực kỳ quan trọng trong số học của máy tính. Ta sẽ mô tả ở đây các thuật toán cộng và nhân hai số nguyên trong biểu diễn nhị phân. Ta cũng sẽ phân tích độ phức tạp tính toán của các thuật toán này thông qua số các phép toán bit thực sự được dùng. Giả sử khai triển nhị phân của hai số nguyên dương a và b là

$$a = (a_{n-1}a_{n-2}...a_1a_0)_2 \text{ và } b = (b_{n-1}b_{n-2}...b_1b_0)_2$$

sao cho a và b đều có n bit (đặt các bit 0 ở đầu mỗi khai triển đó, nếu cần).

1) Phép cộng. Xét bài toán cộng hai số nguyên viết ở dạng nhị phân. Thủ tục thực hiện phép cộng có thể dựa trên phương pháp thông thường là cộng cặp chữ số nhị phân với nhau (có nhớ) để tính tổng của hai số nguyên.

Để cộng a và b, trước hết cộng hai bit ở phải cùng của chúng, tức là:

$$a_0 + b_0 = c_0.2 + s_0.$$

Ở đây s_0 là bit phải cùng trong khai triển nhị phân của $a + b$, c_0 là số nhớ, nó có thể bằng 0 hoặc 1. Sau đó ta cộng hai bit tiếp theo và số nhớ

$$a_1 + b_1 + c_0 = c_1.2 + s_1.$$

Ở đây s_1 là bit tiếp theo (tính từ bên phải) trong khai triển nhị phân của $a + b$ và c_1 là số nhớ. Tiếp tục quá trình này bằng cách cộng các bit tương ứng trong hai khai triển nhị phân và số nhớ để xác định bit tiếp sau tính từ bên phải trong khai triển nhị phân của tổng $a + b$. Ở giai đoạn cuối cùng, cộng a_{n-1}, b_{n-1} và c_{n-2} để nhận được $c_{n-1}.2 + s_{n-1}$. Bit đứng đầu của tổng là $s_n = c_{n-1}$. Kết quả, thủ tục này tạo ra được khai triển nhị phân của tổng, cụ thể là $a + b = (s_n s_{n-1} s_{n-2}...s_1 s_0)_2$.

Ví dụ 5.10. Tìm tổng của $a = (11011)_2$ và $b = (10110)_2$.

Lời giải.

$$\begin{aligned}
a_0 + b_0 &= 1 + 0 = 0.2 + 1(c_0 = 0, s_0 = 1),\\
a_1 + b_1 + c_0 &= 1 + 1 + 0 = 1.2 + 0(c_1 = 1, s_1 = 0),\\
a_2 + b_2 + c1 &= 0 + 1 + 1 = 1.2 + 0(c_2 = 1, s_2 = 0),\\
a_3 + b_3 + c_2 &= 1 + 0 + 1 = 1.2 + 0(c_3 = 1, s_3 = 0),\\
a_4 + b_4 + c_3 &= 1 + 1 + 1 = 1.2 + 1(s_5 = c_4 = 1, s_4 = 1).
\end{aligned}$$

Do đó, $a + b = (110001)_2$.

Thuật toán cộng có thể được mô tả bằng cách dùng đoạn giả mã như sau.

Thuật toán 5.8. Phép cộng hai số nhị phân

---

procedure cong $(a, b$: positive integer)
  $c := 0$
  for $j := 0$ **to** $n - 1$
    begin
    $d := \left\lfloor \dfrac{a_j + b_j + c}{2} \right\rfloor$
    $s_j := a_j + b_j + c - 2d$
    $c := d$
    end
  $s_n := c$
//khai triển nhị phân của tổng là $(s_n s_{n-1} ... s_1 s_0)_2$

Tổng hai số nguyên được tính bằng cách cộng liên tiếp các cặp bit và khi cần phải cộng cả số nhớ nữa. Cộng một cặp bit và số nhớ đòi ba hoặc ít hơn phép cộng các bit. Như vậy, tổng số các phép cộng bit được sử dụng nhỏ hơn ba lần số bit trong khai triển nhị phân. Do đó, độ phức tạp của thuật toán này là $O(n)$.

2) Phép nhân. Xét bài toán nhân hai số nguyên viết ở dạng nhị phân. Thuật toán thông thường tiến hành như sau. Dùng luật phân phối, ta có

$$ab = a \sum_{j=0}^{n-1} b_j 2^j = \sum_{j=0}^{n-1} a(b_j 2^j).$$

Ta có thể tính ab bằng cách dùng phương trình trên. Trước hết, ta thấy rằng $ab_j = a$ nếu $b_j = 1$ và $ab_j = 0$ nếu $b_j = 0$. Mỗi lần ta nhân một số hạng với 2 là ta dịch khai triển nhị phân của nó một chỗ về phía trái bằng cách thêm một số không vào cuối khai triển nhị phân của nó. Do đó, ta có thể nhận được $(ab_j)2^j$ bằng cách dịch khai triển nhị phân của ab_j đi j chỗ về phía trái, tức là thêm j số không vào cuối khai triển nhị phân của nó. Cuối cùng, ta sẽ nhận được tích ab bằng cách cộng n số nguyên $ab_j.2^j$ với $j = 0, 1, ..., n-1$.

Ví dụ 5.11. Tìm tích của $a = (110)_2$ và $b = (101)_2$.

Lời giải. Ta có

$$\begin{aligned}
ab_0.2^0 &= (110)_2.1.2^0 = (110)_2, \\
ab_1.2^1 &= (110)_2.0.2^1 = (0000)_2, \\
ab_2.2^2 &= (110)_2.1.2^2 = (11000)_2.
\end{aligned}$$

Để tìm tích, hãy cộng $(110)_2, (0000)_2$ và $(11000)_2$. Từ đó ta có $ab = (11110)_2$.

Thủ tục trên được mô tả bằng đoạn giả mã sau

Thuật toán 5.9. Phép nhân hai số nhị phân

```
procedure nhan (a, b: positive integer)
  for j := 0 to n − 1
  begin
      if b_j = 1 then c_j := a    //được dịch đi jchỗ
         else c_j := 0
  end
//c_0, c_1, ..., c_{n−1} là các tích riêng phần
  p := 0
  for j := 0 to n − 1
      p := p + c_j
//p là giá trị của tích ab
```

Thuật toán trên tính tích của hai số nguyên a và b bằng cách cộng các tích riêng phần $c_0, c_1, c_2, ..., c_{n-1}$. Khi $b_j = 1$, ta tính tích riêng phần c j bằng cách dịch khai triển nhị phân của a đi j bit.

Khi $b_j = 0$ thì không cần có dịch chuyển nào vì $c_j = 0$. Do đó, để tìm tất cả n số nguyên $ab_j.2^j$ với $j = 0, 1, ..., n-1$, đòi hỏi tối đa là

$$0 + 1 + 2 + ... + n - 1 = \frac{n(n-1)}{2}.$$

phép dịch chỗ. Vì vậy, số các dịch chuyển chỗ đòi hỏi là $O(n^2)$.

Để cộng các số nguyên ab_j từ $j = 0$ đến $n-1$, đòi hỏi phải cộng một số nguyên n bit, một số nguyên $n+1$ bit, ... và một số nguyên $2n$ bit. Ta đã biết rằng mỗi phép cộng đó đòi hỏi $O(n)$ phép cộng bit. Do đó, độ phức tạp của thuật toán này là $O(n^2)$.

5.5 Thuật toán đệ quy

5.5.1. Khái niệm đệ quy

Đôi khi chúng ta có thể quy việc giải bài toán với tập các dữ liệu đầu vào xác định về việc giải cùng bài toán đó nhưng với các giá trị đầu vào nhỏ hơn. Chẳng hạn, bài toán tìm UCLN của hai số a, b với $a > b$ có thể rút gọn về bài toán tìm UCLN của hai số nhỏ hơn, $a \mod b$ và b. Khi việc rút gọn như vậy thực hiện được thì lời giải bài toán ban đầu có thể tìm được bằng một dãy các phép rút gọn cho tới những trường hợp mà ta có thể dễ dàng nhận được lời giải của bài toán. Ta sẽ thấy rằng các thuật toán rút gọn liên tiếp bài toán ban đầu tới bài toán có dữ liệu đầu vào nhỏ hơn, được áp dụng trong một lớp rất rộng các bài toán.

Định nghĩa 5.3. Một thuật toán được gọi là đệ quy nếu nó giải bài toán bằng cách rút gọn liên tiếp bài toán ban đầu tới bài toán cũng như vậy nhưng có dữ liệu đầu vào nhỏ hơn.

Ví dụ 5.12. Tìm thuật toán đệ quy tính giá trị a^n với a là số thực khác không và n là số nguyên không âm.

Lời giải. Ta xây dựng thuật toán đệ quy nhờ định nghĩa đệ quy của a^n, đó là $a^{n+1} = a.a^n$ với $n > 0$ và khi $n = 0$ thì $a^0 = 1$. Vậy để tính a^n ta quy về các trường hợp có số mũ n nhỏ hơn, cho tới khi $n = 0$.

Thuật toán 5.10. Tính lũy thừa theo đệ quy

```
procedure power(a: số thực khác không; n: số nguyên không âm)
  if n = 0 then power(a,n) := 1
     else power(a,n) := a * power(a,n − 1)
```

Ví dụ 5.13. Tìm thuật toán đệ quy tính UCLN của hai số nguyên a, b không âm và $a > b$.

Lời giải.

Thuật toán 5.11. Tìm ước chung lớn nhất theo đệ quy

```
procedure UCLN (a,b: các số nguyên không âm, a > b)
  if b = 0 then UCLN(a,b) := a
     else UCLN(a,b) := UCLN(amodb,b)
```

Ví dụ 5.14. Hãy biểu diễn thuật toán tìm kiếm tuyến tính như một thủ tục đệ quy.

Lời giải. Để tìm x trong dãy tìm kiếm $a_1, a_2, ..., a_n$ trong bước thứ i của thuật toán ta so sánh x với a_i. Nếu x bằng a_i thì i là vị trí cần tìm, ngược lại thì việc tìm kiếm được quy về dãy có số phần tử ít hơn, cụ thể là dãy $a_{i+1}, ..., a_n$. Thuật toán tìm kiếm có dạng thủ tục đệ quy như sau.

Cho $search(i, j, x)$ là thủ tục tìm số x trong dãy $a_i, a_{i+1}, ..., a_j$. Dữ liệu đầu vào là bộ ba $(1, n, x)$. Thủ tục sẽ dừng khi số hạng đầu tiên của dãy còn lại là x hoặc là khi dãy còn lại chỉ có một phần tử khác x. Nếu x không là số hạng đầu tiên và còn có các số hạng khác thì lại áp dụng thủ tục này, nhưng dãy tìm kiếm ít hơn một phần tử nhận được bằng cách xóa đi phần tử đầu tiên của dãy tìm kiếm ở bước vừa qua.

Thuật toán 5.12. Tìm tuyến tính theo đệ quy

```
procedure search (i,j,x)
  if a_i = x then loacation := i
      else if i = j then loacation := 0
             else search (i + 1,j,x)
```

Ví dụ 5.15. Hãy xây dựng phiên bản đệ quy của thuật toán tìm kiếm nhị phân.

Lời giải. Giả sử ta muốn định vị x trong dãy $a_1, a_2, ..., a_n$ bằng tìm kiếm nhị phân. Trước tiên ta so sánh x với số hạng giữa $a_{[(n+1)/2]}$. Nếu chúng bằng nhau thì thuật toán kết thúc, nếu không ta chuyển sang tìm kiếm trong dãy ngắn hơn, nửa đầu của dãy nếu x nhỏ hơn giá trị giữa của của dãy xuất phát, nửa sau nếu ngược lại. Như vậy ta rút gọn việc giải bài toán tìm kiếm về việc giải cũng bài toán đó nhưng trong dãy tìm kiếm có độ dài lần lượt giảm đi một nửa.

Thuật toán 5.13. Tìm kiếm nhị phân đệ quy

```
procedure binarysearch (i, j, x)
m := [(i + j)/2]
if x = a_m then loacation := m
 else if (x < a_m and i < m) then binarysearch(x, i, m − 1)
 else if (x > a_m and j > m) then binarysearch(x, m + 1, j)
      else loacation := 0
```

5.5.2. Đệ quy và lặp

Ví dụ 5.16. Thủ tục đệ quy sau đây cho ta giá trị của $n!$ với n là số nguyên dương.

Thuật toán 5.14. Tìm giai thừa theo đệ quy

```
procedure factorial (n : positive integer)
  if n = 1 then factorial(n) := 1
       else factorial(n) := n * factorial(n − 1)
```

Có cách khác tính hàm giai thừa của một số nguyên từ định nghĩa đệ quy của nó. Thay cho việc lần lượt rút gọn việc tính toán cho các giá trị nhỏ hơn, ta có thể xuất phát từ giá trị của hàm tại 1và lần lượt áp dụng định nghĩa đệ quy để tìm giá trị của hàm tại các số nguyên lớn dần. Đó là thủ tục lặp.

Thuật toán 5.15. Tìm giai thừa theo lặp

procedure $iterativefactorial$ (n : positive integer)
  $x := 1$
  for $i := 1$ **to** n
    $x := i * x$
$//x$ là $n!$

Thông thường để tính một dãy các giá trị được định nghĩa bằng đệ quy, nếu dùng phương pháp lặp thì số các phép tính sẽ ít hơn là dùng thuật toán đệ quy (trừ khi dùng các máy đệ quy chuyên dụng). Ta sẽ xem xét bài toán tính số hạng thứ n của dãy Fibonacci.

Thuật toán 5.16. Tính số Fibonacci theo đệ quy

procedure $fibonacci$ (n : positive integer)
if $n = 0$ **the** $fibonacci(n) := 0$
  else if $n = 1$ **then** $fibonacci(n) := 1$
    else $fibonacci(n) := fibonacci(n-1) + fibonacci(n-2)$

Theo thuật toán này, để tìm f_n ta biểu diễn $f_n = f_{n-1} + f_{n-2}$. Sau đó thay thế cả hai số này bằng tổng của hai số Fibonacci bậc thấp hơn, cứ tiếp tục như vậy cho tới khi f_0 và f_1 xuất hiện thì được thay bằng các giá trị của chúng theo định nghĩa. Do đó để tính f_n cần $f_{n+1} - 1$ phép cộng.

Bây giờ ta sẽ tính các phép toán cần dùng để tính f_n khi sử dụng phương pháp lặp. Thủ tục này khởi tạo x là $f_0 = 0$ và y là $f_1 = 1$. Khi vòng lặp được duyệt qua tổng của x và y được gán cho biến phụ z. Sau đó x được gán giá trị của y và y được gán giá trị của 4. Vậy sau khi đi qua vòng lặp lần 1, ta có $x = f_1$ và $y = f_0 + f_1 = f_2$. Khi qua vòng lặp lần $n-1$ thì $x = f_{n-1}$. Như vậy chỉ có $n-1$ phép cộng được dùng để tìm f_n khi $n > 1$.

Thuật toán 5.17. Tính số Fibonacci theo đệ quy

procedure $Iterativefibonacci$ (n : positive integer)
if $n = 0$ **then** $y := 0$
else

```
begin
    x := 0; y := 1
    for i := 1 to n − 1
    begin
        z := x + y
        x := y; y := z
    end
end
//y là số Fibonacci thứ n
```

Ta đã chỉ ra rằng số các phép toán dùng trong thuật toán đệ quy nhiều hơn khi dùng phương pháp lặp. Tuy nhiên đôi khi người ta vẫn thích dùng thủ tục đệ quy hơn ngay cả khi nó tỏ ra kém hiệu quả so với thủ tục lặp. Đặc biệt, có những bài toán chỉ có thể giải bằng thủ tục đệ quy mà không thể giải bằng thủ tục lặp.

5.6 Bài tập

▷ **5.1.** Tìm một số nguyên n nhỏ nhất sao cho $f(x)$ là $O(x^n)$ đối với các hàm $f(x)$ sau

a) $f(x) = 2x^3 + x^2 \log x$.

b) $f(x) = 2x^3 + (\log x)^4$.

c) $f(x) = \dfrac{x^4 + x^2 + 1}{x^3 + 1}$.

d) $f(x) = \dfrac{x^5 + 5\log x}{x^4 + 1}$.

▷ **5.2.** Chứng minh rằng

a) $x^2 + 4x + 7$ là $O(x^3)$, nhưng x^3 không là $O(x^2 + 4x + 17)$.

b) $x\log x$ là $O(x^2)$, nhưng x^2 không là $O(x\log x)$.

▷ **5.3.** Cho một đánh giá big-O đối với các hàm cho dưới đây. Đối với hàm $g(x)$ trong đánh giá $f(x)$ là $O(g(x))$, hãy chọn hàm đơn giản có bậc thấp nhất.

a) $n\log(n^2 + 1) + n^2\log n$.

b) $(n\log n + 1)^2 + (\log n + 1)(n^2 + 1)$.

c) $n^{2^n} + n^{n^2}$.

▷ **5.4.** Cho H_n là số điều hoà thứ n

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$$

Chứng minh rằng H_n là $O(\log n)$.

▷ **5.5.** Lập một thuật toán tính tổng tất cả các số nguyên trong một bảng.

▷ **5.6.** Lập thuật toán tính x_n với x là một số thực và n là một số nguyên.

▷ **5.7.** Mô tả thuật toán chèn một số nguyên x vào vị trí thích hợp trong dãy các số nguyên $a_1, a_2, ..., a_n$ xếp theo thứ tự tăng dần.

▷ **5.8.** Tìm thuật toán xác định vị trí gặp đầu tiên của phần tử lớn nhất trong bảng liệt kê các số nguyên, trong đó các số này không nhất thiết phải khác nhau.

▷ **5.9.** Tìm thuật toán xác định vị trí gặp cuối cùng của phần tử nhỏ nhất trong bảng liệt kê các số nguyên, trong đó các số này không nhất thiết phải khác nhau.

▷ **5.10.** Mô tả thuật toán đếm số các số 1 trong một xâu bit bằng cách kiểm tra mỗi bit của xâu để xác định nó có là bit 1 hay không.

▷ **5.11.** *Thuật toán tìm kiếm tam phân.* Xác định vị trí của một phần tử trong một bảng liệt kê các số nguyên theo thứ tự tăng dần bằng cách tách liên tiếp bảng liệt kê đó thành ba bảng liệt kê con có kích thước bằng nhau (hoặc gần bằng nhau nhất có thể được) và giới hạn việc tìm kiếm trong một bảng liệt kê con thích hợp. Hãy chỉ rõ các bước của thuật toán đó.

▷ **5.12.** Lập thuật toán tìm trong một dãy các số nguyên số hạng đầu tiên bằng một số hạng nào đó đứng trước nó trong dãy.

▷ **5.13.** Lập thuật toán tìm trong một dãy các số nguyên tất cả các số hạng lớn hơn tổng tất cả các số hạng đứng trước nó trong dãy.

▷ **5.14.** Cho đánh giá big-O đối với số các phép so sánh được dùng bởi thuật toán trong Bài tập 10.

▷ **5.15.** Đánh giá độ phức tạp của thuật toán tìm kiếm tam phân được cho trong Bài tập 11.

▷ **5.16.** Đánh giá độ phức tạp của thuật toán trong Bài tập 12.

▷ **5.17.** Mô tả thuật toán tính hiệu của hai khai triển nhị phân.

▷ **5.18.** Lập một thuật toán để xác định $a > b, a = b$ hay $a < b$ đối với hai số nguyên a và b ở dạng khai triển nhị phân.

▷ **5.19.** Đánh giá độ phức tạp của thuật toán tìm khai triển theo cơ số b của số nguyên n qua số các phép chia được dùng.

▷ **5.20.** Hãy cho thuật toán đệ quy tìm tổng n số nguyên dương lẻ đầu tiên.

▷ **5.21.** Hãy cho thuật toán đệ quy tìm số cực đại của tập hữu hạn các số nguyên.

▷ **5.22.** Mô tả thuật toán đệ quy tìm $x^n \mod m$ với n, x, m là các số nguyên dương.

▷ **5.23.** Hãy nghĩ ra thuật toán đệ quy tính a^{2^n} trong đó a là một số thực và n là một số nguyên dương.

▷ **5.24.** Hãy nghĩ ra thuật toán đệ quy tìm số hạng thứ n của dãy được xác định như sau

$$a_0 = 1, a_1 = 2 \text{ và } a_n = a^{n-1}a^{n-2} \text{ với } n = 2, 3, 4, ...$$

▷ **5.25.** Thuật toán đệ quy hay thuật toán lặp tìm số hạng thứ n của dãy trong Bài tập 24 là có hiệu quả hơn?

Chương

6 KHÁI NIỆM ĐỒ THỊ

Lý thuyết đồ thị là một ngành khoa học được phát triển từ lâu nhưng lại có nhiều ứng dụng hiện đại. Những ý tưởng cơ bản của nó được đưa ra từ thế kỷ 18 bởi nhà toán học Thụy Sĩ tên là Leonhard Euler. Ông đã dùng đồ thị để giải quyết bài toán 7 chiếc cầu Konigsberg nổi tiếng.

Đồ thị cũng được dùng để giải các bài toán trong nhiều lĩnh vực khác nhau. Thí dụ, dùng đồ thị để xác định xem có thực hiện một mạch điện trên một bảng điện phẳng được không. Chúng ta cũng có thể phân biệt hai hợp chất hóa học có cùng công thức phân tử nhưng có cấu trúc khác nhau nhờ đồ thị. Chúng ta cũng có thể xác định xem hai máy tính có được nối với nhau bằng một đường truyền thông hay không nếu dùng mô hình đồ thị mạng máy tính. Đồ thị với các trọng số được gán cho các cạnh của nó có thể dùng để giải các bài toán như bài toán tìm đường đi ngắn nhất giữa hai thành phố trong một mạng giao thông. Chúng ta cũng có thể dùng đồ thị để lập lịch thi và phân chia kênh cho các đài truyền hình.

6.1 Định nghĩa và ví dụ

Đồ thị là một cấu trúc rời rạc gồm các đỉnh và các cạnh (vô hướng hoặc có hướng) nối các đỉnh đó. Người ta phân loại đồ thị tùy theo đặc tính và số các cạnh nối các cặp đỉnh của đồ thị. Nhiều bài toán thuộc những lĩnh vực rất khác nhau có thể giải được bằng mô hình đồ thị. Chẳng hạn người ta có thể dùng đồ thị để biểu diễn sự cạnh tranh các loài trong một môi trường sinh thái, dùng đồ thị để biểu diễn ai có ảnh hưởng lên ai trong một tổ chức nào đó, và cũng có thể dùng đồ thị để biểu diễn các kết cục của cuộc thi đấu thể thao. Chúng ta cũng có thể dùng đồ thị để giải các bài toán như bài toán tính số các tổ hợp khác nhau của các chuyến bay giữa hai thành phố trong một mạng hàng không, hay để giải bài toán đi tham quan tất cả các đường phố của một thành phố sao cho mỗi đường phố đi qua đúng một lần, hoặc bài toán tìm số các màu cần thiết để tô các vùng khác nhau của một bản đồ.

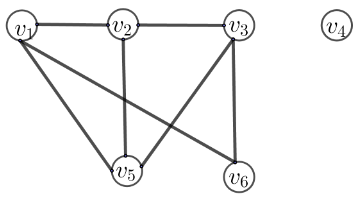
Trong đời sống, chúng ta thường gặp những sơ đồ, như sơ đồ tổ chức bộ máy, sơ đồ giao thông, sơ đồ hướng dẫn thứ tự đọc các chương trong một cuốn sách, ..., gồm những điểm biểu thị các đối tượng được xem xét (người, tổ chức, địa danh, chương mục sách, ...) và nối một số điểm với nhau bằng những đoạn thẳng (hoặc cong) hay những mũi tên, tượng trưng cho một quan hệ nào đó giữa các đối tượng. Đó là những thí dụ về đồ thị.

Định nghĩa 6.1. Một *đơn đồ thị* $\mathcal{G} = (V, E)$ gồm một tập khác rỗng V mà các phần tử của nó gọi là các đỉnh và một tập E mà các phần tử của nó gọi là các cạnh, đó là các cặp không có thứ tự của các đỉnh phân biệt.

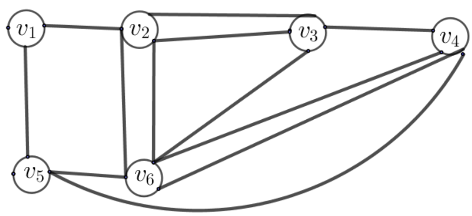
Định nghĩa 6.2. Một *đa đồ thị* $\mathcal{G} = (V, E)$ gồm một tập khác rỗng V mà các phần tử của nó gọi là các đỉnh và một họ E mà các phần tử của nó gọi là các cạnh, đó là các cặp không có thứ tự của các đỉnh phân biệt. Hai cạnh được gọi là cạnh bội hay song song nếu chúng cùng tương ứng với một cặp đỉnh.

Rõ ràng mỗi đơn đồ thị là đa đồ thị, nhưng không phải đa đồ thị nào cũng là đơn đồ thị.

Ví dụ 6.1. Hình 6.1 đơn đồ thị $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$,
$E = \{(v_1, v_2), (v_2, v_3), (v_3, v_6), (v_3, v_5), (v_2, v_5), (v_1, v_5), (v_1, v_6)\}$.
Hình 6.2 đa đồ thị $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$,
$E = \{(v_1, v_2), (v_1, v_5), (v_2, v_3), (v_2, v_6), (v_3, v_2), (v_3, v_4),$
$(v_3, v_6), (v_4, v_6), (v_4, v_5), (v_5, v_6), (v_6, v_2), (v_6, v_4)\}$.



Hình 6.1. Đơn đồ thị



Hình 6.2. Đa đồ thị

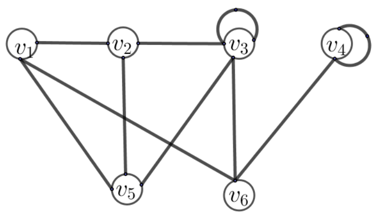
Định nghĩa 6.3. Một *giả đồ thị* $\mathcal{G} = (V, E)$ gồm một tập khác rỗng V mà các phần tử của nó gọi là các đỉnh và một họ E mà các phần tử của nó gọi là các cạnh, đó là các cặp không có thứ tự của các đỉnh (không nhất thiết là phân biệt).

Với $v \in V$, nếu $(v, v) \in E$ thì ta nói có một *khuyên* tại đỉnh v.

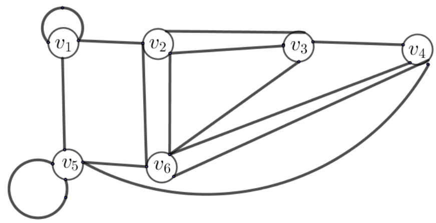
Tóm lại, giả đồ thị là loại đồ thị vô hướng tổng quát nhất vì nó có thể chứa các khuyên và các cạnh bội. Đa đồ thị là loại đồ thị vô hướng có thể chứa cạnh bội nhưng không thể có các khuyên, còn đơn đồ thị là loại đồ thị vô hướng không chứa cạnh bội hoặc các khuyên.

Ví dụ 6.2. Hình 6.3 đơn đồ thị $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$,
$E = \{(v_1, v_2), (v_2, v_3), (v_3, v_6), (v_3, v_5), (v_3, v_3), (v_2, v_5), (v_1, v_5),$
$(v_1, v_6), (v_4, v_6), (v_4, v_4)\}$.

Hình 6.4 đa đồ thị $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$,
$E = \{(v_1, v_1), (v_1, v_2), (v_1, v_5), (v_2, v_3), (v_2, v_6), (v_3, v_2), (v_3, v_4), (v_3, v_6), (v_4, v_6), (v_4, v_5), (v_5, v_5), (v_5, v_6), (v_6, v_2), (v_6, v_4)\}$.



Hình 6.3. Giả đơn đồ thị



Hình 6.4. Giả da đồ thị

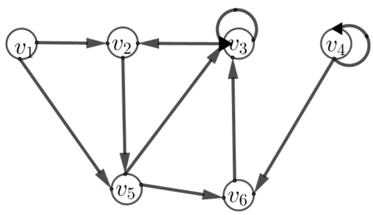
Định nghĩa 6.4. Một *đồ thị có hướng* $\mathcal{G} = (V, E)$ gồm một tập khác rỗng V mà các phần tử của nó gọi là các đỉnh và một tập E mà các phần tử của nó gọi là các cung, đó là các cặp có thứ tự của các phần tử thuộc V.

Định nghĩa 6.5. Một *đa đồ thị có hướng* $\mathcal{G} = (V, E)$ gồm một tập khác rỗng V mà các phần tử của nó gọi là các đỉnh và một họ E mà các phần tử của nó gọi là các cung, đó là các cặp có thứ tự của các phần tử thuộc V.

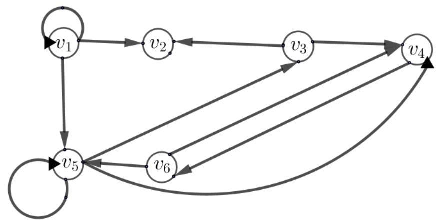
Định nghĩa 6.6. Đồ thị vô hướng nhận được từ đồ thị có hướng $\mathcal{G}$ bằng cách xoá bỏ các chiều mũi tên trên các cung được gọi là *đồ thị vô hướng nền* của $\mathcal{G}$.

Ví dụ 6.3. Hình 6.5 đơn đồ thị có hướng
$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$,
$E = \{(v_1, v_2), (v_2, v_5), (v_1, v_5), (v_3, v_2), (v_5, v_3), (v_3, v_3), (v_5, v_6), (v_6, v_3), (v_4, v_6), (v_4, v_4)\}$.
Hình 6.6 đa đồ thị có hướng $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$,
$E = \{(v_1, v_1), (v_1, v_2), (v_1, v_5), (v_3, v_2), (v_3, v_4), (v_4, v_6), (v_6, v_4), (v_6, v_5), (v_5, v_3), (v_5, v_4), (v_5, v_5)\}$.

Ví dụ 6.4. Đồ thị "lấn tổ" trong sinh thái học. Đồ thị được dùng 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



Hình 6.5. Giả đơn đồ thị



Hình 6.6. Giả da đồ thị

thể mô hình hóa bằng đồ thị "lấn tổ". 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 6.5. Đồ thị ảnh hưởng. Khi nghiên cứu tính cách của một nhóm người, ta thấy một số người có thể có ảnh hưởng lên suy nghĩ của những người khác. Đồ thị có hướng được gọi là đồ thị ảnh hưởng có thể dùng để mô hình bài toán này. Mỗi người của nhóm được biểu diễn bằng một đỉnh. 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*.

Ví dụ 6.6. Thi đấu vòng tròn. Một cuộc thi đấu thể thao trong đó mỗi đội đấu với mỗi đội khác đúng một lần gọi là đấu vòng tròn. Cuộc thi đấu như thế có thể được mô hình bằng một đồ thị có hướng trong đó mỗi đội là một đỉnh. Một cung đi từ đỉnh *a* đến đỉnh *b* nếu đội *a* thắng đội *b*.

Ví dụ 6.7. Phần mềm máy tính. Các chương trình máy tính có thể thi hành nhanh hơn bằng cách thi hành đồng thời một số câu lệnh nào đó. Điều quan trọng là không được thực hiện một câu lệnh đòi hỏi kết quả của câu lệnh khác chưa được thực hiện. Sự phụ thuộc của các câu lệnh vào các câu lệnh trước có thể biểu diễn bằng một đồ thị có hướng. Mỗi câu lệnh được biểu diễn bằng một đỉnh và có một cung từ một đỉnh tới một đỉnh khác nếu câu lệnh được biểu diễn bằng đỉnh thứ hai không thể thực hiện được trước khi câu lệnh được biểu diễn bằng đỉnh thứ nhất được thực hiện. Đồ thị này được gọi là đồ thị có ưu tiên trước sau.

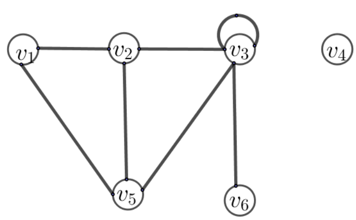
6.2 Bậc của đỉnh đồ thị

Định nghĩa 6.7. Hai đỉnh u và v trong đồ thị (vô hướng) $\mathcal{G} = (V, E)$ được gọi là *liền kề* nếu $(u, v) \in E$. Nếu $e = (u, v)$ thì e gọi là *cạnh liên thuộc* với các đỉnh u và v. Cạnh e cũng được gọi là *cạnh nối* các đỉnh u và v. Các đỉnh u và v gọi là các *điểm đầu mút* của cạnh e.

Định nghĩa 6.8. *Bậc* của đỉnh v trong đồ thị $\mathcal{G} = (V, E)$, ký hiệu $\deg(v)$, là số các cạnh liên thuộc với nó, riêng khuyên tại một đỉnh được tính hai lần cho bậc của nó.

Đỉnh v gọi là *đỉnh treo* nếu $\deg(v) = 1$ và gọi là *đỉnh cô lập* nếu $\deg(v) = 0$.

Ví dụ 6.8. Hình 6.7 bậc của các đỉnh là $\deg(v_1) = 2, \deg(v_2) = 3, \deg(v_3) = 5, \deg(v_4) = 0, \deg(v_5) = 3, \deg(v_6) = 1$.



Hình 6.7. Bậc đỉnh của đồ thị

Đỉnh v_4 là đỉnh cô lập và đỉnh v_6 là đỉnh treo.

Mệnh đề 6.1. *Cho đồ thị* $\mathcal{G} = (V, E)$. *Khi đó* $2|E| = \sum_{v \in V} \deg(v)$.

Chứng minh. Rõ ràng mỗi cạnh $e = (u, v)$ được tính một lần trong $\deg(u)$ và một lần trong $\deg(v)$. Từ đó suy ra tổng tất cả các bậc của các đỉnh bằng hai lần số cạnh.

Hệ quả 6.1. *Số đỉnh bậc lẻ của một đồ thị 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 $\mathcal{G} = (V, E)$. Khi đó

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

Vế trái 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 $|V_2|$ là một số chẵn.

Mệnh đề 6.2. *Trong một đơn đồ thị, luôn tồn tại hai đỉnh có cùng bậc.*

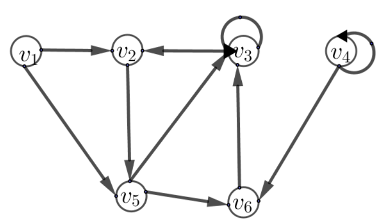
Chứng minh. Xét đơn đồ thị $\mathcal{G} = (V, E)$ có $|V| = n$. Khi đó phát biểu trên được đưa về bài toán: trong một phòng họp có n người, bao giờ cũng tìm được 2 người có số người quen trong số những người dự họp là như nhau (Đã chứng minh ở phần trước dùng nguyên lý Điricle).

Định nghĩa 6.9. Đỉnh u được gọi là nối tới v hay v được gọi là được nối từ u trong đồ thị có hướng $\mathcal{G}$ nếu (u, v) là một cung của $\mathcal{G}$. Đỉnh u gọi là *đỉnh đầu* và đỉnh v gọi là *đỉnh cuối* của cung này.

Định nghĩa 6.10. *Bậc vào* (*Bậc ra*) của đỉnh v trong đồ thị có hướng $\mathcal{G}$, ký hiệu $\deg_t(v)$ ($\deg_o(v)$), là số các cung có đỉnh cuối là v (số các cung có đỉnh đầu là v).

Ví dụ 6.9. Hình 6.8 đồ thị có hướng, bậc vào và bậc ra các đỉnh là

$$\begin{aligned}
&\deg_t(v_1) = 0, \deg_o(v_1) = 2,\\
&\deg_t(v_2) = 2, \deg_o(v_2) = 1,\\
&\deg_t(v_3) = 3, \deg_o(v_3) = 2,\\
&\deg_t(v_4) = 1, \deg_o(v_4) = 2,\\
&\deg_t(v_5) = 2, \deg_o(v_5) = 2,\\
&\deg_t(v_6) = 2, \deg_o(v_6) = 1.
\end{aligned}$$



Hình 6.8. Bậc vào và bậc ra đồ thị có hướng

Định nghĩa 6.11. Đỉnh có bậc vào và bậc ra cùng bằng 0 gọi là *đỉnh cô lập*. Đỉnh có bậc vào bằng 1 và bậc ra bằng 0 gọi là *đỉnh treo*, cung có đỉnh cuối là đỉnh treo gọi là *cung treo*.

Mệnh đề 6.3. *Cho $\mathcal{G} = (V, E)$ là một đồ thị có hướng. Khi đó*

$$\sum_{v \in V} \deg_t(v) = \sum_{v \in V} \deg_0(v) = |E|.$$

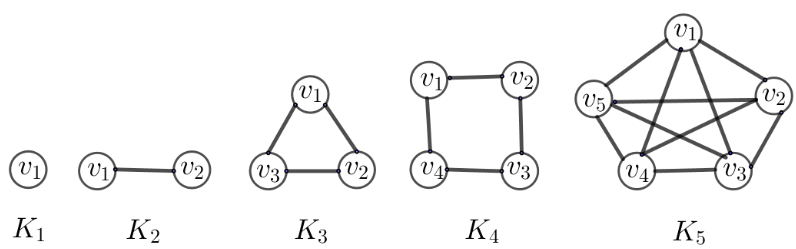
Chứng minh. Kết quả có ngay là vì mỗi cung được tính một lần cho đỉnh đầu và một lần cho đỉnh cuối.

6.3 Những đơn đồ thị đặc biệt

6.3.1. Đồ thị đầy đủ

Đồ thị đầy đủ n đỉnh, ký hiệu là K_n, là đơn đồ thị mà hai đỉnh phân biệt bất kỳ của nó luôn liền kề. Như vậy, K_n có $\dfrac{n(n-1)}{2}$ cạnh và mỗi đỉnh của K_n có bậc là $n-1$.

Ví dụ 6.10. Một số đồ thị đầy đủ ban đầu



Hình 6.9. Đồ thị đầy đủ

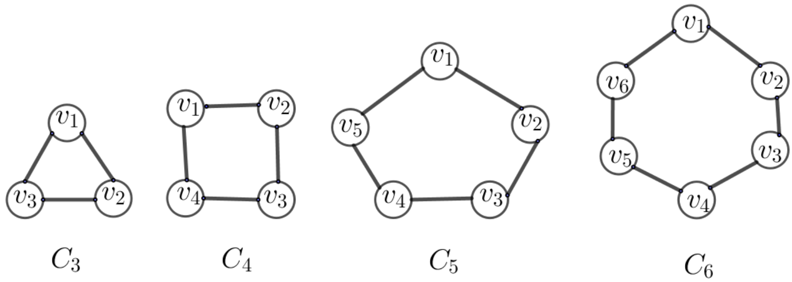
6.3.2. Đồ thị vòng

Đơn đồ thị n đỉnh $v_1, v_2, ..., v_n (n \geq 3)$ và n cạnh $(v_1, v_2), (v_2, v_3), ..., (v_{n-1}, v_n), (v_n, v_1)$ được gọi là *đồ thị vòng*, ký hiệu là C_n. Như vậy, mỗi đỉnh của C_n có bậc là 2.

Ví dụ 6.11. Một số đồ thị vòng ban đầu

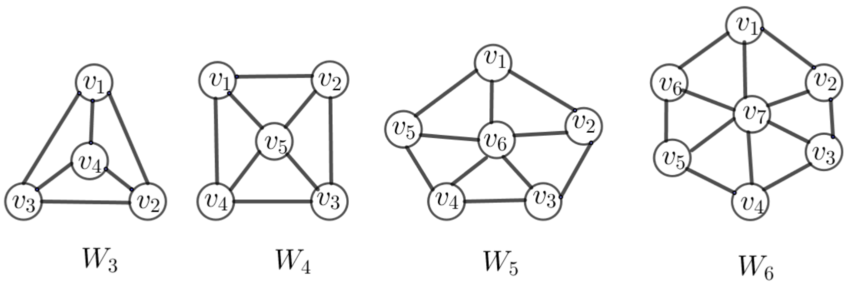
6.3.3. Đồ thị bánh xe

Từ đồ thị vòng C_n, thêm vào đỉnh v_{n+1} và các cạnh $(v_{n+1}, v_1), (v_{n+1}, v_2), ..., (v_{n+1}, v_n)$, ta nhận được đơn đồ thị gọi là *đồ thị bánh xe*, ký hiệu là W_n . Như vậy, đồ thị W_n có $n+1$ đỉnh, $2n$ cạnh, một đỉnh bậc n và n đỉnh bậc 3.



Hình 6.10. Đồ thị vòng

Ví dụ 6.12. Một số đồ thị bánh xe ban đầu



Hình 6.11. Đồ thị bánh xe

6.3.4. Đồ thị lập phương

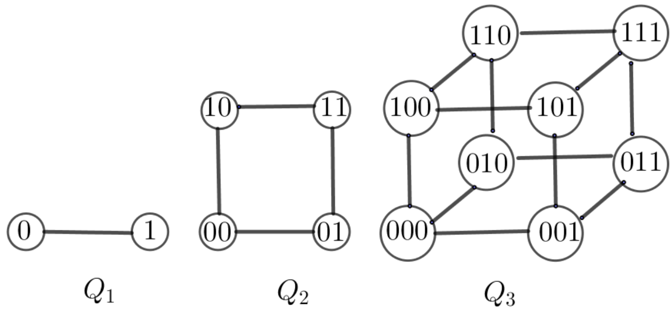
Đơn đồ thị 2^n đỉnh, tương ứng với 2^n xâu nhị phân độ dài n và hai đỉnh kề nhau khi và chỉ khi 2 xâu nhị phân tương ứng với hai đỉnh này chỉ khác nhau đúng một bit được gọi là *đồ thị lập phương*, ký hiệu là Q_n. Như vậy, mỗi đỉnh của Q_n có bậc là n và số cạnh của Q_n là $n.2^{n-1}$ (từ công thức $2|E| = \sum\limits_{v \in V} \deg(v)$).

Ví dụ 6.13. Một số đồ thị lập phương ban đầu

6.3.5. Đồ thị hai phần

Đơn đồ thị $\mathcal{G} = (V, E)$ sao cho $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, $V_1 \neq \emptyset$, $V_2 \neq \emptyset$ và mỗi cạnh của $\mathcal{G}$ được nối một đỉnh trong V_1 và một đỉnh trong V_2 được gọi là *đồ thị phân đôi*.

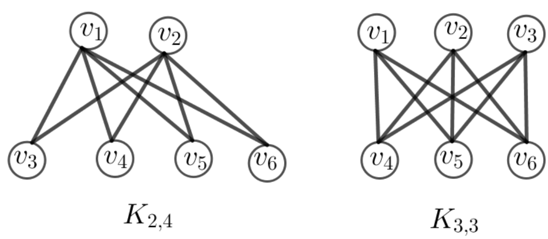
Nếu đồ thị phân đôi $\mathcal{G} = (V_1 \cup V_2, E)$ sao cho với mọi $v_1 \in V_1, v_2 \in V_2, (v_1, v_2) \in E$ thì $\mathcal{G}$ được gọi là đồ thị phân đôi đầy đủ.



Hình 6.12. Đồ thị lập phương

Nếu $|V_1| = m, |V_2| = n$ thì đồ thị phân đôi đầy đủ $\mathcal{G}$ ký hiệu là $K_{m,n}$. Như vậy $K_{m,n}$ có $m.n$ cạnh, các đỉnh của V_1 có bậc n và các đỉnh của V_2có bậc m.

Ví dụ 6.14. Một số đồ hai phần ban đầu

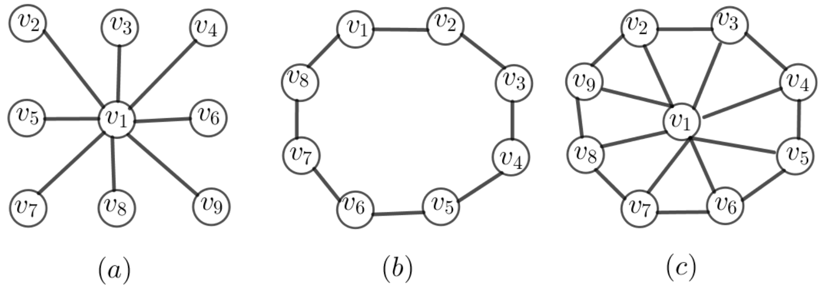


Hình 6.13. Đồ thị hai phần

6.3.6. Một vài ứng dụng của các đồ thị đặc biệt

1) Các mạng cục bộ (LAN). Một số mạng cục bộ dùng cấu trúc hình sao, trong đó tất cả các thiết bị được nối với thiết bị điều khiển trung tâm. Mạng cục bộ kiểu này có thể biểu diễn bằng một đồ thị phân đôi đầy đủ ủ $K_{1,n}$. Các thông báo gửi từ thiết bị này tới thiết bị khác đều phải qua thiết bị điều khiển trung tâm.

Mạng cục bộ cũng có thể có cấu trúc vòng tròn, trong đó mỗi thiết bị nối với đúng hai thiết bị khác. Mạng cục bộ kiểu này có thể biểu diễn bằng một đồ thị vòng C_n. Thông báo gửi từ thiết bị này tới thiết bị khác được truyền đi theo vòng tròn cho tới khi đến nơi nhận.



Hình 6.14. (a) Cấu trúc hình sao
(b) Cấu trúc vòng tròn (c) Cấu trúc hỗn hợp

Ví dụ 6.15. Một số cấu trúc mạng

Cuối cùng, một số mạng cục bộ dùng cấu trúc hỗn hợp của hai cấu trúc trên. Các thông báo được truyền vòng quanh theo vòng tròn hoặc có thể qua thiết bị trung tâm. Sự dư thừa này có thể làm cho mạng đáng tin cậy hơn. Mạng cục bộ kiểu này có thể biểu diễn bằng một đồ thị bánh xe W_n.

2) Xử lý song song. Các thuật toán để giải các bài toán được thiết kế để thực hiện một phép toán tại mỗi thời điểm là thuật toán nối tiếp. Tuy nhiên, nhiều bài toán với số lượng tính toán rất lớn như bài toán mô phỏng thời tiết, tạo hình trong y học hay phân tích mật mã không thể giải được trong một khoảng thời gian hợp lý nếu dùng thuật toán nối tiếp ngay cả khi dùng các siêu máy tính. Ngoài ra, do những giới hạn về mặt vật lý đối với tốc độ thực hiện các phép toán cơ sở, nên thường gặp các bài toán không thể giải trong khoảng thời gian hợp lý bằng các thao tác nối tiếp. Vì vậy, người ta phải nghĩ đến kiểu xử lý song song.

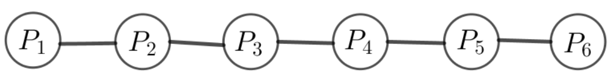
Khi xử lý song song, người ta dùng các máy tính có nhiều bộ xử lý riêng biệt, mỗi bộ xử lý có bộ nhớ riêng, nhờ đó có thể khắc phục được những hạn chế của các máy nối tiếp. Các thuật toán song song phân chia bài toán chính thành một số bài toán con sao cho có thể giải đồng thời được. Do vậy, bằng các thuật toán song song và nhờ việc sử dụng các máy tính có bộ đa xử lý, người ta hy vọng có thể giải nhanh các bài toán phức tạp. Trong thuật toán song song có một dãy các chỉ thị theo dõi việc thực hiện thuật toán, gửi các bài toán con tới các bộ xử lý khác nhau, chuyển các thông tin vào, thông tin ra tới các bộ xử lý thích hợp.

Khi dùng cách xử lý song song, mỗi bộ xử lý có thể cần các thông tin ra của các bộ xử lý khác. Do đó chúng cần phải được kết nối với nhau. Người ta có thể dùng loại đồ thị thích hợp để biểu diễn mạng kết nối các bộ xử lý trong một máy tính có nhiều bộ xử lý. Kiểu mạng kết nối dùng để thực hiện một thuật toán song song cụ thể phụ thuộc vào những yêu cầu với việc trao đổi dữ liệu giữa các bộ xử lý, phụ thuộc vào tốc độ mong muốn và tất nhiên vào phần cứng hiện có.

Mạng kết nối các bộ xử lý đơn giản nhất và cũng đắt nhất là có các liên kết hai chiều giữa mỗi cặp bộ xử lý. Các mạng này có thể mô hình bằng đồ thị đầy đủ K_n, trong đó n là số bộ xử lý. Tuy nhiên, các mạng liên kết kiểu này có số kết nối quá nhiều mà trong thực tế số kết nối cần phải có giới hạn.

Các bộ xử lý có thể kết nối đơn giản là sắp xếp chúng theo một mảng một chiều. Ưu điểm của mảng một chiều là mỗi bộ xử lý có nhiều nhất 2 đường nối trực tiếp với các bộ xử lý khác. Nhược điểm là nhiều khi cần có rất nhiều các kết nối trung gian để các bộ xử lý trao đổi thông tin với nhau.

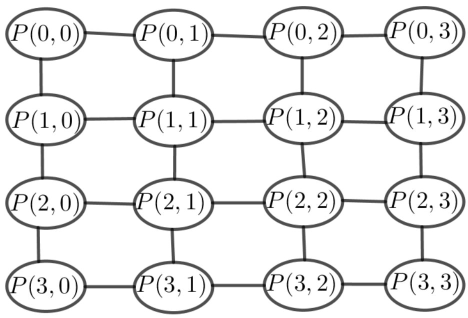
Ví dụ 6.16. Kết nối bộ vi xử lý



Hình 6.15. Mô hình kết nối mảng một chiều

Mạng kiểu lưới (hoặc mảng hai chiều) rất hay được dùng cho các mạng liên kết. Trong một mạng như thế, số các bộ xử lý là một số chính phương, $n = m^2$. Các bộ xử lý được gán nhãn $P(i,j), 0 \leq i,j \leq m-1$. Các kết nối hai chiều sẽ nối bộ xử lý $P(i,j)$ với bốn bộ xử lý bên cạnh, tức là với $P(i,j \pm 1)$ và $P(i \pm 1,j)$ chừng nào các bộ xử lý còn ở trong lưới.

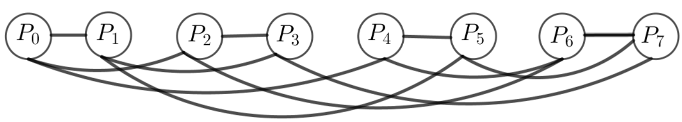
Ví dụ 6.17. Kết nối bộ vi xử lý



Hình 6.16. Mô hình kết nối mảng hai chiều

Mạng kết nối quan trọng nhất là mạng kiểu siêu khối. Với các mạng loại này số các bộ xử lý là luỹ thừa của 2, $n = 2^m$. Các bộ xử lý được gán nhãn là $P_0, P_1, ..., P_{n-1}$. Mỗi bộ xử lý có liên kết hai chiều với m bộ xử lý khác. Bộ xử lý P_i nối với bộ xử lý có chỉ số biểu diễn bằng dãy nhị phân khác với dãy nhị phân biểu diễn i tại đúng một bit. Mạng kiểu siêu khối cân bằng số các kết nối trực tiếp của mỗi bộ xử lý và số các kết nối gián tiếp sao cho các bộ xử lý có thể truyền thông được. Nhiều máy tính đã chế tạo theo mạng kiểu siêu khối và nhiều thuật toán đã được thiết kế để sử dụng mạng kiểu siêu khối. Đồ thị lập phương Q_m biểu diễn mạng kiểu siêu khối có 2^m bộ xử lý.

Ví dụ 6.18. Kết nối mạng



Hình 6.17. Mô hình kết nối mạng kiểu siêu khối

6.4 Biểu diễn đồ thị bằng ma trận và sự đẳng cấu đồ thị

Định nghĩa 6.12. Cho đồ thị $\mathcal{G} = (V, E)$ (vô hướng hoặc có hướng), với $V = \{v_1, v_2, ..., v_n\}$. *Ma trận liền kề* của $\mathcal{G}$ ứng với thứ tự các đỉnh $v_1, v_2, ..., v_n$ là ma trận

$$A = (a_{ij})_{n\times n}$$

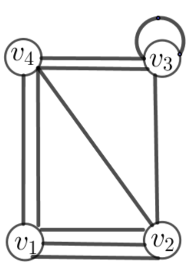
Trong đó a_{ij} là

$$a_{ij} = \begin{cases} \text{Số các cạnh (cung) nối } v_i \text{ nối với } v_j \text{ nếu } v_i \neq v_j \\ \text{Số cạnh (cung) vòng tại } v_j \text{ nếu } v_i = v_j \end{cases}$$

Như vậy, ma trận liền kề của một đồ thị vô hướng là ma trận đối xứng, nghĩa là $a_{ij} = a_{ji}$, trong khi ma trận liền kề của một đồ thị có hướng không có tính đối xứng.

Ví dụ 6.19. Ma trận liền kề với thứ tự các đỉnh v_1, v_2, v_3, v_4 là

$$A = \begin{array}{c c} & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_3 \end{array} & \begin{pmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{pmatrix} \end{array}$$



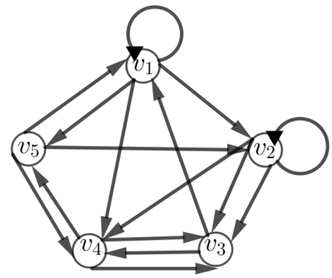
Hình 6.18. Ma trận liền kề đồ thị vô hướng

Ví dụ 6.20. Ma trận liền kề với thứ tự các đỉnh v_1, v_2, v_3, v_4, v_5 là

Định nghĩa 6.13. Cho đồ thị vô hướng $\mathcal{G} = (V, E), v_1, v_2, ..., v_n$ là các đỉnh và $e_1, e_2, ..., e_m$ là các cạnh của $\mathcal{G}$. *Ma trận liên thuộc của $\mathcal{G}$* theo thứ tự trên của V và E là ma trận

$$A = (a_{ij})_{n\times m}$$

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$



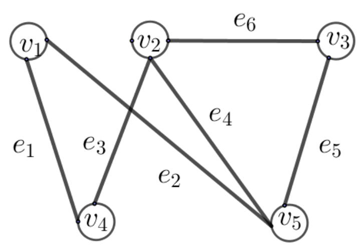
Hình 6.19. Ma trận liền kề đồ thị có hướng

trong đó

$$a_{ij} = \begin{cases} 1 & \text{nếu cạnh } e_j \text{ nối với đỉnh } v_i \\ 0 & \text{nếu cạnh } e_j \text{ không nối với đỉnh } v_i. \end{cases}$$

Ví dụ 6.21. Ma trận liên thuộc theo thứ tự các đỉnh $v_1, v_2, v_3, v_4, v - 5$ và các cạnh $e_1, e_2, e_3, e_4, e_5, e_6$ là

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 9 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

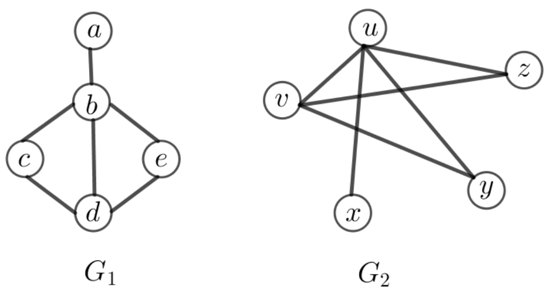


Hình 6.20. Ma trận liên thuộc đồ thị không hướng

Định nghĩa 6.14. Các đơn đồ thị $\mathcal{G}_1 = (V_1, E_1)$ và $\mathcal{G}_2 = (V_2, E_2)$ được gọi là *đẳng cấu* nếu tồn tại một song ánh f từ V_1 lên V_2 sao cho các đỉnh u và v là liền kề trong $\mathcal{G}_1$ khi và chỉ khi $f(u)$ và $f(v)$ là liền kề trong $\mathcal{G}_2$ với mọi u và v trong V_1. Ánh xạ f như thế gọi là một *phép đẳng cấu*.

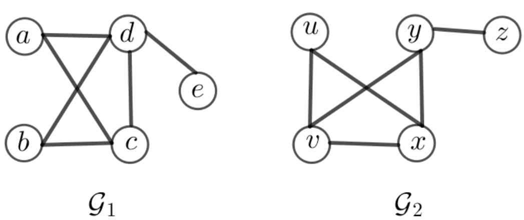
Thông thường, để chứng tỏ hai đơn đồ thị là không đẳng cấu, người ta chỉ ra chúng không có chung một tính chất mà các đơn đồ thị đẳng cấu cần phải có. Tính chất như thế gọi là một bất biến đối với phép đẳng cấu của các đơn đồ thị.

Ví dụ 6.22. Hai đơn đồ thị $\mathcal{G}_1$ và $\mathcal{G}_2$ sau là đẳng cấu qua phép đẳng cấu $f : a \to x, b \to u, c \to z, d \to v, e \to y$, (Hình 6.21).



Hình 6.21. Hai đồ thị đẳng cấu

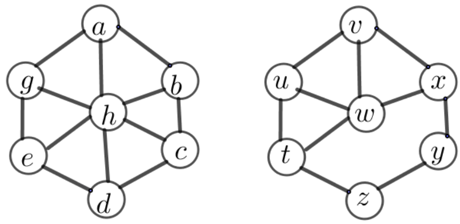
Ví dụ 6.23. Hai đồ thị $\mathcal{G}_1$ và $\mathcal{G}_2$ (Hình 6.22) đều có 5 đỉnh và 6 cạnh nhưng không đẳng cấu vì trong $\mathcal{G}_1$ có một đỉnh bậc 4 mà trong $\mathcal{G}_2$ không có đỉnh bậc 4 nào.



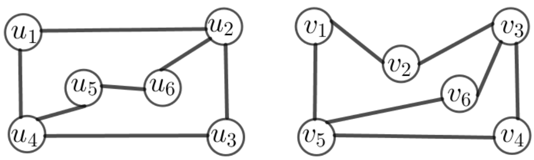
Hình 6.22. Hai đồ thị không đẳng cấu do có bậc đỉnh khác nhau

Ví dụ 6.24. Hai đồ thị $\mathcal{G}_1$ và $\mathcal{G}_2$ (Hình 6.23) đều có 7 đỉnh, 10 cạnh, cùng có một đỉnh bậc 4, bốn đỉnh bậc 3 và hai đỉnh bậc 2. Tuy nhiên $\mathcal{G}_1$ và $\mathcal{G}_2$ là không đẳng cấu vì hai đỉnh bậc 2 của $\mathcal{G}_1$ (a và d) là không kề nhau, trong khi hai đỉnh bậc 2 của $\mathcal{G}_2$ (y và z) là kề nhau.

Ví dụ 6.25. Hai đồ thị $\mathcal{G}_1$ và $\mathcal{G}_2$ là đẳng cấu vì hai ma trận liền kề của $\mathcal{G}_1$ theo thứ tự các đỉnh $u_1, u_2, u_3, u_4, u_5, u_6$ và của $\mathcal{G}_2$ theo thứ tự các đỉnh $v_6, v_3, v_4, v_5, v_1, v_2$ là như nhau và bằng



Hình 6.23. Hai đồ thị không đẳng cấu do có hai đỉnh không kề



Hình 6.24. Hai đồ thị đẳng cấu nhờ ma trận kề

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

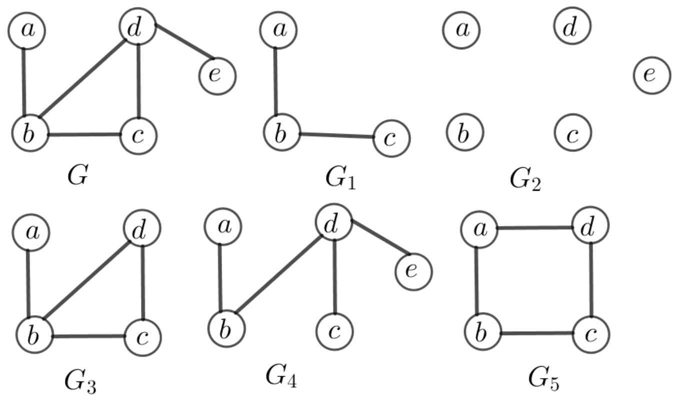
6.5 Đồ thị con và đồ thị bao trùm

Định nghĩa 6.15. Cho hai đồ thị $\mathcal{G}_1 = (V_1, E_1)$ và $\mathcal{G}_2 = (V_2, E_2)$. Ta nói $\mathcal{G}_2$ là *đồ thị con* của $\mathcal{G}_1$ nếu $V_2 \subset V_1$ và $E_2 \subset E_1$. Trong trường hợp $V_1 = V_2$ thì $\mathcal{G}_2$ gọi là con *bao trùm* của $\mathcal{G}_1$

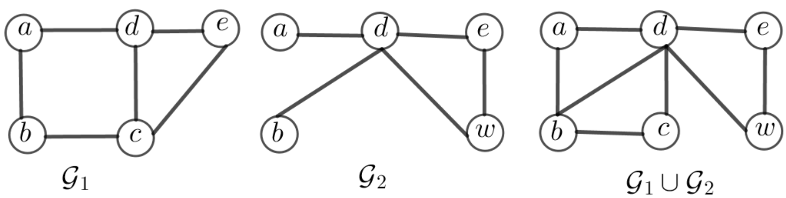
Ví dụ 6.26. Hình 6.25, $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ và $\mathcal{G}_4$ là các đồ thị con của $\mathcal{G}$, trong đó $\mathcal{G}_2$ và $\mathcal{G}_4$ là đồ thị con bao trùm của $\mathcal{G}$, còn $\mathcal{G}_5$ không phải là đồ thị con của $\mathcal{G}$.

Định nghĩa 6.16. *Hợp của hai đơn đồ thị* $\mathcal{G}_1 = (V_1, E_1)$ và $\mathcal{G}_2 = (V_2, E_2)$ là một đơn đồ thị có tập các đỉnh là $V_1 \cup V2$ và tập các cạnh là $E_1 \cup E_2$, ký hiệu là $\mathcal{G}_1 \cup \mathcal{G}_2$

Ví dụ 6.27. Hình 6.26 Hợp hai đồ thị



Hình 6.25. Đồ thị con

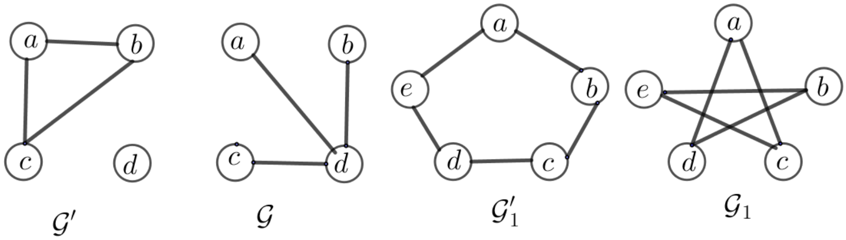


Hình 6.26. Đồ thị con

Định nghĩa 6.17. Đơn đồ thị $\mathcal{G}' = (V, E')$ được gọi là *đồ thị bù* của đơn đồ thị $\mathcal{G} = (V, E)$ nếu $\mathcal{G}$ và $\mathcal{G}'$ không có cạnh chung nào $(E \cap E' = \emptyset)$ và $\mathcal{G} \cup \mathcal{G}'$ là đồ thị đầy đủ.

Dễ thấy rằng nếu $\mathcal{G}'$ là bù của $\mathcal{G}$ thì $\mathcal{G}$ cũng là bù của $\mathcal{G}'$. Khi đó ta nói hai đồ thị là bù nhau.

Ví dụ 6.28. Hình 6.27 hai đồ thị $\mathcal{G}'$ và $\mathcal{G}$ là bù nhau và hai đồ thị $\mathcal{G}_1$ và $\mathcal{G}_1'$ là bù nhau.



Hình 6.27. Đồ thị bù

6.6 Tính liên thông

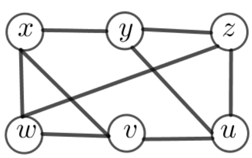
Định nghĩa 6.18. *Đường đi* độ dài n từ đỉnh u đến đỉnh v, với n là một số nguyên dương, trong đồ thị (giả đồ thị vô hướng hoặc đa đồ thị có hướng) $\mathcal{G} = (V, E)$ là một dãy các cạnh (hoặc cung) $e_1, e_2, ..., e_n$ của đồ thị sao cho $e_1 = (x_0, x_1), e_2 = (x_1, x_2), ..., e_n = (x_{n-1}, x_n)$, với $x_0 = u$ và $x_n = v$. Khi đồ thị không có cạnh (hoặc cung) bội, ta ký hiệu đường đi này bằng dãy các đỉnh $x_0, x_1, ..., x_n$.

Đường đi được gọi là *chu trình* nếu nó bắt đầu và kết thúc tại cùng một đỉnh.

Đường đi hoặc chu trình gọi là *đơn* nếu nó không chứa cùng một cạnh (hoặc cung) quá một lần.

Một đường đi hoặc 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* hoặc *chu trình sơ cấp*. Rõ ràng rằng một đường đi (t.ư. chu trình) sơ cấp là đường đi (t.ư. chu trình) đơn.

Ví dụ 6.29. Hình 6.28 trong đơn đồ thị x, y, z, w, v, y là đường đi đơn (không sơ cấp) độ dài 5; x, w, v, z, y không là đường đi vì (v, z) không là cạnh; y, z, w, x, v, u, y là chu trình sơ cấp độ dài 6.

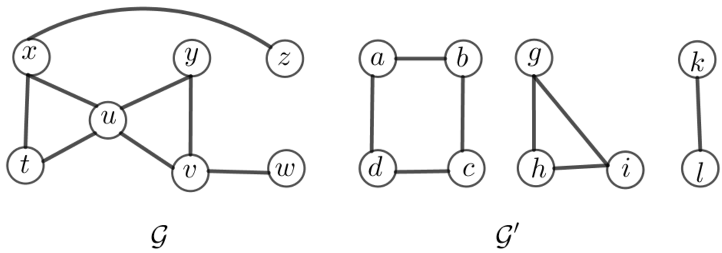


Hình 6.28. Đường đi và chu trình

Định nghĩa 6.19. Một đồ thị (vô hướng) được gọi là *liên thông* nếu có đường đi giữa mọi cặp đỉnh phân biệt của đồ thị.

Một đồ thị không liên thông là hợp của hai hay nhiều đồ thị con liên thông, mỗi cặp các đồ thị con này không có đỉnh chung. Các đồ thị con liên thông rời nhau như vậy được gọi là các thành phần liên thông của đồ thị đang xét. Như vậy, một đồ thị là liên thông khi và chỉ khi nó chỉ có một thành phần liên thông.

Ví dụ 6.30. Hình 6.29 đồ thị $\mathcal{G}$ là liên thông, nhưng đồ thị $\mathcal{G}'$ không liên thông và có 3 thành phần liên thông.

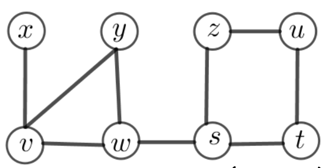


Hình 6.29. Đồ thị liên thông

Định nghĩa 6.20. Một đỉnh trong đồ thị $\mathcal{G}$ mà khi xoá đi nó và tất cả các cạnh liên thuộc với nó ta nhận được đồ thị con mới có nhiều thành phần liên thông hơn đồ thị $\mathcal{G}$ được gọi là *đỉnh cắt* hay *điểm khớp*. Việc xoá đỉnh cắt khỏi một đồ thị liên thông sẽ tạo ra một đồ thị con không liên thông.

Hoàn toàn tương tự, một cạnh mà khi ta bỏ nó đi sẽ tạo ra một đồ thị có nhiều thành phần liên thông hơn so với đồ thị xuất phát được gọi là *cạnh cắt* hay là *cầu*.

Ví dụ 6.31. Hình 6.30 đồ thị có các đỉnh cắt là v, w, s và các cầu là $(x, v), (w, s)$.



Hình 6.30. Đỉnh cắt và cầu

Mệnh đề 6.4. *Giữa mọi cặp đỉnh phân biệt của một đồ thị liên thông luôn có đường đi sơ cấp.*

Chứng minh. Giả sử u và v là hai đỉnh phân biệt của một đồ thị liên thông $\mathcal{G}$. Vì $\mathcal{G}$ liên thông nên có ít nhất một đường đi giữa u và v. Gọi $x_0, x_1, ..., x_n$, với $x_0 = u$ và $x_n = v$, là dãy các đỉnh của đường đi có độ dài ngắn nhất. Đây chính là đường đi sơ cấp cần tìm. Thật vậy, giả sử nó không là đường đi đơn, khi đó $x_i = x_j$ với $0 \leq i < j$. Điều này có nghĩa là giữa các đỉnh u và v có đường đi ngắn hơn qua các đỉnh $x_0, x_1, ..., x_{i-1}, x_j, ..., x_n$ nhận được bằng cách xoá đi các cạnh tương ứng với dãy các đỉnh $x_i, ..., x_{j-1}$.

Mệnh đề 6.5. *Mọi đơn đồ thị n đỉnh ($n \geq 2$) có tổng bậc của hai đỉnh tuỳ ý không nhỏ hơn n đều là đồ thị liên thông.*

Chứng minh. Cho đơn đồ thị $\mathcal{G} = (V, E)$ có n đỉnh ($n \geq 2$) và thoả mãn yêu cầu của bài toán. Giả sử $\mathcal{G}$ không liên thông, tức là tồn tại hai đỉnh u và v sao cho không có đường đi nào nối u và v. Khi đó trong đồ thị $\mathcal{G}$ tồn tại hai thành phần liên thông là $\mathcal{G}_1$ có n_1 đỉnh và chứa u, $\mathcal{G}_2$ chứa đỉnh v và có n_2 đỉnh. Vì $\mathcal{G}_1, \mathcal{G}_2$ là hai trong số các thành phần liên thông của $\mathcal{G}$ nên $n_1 + n_2 \leq n$. ta có:

$$\deg(u) + \deg(v) \leq (n_1 - 1) + (n_2 - 1) = n_1 + n_2 - 2 \leq n - 2 < n.$$

Điều mâu thuẫn ở trên dẫn đến kết luận là đồ thị $\mathcal{G}$ phải liên thông.

Hệ quả 6.2. *Đơn đồ thị mà bậc của mỗi đỉnh của nó không nhỏ hơn một nửa số đỉnh là đồ thị liên thông.*

Mệnh đề 6.6. *Nếu một đồ thị có đúng hai đỉnh bậc lẻ thì hai đỉnh này phải liên thông, tức là có một đường đi nối chúng.*

Chứng minh. Cho $\mathcal{G} = (V, E)$ là đồ thị thị có đúng hai đỉnh bậc lẻ là u và v. Giả sử u và v không liên thông với nhau. Khi đó chúng phải thuộc hai thành phần liên thông nào đó của đồ thị $\mathcal{G}$, $\mathcal{G}_1$ chứa u và $\mathcal{G}_2$ chứa v.

Bậc của đỉnh u trong $\mathcal{G}_1$ cũng chính là bậc của u trong $\mathcal{G}$, nên trong $\mathcal{G}_1$ đỉnh u vẫn có bậc lẻ và $\mathcal{G}_1$ có duy nhất một đỉnh bậc lẻ. Điều này mâu thuẫn. Vậy hai đỉnh u và v phải liên thông.

Mệnh đề 6.7. *Cho $\mathcal{G} = (V, E)$ là một đồ thị liên thông. Khi đó một đỉnh của $\mathcal{G}$ là điểm khớp khi và chỉ khi trong $\mathcal{G}$ tồn tại hai đỉnh u và v sao cho mỗi đường đi nối u và v đều phải đi qua đỉnh này.*

Chứng minh. *Điều kiện cần:* Giả sử đỉnh x là điểm khớp trong đồ thị $\mathcal{G}$. Khi đó đồ thị con $\mathcal{G}_1$ của $\mathcal{G}$ nhận được bằng cách xoá x và các cạnh liên thuộc với nó là không liên thông. Giả sử $\mathcal{G}_2, \mathcal{G}_3$ là hai trong các thành phần liên thông của $\mathcal{G}_1$. Lấy u là đỉnh trong $\mathcal{G}_2$ và v là đỉnh trong $\mathcal{G}_3$. Do u, v thuộc hai thành phần liên thông khác nhau, nên trong $\mathcal{G}_1$ các đỉnh u, v không liên thông. Nhưng trong $\mathcal{G}$ các đỉnh u, v lại liên thông, nên mọi đường đi nối u, v đều phải đi qua đỉnh x.

Điều kiện đủ: Giả sử mọi đường đi nối u, v đều đi qua đỉnh x, nên nếu bỏ đỉnh x và các cạnh liên thuộc với x thì đồ thị con $\mathcal{G}_1$ nhận được từ $\mathcal{G}$ chứa hai đỉnh u, v không liên thông. Do đó $\mathcal{G}_1$ là đồ thị không liên thông hay đỉnh x là điểm khớp của $\mathcal{G}$.

Định lý 6.1. *Cho $\mathcal{G}$ là một đơn đồ thị có n đỉnh, m cạnh và k thành phần liên thông. Khi đó*

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

Chứng minh. Bất đẳng thức $n - k \leq m$ được chứng minh bằng quy nạp theo m. Nếu $m = 0$ thì $k = n$ nên bất đẳng thức đúng. Giả sử bất đẳng thức đúng đến $m - 1$, với $m \geq 1$. Gọi $\mathcal{G}'$ là đồ thị con bao trùm của $\mathcal{G}$ có số cạnh m_0 là nhỏ nhất sao cho nó có k thành phần liên thông. Do đó việc loại bỏ bất cứ cạnh nào trong $\mathcal{G}'$ cũng tăng số thành phần liên thông lên 1 và khi đó đồ thị thu được sẽ có n đỉnh, $k+1$ thành phần liên thông và $m_0 - 1$ cạnh. Theo giả thiết quy nạp, ta có $m_0 - 1 \geq n - (k+1)$ hay $m_0 \geq n - k$. Vậy $m \geq n - k$.

Bổ sung cạnh vào $\mathcal{G}$ để nhận được đồ thị $\mathcal{G}''$ có m_1 cạnh sao cho k thành phần liên thông là những đồ thị đầy đủ. Ta có $m \leq m_1$ nên chỉ cần chứng minh

$$m_1 \leq \frac{(n-k)(n-k+1)}{2}.$$

Giả sử $\mathcal{G}_i$ và $\mathcal{G}_j$ là hai thành phần liên thông của $\mathcal{G}''$ với n_i và n_j đỉnh và $n_i \geq n_j > 1$(*). Nếu ta thay $\mathcal{G}_i$ và $\mathcal{G}_j$ bằng đồ thị đầy đủ với $n_i + 1$ và $n_j - 1$ đỉnh thì tổng số đỉnh không thay đổi nhưng số cạnh tăng thêm một lượng là

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

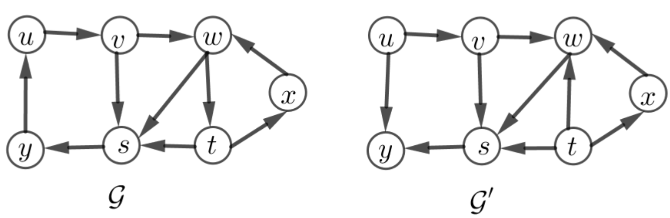
Thủ tục này được lặp lại khi hai thành phần nào đó có số đỉnh thoả (*). Vì vậy m_1 là lớn nhất (n, k là cố định) khi đồ thị gồm $k-1$ đỉnh cô lập và một đồ thị đầy đủ với $n-k+1$ đỉnh. Từ đó suy ra bất đẳng thức cần tìm.

Định nghĩa 6.21. Đồ thị có hướng $\mathcal{G}$ được gọi là *liên thông mạnh* nếu với hai đỉnh phân biệt bất kỳ u và v của $\mathcal{G}$ đều có đường đi từ u tới v và đường đi từ v tới u.

Đồ thị có hướng $\mathcal{G}$ được gọi là *liên thông yếu* nếu đồ thị vô hướng nền của nó là liên thông.

Đồ thị có hướng $\mathcal{G}$ được gọi là *liên thông một chiều* nếu với hai đỉnh phân biệt bất kỳ u và v của $\mathcal{G}$ đều có đường đi từ u tới v hoặc đường đi từ v tới u.

Ví dụ 6.32. Hình 6.31 đồ thị $\mathcal{G}$ là liên thông mạnh nhưng đồ thị $\mathcal{G}'$ là liên thông yếu (không có đường đi từ u tới x cũng như từ x tới u).



Hình 6.31. Đồ thị có hướng liên thông

Mệnh đề 6.8. *Cho $\mathcal{G}$ là một đồ thị (vô hướng hoặc có hướng) với ma trận liền kề A theo thứ tự các đỉnh $v_1, v_2, ..., v_n$. Khi đó số các đường đi khác nhau độ dài r từ v_i tới v_j trong đó r là một số nguyên dương, bằng giá trị của phần tử dòng i cột j của ma trận A^r.*

Chứng minh. Ta chứng minh mệnh đề bằng quy nạp theo r. Số các đường đi khác nhau độ dài 1 từ v_i tới v_j là số các cạnh (hoặc cung) từ v_i tới v_j, đó chính là phần tử dòng i cột j của ma trận A; nghĩa là, mệnh đề đúng khi $r = 1$.

Giả sử mệnh đề đúng đến r; nghĩa là, phần tử dòng i cột j của A^r là số các đường đi khác nhau độ dài r từ v_i tới v_j. Vì $A^{r+1} = A^r.A$ nên phần tử dòng i cột j của A^{r+1} bằng

$$b_{i1}a_{1j} + b_{i2}a_{2j} + ... + b_{in}a_{nj},$$

trong đó b_{ik} là phần tử dòng i cột k của A^r. Theo giả thiết quy nạp b_{ik} là số đường đi khác nhau độ dài r từ v_i tới v_k.

Đường đi độ dài $r+1$ từ v_i tới v_j sẽ được tạo nên từ đường đi độ dài r từ v_i tới đỉnh trung gian v_k nào đó và một cạnh (hoặc cung) từ v_k tới v_j. Theo quy tắc nhân số các đường đi như thế là tích của số đường đi độ dài r từ v_i tới v_k, tức là b_{ik}, và số các cạnh (hoặc cung) từ v_k tới v_j, tức là a_{kj}. Cộng các tích này lại theo tất cả các đỉnh trung gian v_k ta có mệnh đề đúng đến $r+1$.

6.7 Bài tập

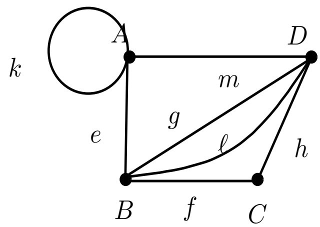
▷ **6.1.** Cho $\mathcal{G}$ là đồ thị có v đỉnh và e cạnh, còn M, m tương ứng là bậc lớn nhất và nhỏ nhất của các đỉnh của $\mathcal{G}$. Chứng tỏ rằng

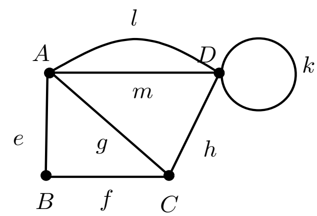
$$m \leq \frac{2e}{v} \leq M.$$

▷ **6.2.** Chứng minh rằng nếu $\mathcal{G}$ là đơn đồ thị phân đôi có v đỉnh và e cạnh, khi đó $e \leq \frac{v^2}{4}$.

▷ **6.3.** Trongmột phương án mạng kiểu lưới kết nối $n = m^2$ bộ xử lý song song, bộ xử lý $P(i, j)$ được kết nối với 4 bộ xử lý $(P(i \pm 1) \mod m$, j), $P(i, (j \pm 1) \mod m)$, sao cho các kết nối bao xung quanh các cạnh của lưới. Hãy vẽ mạng kiểu lưới có 16 bộ xử lý theo phương án này.

▷ **6.4.** Lập ma trận liên thuộc và lập ma trận cạnh kề đồ thị sau đây

a) 

b) 

▷ **6.5.** Hãy vẽ các đồ thị vô hướng được biểu diễn bởi ma trận liền kề sau: a) $\begin{pmatrix}1&2&3\\2&0&4\\3&4&0\end{pmatrix}$, b) $\begin{pmatrix}1&2&0&1\\2&0&3&0\\0&3&1&1\\1&0&1&0\end{pmatrix}$, c) $\begin{pmatrix}0&1&3&0&4\\1&2&1&3&0\\3&1&1&0&1\\0&3&0&0&2\\4&0&1&2&3\end{pmatrix}$.

▷ **6.6.** Nêu ý nghĩa của tổng các phần tử trên một hàng (t.ư. cột) của một ma trận liền kề đối với một đồ thị vô hướng ? Đối với đồ thị có hướng ?

▷ **6.7.** Tìm ma trận liền kề cho các đồ thị sau
a) K_n, b) C_n, c) W_n, d) $K_{m,n}$, e) Q_n.

▷ **6.8.** Có bao nhiêu đơn đồ thị không đẳng cấu với n đỉnh khi
a) $n = 2$, b) $n = 3$, c) $n = 4$.

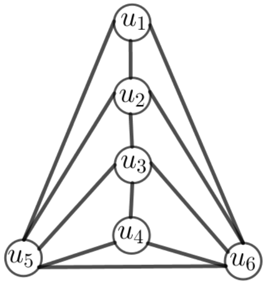
▷ **6.9.** Hai đơn đồ thị với ma trận liền kề sau đây có là đẳng cấu không?

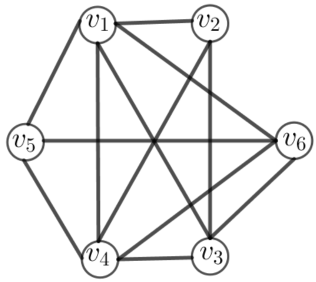
$$\begin{pmatrix}0&1&0&1\\1&0&0&1\\0&0&0&1\\1&1&1&0\end{pmatrix}, \begin{pmatrix}0&1&1&1\\1&0&0&1\\1&0&0&1\\1&1&1&0\end{pmatrix}$$

▷ **6.10.** Hai đơn đồ thị với ma trận liền kề sau đây có là đẳng cấu không?

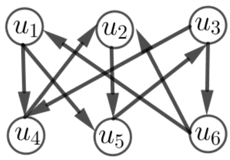
$$\begin{pmatrix}1&1&0&0&0\\1&0&1&0&1\\0&0&0&1&1\\0&1&1&1&0\end{pmatrix}, \qquad \begin{pmatrix}0&1&0&0&1\\0&1&1&1&0\\1&0&0&1&0\\1&0&1&0&1\end{pmatrix}$$

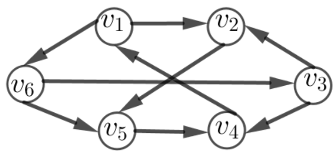
▷ **6.11.** Các đồ thị $\mathcal{G}$ và $\mathcal{G}'$ sau có đẳng cấu với nhau không?





a)





b)

▷ **6.12.** Cho $V = \{2, 3, 4, 5, 6, 7, 8\}$ và E là tập hợp các cặp phần tử (u, v) của V sao cho $u < v$ và u, v nguyên tố cùng nhau. Hãy vẽ đồ thị có hướng $\mathcal{G} = (V, E)$. Tìm số các đường đi phân biệt độ dài 3 từ đỉnh 2 tới đỉnh 8.

▷ **6.13.** Hãy tìm số đường đi độ dài n giữa hai đỉnh liền kề (t.ư. không liền kề) tùy ý trong $K_{3,3}$ với mỗi giá trị của n sau
a) $n = 2$, b) $n = 3$, c) $n = 4$, d) $n = 5$.

▷ **6.14.** Một cuộc họp có ít nhất ba đại biểu đến dự. Mỗi người quen ít nhất hai đại biểu khác. Chứng minh rằng có thể xếp được một số đại biểu ngồi xung quanh một bàn tròn, để mỗi người ngồi giữa hai người mà đại biểu đó quen.

▷ **6.15.** Một lớp học có ít nhất 4 sinh viên. Mỗi sinh viên thân với ít nhất 3 sinh viên khác. Chứng minh rằng có thể xếp một số chẵn sinh viên ngồi quanh một cái bàn tròn để mỗi sinh viên ngồi giữa hai sinh viên mà họ thân.

▷ **6.16.** Trong một cuộc họp có đúng hai đại biểu không quen nhau và mỗi đại biểu này có một số lẻ người quen đến dự. Chứng minh rằng luôn luôn có thể xếp một số đại biểu ngồi chen giữa hai đại biểu nói trên, để mỗi người ngồi giữa hai người mà anh ta quen.

▷ **6.17.** Một thành phố có n ($n \geq 2$) nút giao thông và hai nút giao thông bất kỳ đều có số đầu mối đường ngầm tới một trong các nút giao thông này đều không nhỏ hơn n. Chứng minh rằng từ một nút giao thông tuỳ ý ta có thể đi đến một nút giao thông bất kỳ khác bằng đường ngầm.

Chương

7 ĐƯỜNG ĐI VÀ CHU TRÌNH

7.1 Các thuật toán duyệt đồ thị

Phép duyệt đồ thị là một cách liệt kê tất cả các đỉnh của đồ thị này thành một danh sách tuyến tính. Hay nói một cách khác, phép duyệt đồ thị cho ta một cách "đi qua" tất cả các đỉnh của đồ thị để truy nhập, thêm bớt thông tin ở các đỉnh của đồ thị đó.

Phép duyệt đồ thị không phụ thuộc vào hướng của các cạnh. Do vậy, với đồ thị có hướng thì ta vô hướng hoá trước khi duyệt.

7.1.1. Mô hình thuật toán duyệt đồ thị

Giả sử $\mathcal{G} = (V, E)$ là đồ thị đã cho và v_0 là một đỉnh nào đó của $\mathcal{G}$. Ký hiệu DS là một cấu trúc dữ liệu kiểu danh sách dùng để chứa các đỉnh.

1) Khởi đầu: $DS \leftarrow \{v_0\}$
2) Lấy đỉnh v ra khỏi đầu DS
3) Duyệt đỉnh v
4) Nạp các đỉnh của liền kề vào DS
5) Nếu $DS \neq \emptyset$ thì quay lên bước 2)
6) Dừng.

7.1.2. Duyệt theo chiều sâu (Depth-First Search)

Nếu trong thuật toán trên, danh sách DS được tổ chức theo kiểu stack (danh sách vào sau - ra trước – LIFO) thì ta có phương pháp duyệt theo chiều sâu.

Trong phương pháp này mỗi lần duyệt một đỉnh ta duyệt đến tận cùng mỗi nhánh rồi mới chuyển sang duyệt nhánh khác.

Giả sử $\mathcal{G} = (V, E)$ là một đồ thị vô hướng.

Ta bắt đầu duyệt từ một đỉnh v_0 nào đó của đồ thị. Sau đó chọn v là đỉnh kề nào đó của v_0 và lặp lại quá trình duyệt đối với đỉnh v.

Giả sử ta đang xét đỉnh v. Nếu trong số các đỉnh kề với v, ta tìm được đỉnh w chưa được duyệt thì ta sẽ xét đỉnh này và bắt đầu từ đó ta tiếp tục quá trình duyệt.

Ngược lại, nếu không còn đỉnh nào kề với v chưa được duyệt thì ta nói rằng đỉnh v đã duyệt xong và quay trở lại tiếp tục duyệt từ đỉnh mà từ đó ta đến được đỉnh v.

Nếu quay trở lại đúng đỉnh v_0 thì phép duyệt kết thúc.

Thuật toán 7.1. Duyệt đồ thị theo chiều sâu

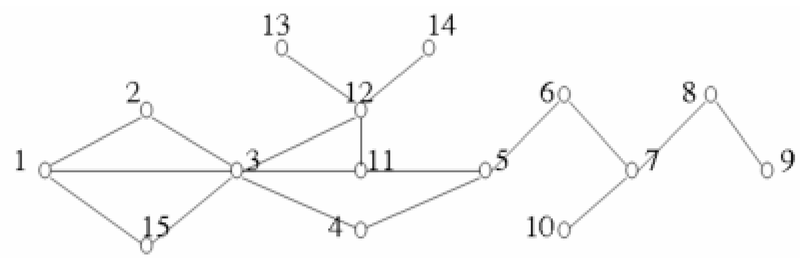
Dữ liệu: Biểu diễn mảng DK các danh sách kề của đồ thị vô hướng $\mathcal{G}$.

Kết quả: Danh sách các đỉnh của đồ thị $\mathcal{G}$.

procedure $DFSearch(v)$;
 begin
   Tham_dinh (v);
   $Duyet[v] := true$;
   for $u \in DK[v]$ **do**
      if $!Duyet[u]$ **then** $DFSearch(u)$;
 end ;
 begin // *Chương trình chính*
    for $v \in V$ **do** $Duyet[v] := false$;
    for $v \in V$ **do**
    if $!Duyet[v]$ **then** $DFSearch(v)$;
end .

Độ phức tạp của thật toán là: $O(n+m)$

Ví dụ 7.1. Đồ thị được duyệt theo chiều sâu.



Hình 7.1. Thứ tự của các đỉnh được duyệt theo chiều sâu

Trong thuật toán duyệt theo chiều sâu, đỉnh được thăm càng muộn càng sớm trở thành duyệt xong. Do vậy việc dùng một ngăn xếp (stack) để lưu trữ các đỉnh đang duyệt là rất thích hợp. Ta có thủ tục cải tiến sau đây:

Thuật toán 7.2. Duyệt đồ thị theo chiều sâu dùng ngăn xếp

Dữ liệu: Biểu diễn mảng DK các danh sách kề của đồ thị vô hướng $\mathcal{G}$.

Kết quả: Danh sách các đỉnh của đồ thị $\mathcal{G}$.

procedure $DFSearch(v)$;

```
begin
   S := ∅
   Tham_dinh (v);
   Duyet[v] := true;
    push v onto S;// Nạp v lên đỉnh của S
    while S ≠ ∅ do
       begin
           while ∃u ∈ DK[top(S)] do
              if !Duyet[u] then
               begin
                  Tham_dinh (u);
                  Duyet[u] := true;
                   push u onto S; //Loại bỏ phần tử ở đỉnh
của S
               end;
             pop(S);
       end;
 end;
 begin  // Chương trình chính
    for v ∈ V do Duyet[v] := false;
    for v ∈ V do
    if !Duyet[v] then DFSearch(v);
end.
```

7.1.3. Duyệt theo chiều rộng (Breadth-First Search)

Nếu trong thuật toán duyệt đồ thị, cấu trúc danh sách *DS* được tổ chức theo kiểu hàng đợi (danh sách vào trước - ra trước – FIFO) thì ta có phương pháp duyệt theo chiều rộng. Trong phương pháp này việc duyệt có tính chất "lan rộng".

Một đỉnh được duyệt xong ngay sau khi ta đã xét hết tất cả các đỉnh kề với nó. Đỉnh được xét càng sớm thì sớm trở thành duyệt xong.

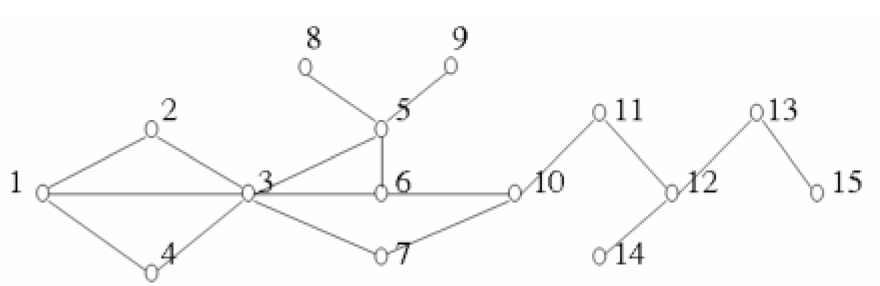
Thuật toán 7.3. Duyệt đồ thị theo chiều rộng

Đầu vào: Biểu diễn mảng *DK* các danh sách kề của đồ thị vô hướng $\mathcal{G}$.
Đầu ra: Danh sách các đỉnh của đồ thị $\mathcal{G}$.

```
procedure BFSearch(v);
begin
  Q := ∅;
  enqueue v into Q; // Nạp v vào cuối hàng đợi Q
  Duyet[v] := true;
  while Q ≠ ∅ do
  begin
   dequeue z from Q; //Loại z ra khỏi đầu hàng đợi Q
   Tham_dinh (z);
   for u ∈ DK[z] do
    if !Duyet[u] then
     begin
      enqueue u into Q;
      Duyet[u] := true;
     end;
    end;
end;
 begin // Chương trình chính
     for v ∈ V do Duyet[v] := false;
     for v ∈ V do
     if !Duyet[v] then BFSearch(v);
end.
```

Thuật toán này cũng có độ phức tạp là $O(n+m)$.

Ví dụ 7.2. Đồ thị trong Ví dụ 7.1 được duyệt theo chiều rộng.



Hình 7.2. Thứ tự của các đỉnh được duyệt theo chiều rộng

7.2 Một số ứng dụng duyệt đồ thị

7.2.1. Bài toán tìm đường đi

Giả sử a và b là hai đỉnh nào đó của đồ thị vô hướng $\mathcal{G}$. Hãy tìm đường đi (nếu có) từ đỉnh a đến đỉnh b.

Theo các phương pháp duyệt đồ thị, các lời gọi thủ tục $DFSearch(a)$ hoặc $BFSearch(a)$ cho phép thăm tất cả các đỉnh thuộc cùng thành phần liên thông với đỉnh a. Vì vậy, sau khi thực hiện xong thủ tục nếu $Duyet[b] = false$ thì không có đường đi từ đỉnh a đến đỉnh b ngược lại thì đỉnh b thuộc cùng mảng liên thông với đỉnh a, hay nói một cách khác, có đường đi từ a đến b.

Để khôi phục đường đi ta dùng thêm một biến mảng $Truoc$. Thành phần $Truoc[u]$ ghi lại đỉnh đến trước đỉnh u trên đường duyệt từ a tới u. Khi đó, trong thủ tục $DFSearch(v)$ cần sửa đổi câu lệnh **if** ở dòng lệnh 6 như sau:

if $!Duyet[u]$ **then begin** $Truoc[u] := v; DFSearch(u)$ **end**;

còn trong thủ tục $BFSearch(v)$ thì cần sửa câu lệnh **if** ở các dòng lệnh

Sửa đoạn mã trong duyệt theo chiều rộng *BFSearch*

```
if !Duyet[u] then
  begin
   enqueue u into Q;
   Duyet[u] := true; Truoc[u] := z;
  end;
```

Đường đi cần tìm (nếu có) sẽ được khôi phục như sau

$$b \leftarrow a_1 = Truoc[b] \leftarrow a_2 = Truoc[a1] \leftarrow .... \leftarrow a$$

Chú ý: Đường đi tìm được theo thuật toán duyệt theo chiều rộng là đường đi ngắn nhất từ đỉnh a đến đỉnh b.

7.2.2. Bài toán tìm các mảng liên thông

Cho đồ thị $\mathcal{G}$, hãy tìm số mảng liên thông p của đồ thị này và xác định xem mỗi mảng liên thông bao gồm những đỉnh nào.

Do các thủ tục $DFSearch(v)$ hoặc $BFSearch(v)$ cho phép duyệt tất cả các đỉnh thuộc cùng mảng liên thông với đỉnh v nên số mảng liên thông p của đồ thị $\mathcal{G}$ chính bằng số lần gọi đến các thủ tục này trong chương trình chính.

Để ghi nhận các đỉnh trong từng mảng liên thông, ta dùng thêm biến mảng $Mang[v]$ để ghi chỉ số của mảng liên thông chứa đỉnh v. Chỉ số này tăng từ 1 đến p.

Ta dùng biến p để đếm số mảng liên thông của đồ thị và gán chỉ số cho các mảng liên thông tìm được. Bắt đầu nó được khởi tạo giá trị bằng 0. Thủ tục $Tham\_dinh(v)$ trong các thủ tục $DFSearch(v)$ và $BFSearch)$ làm thêm nhiệm vụ gán: $Mang[v] := p$;

Còn chương trình chính của thuật toán duyệt cần được sửa lại như sau

Sửa đoạn mã trong phần chính

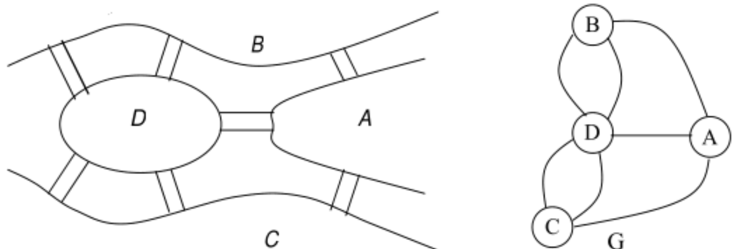
```
begin  // Chương trình chính
    for v ∈ V do Duyet[v] := false;
    p:=0;
    for v ∈ V do
    if !Duyet[v] then
     begin
        p:=p+1;
        DFSearch(v);  //hoặc BFSearch(v);
     end
end.
```

Khi chương trình kết thúc, biến p sẽ cho số mảng liên thông của đồ thị còn các giá trị của biến mảng $Mang[v], v \in V$ cho phép liệt kê tất cả các đỉnh trong từng mảng liên thông của đồ thị.

7.3 Đường đi euler và đồ thị euler

7.3.1. Định nghĩa và tính chất của đồ thị Euler

Có thể coi năm 1736 là năm khai sinh lý thuyết đồ thị, với việc công bố lời giải "bài toán về các cầu ở Konigsberg" của nhà toán học lỗi lạc Euler (1707-1783). Thành phố Konigsberg 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 miền nằm giữa hai nhánh của sông Pregel. Vào thế kỷ 18, người ta xây bảy chiếc cầu nối các vùng này với nhau.



Hình 7.3. Bảy cầu của thành phố Konigsberg

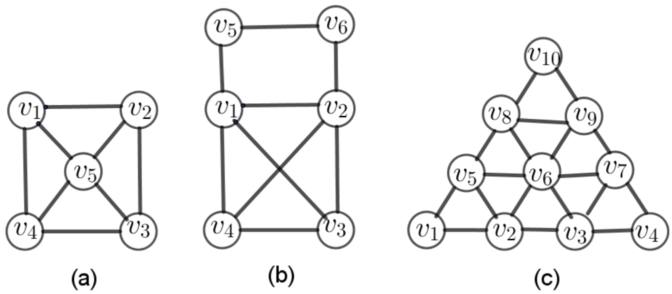
Dân thành phố từng thắc mắc: "Có thể nào đi dạo qua tất cả bảy cầu, mỗi cầu chỉ một lần thôi không?". Nếu ta coi mỗi khu vực A, B, C, D như một đỉnh và mỗi cầu qua lại hai khu vực là một cạnh nối hai đỉnh thì ta có sơ đồ của Konigsberg là một đa đồ thị $\mathcal{G}$ như hình trên.

Bài toán tìm đường đi qua tất cả các cầu, mỗi cầu chỉ qua một lần có thể được phát biểu lại bằng mô hình này như sau: Có tồn tại chu trình đơn trong đa đồ thị $\mathcal{G}$ chứa tất cả các cạnh?

Định nghĩa 7.1. Chu trình (hoặc đường đi) đơn chứa tất cả các cạnh (hoặc cung) của đồ thị (vô hướng hoặc có hướng) $\mathcal{G}$ được gọi là *chu trình (hoặc đường đi) Euler*.

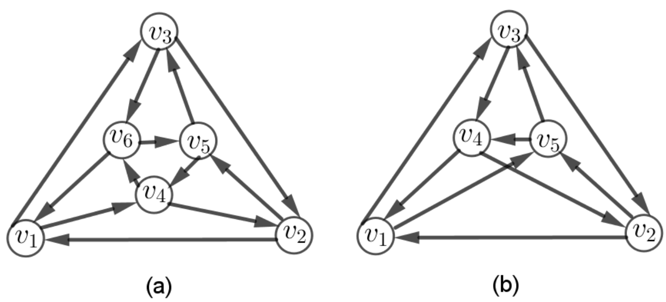
Định nghĩa 7.2. Một đồ thị liên thông (liên thông yếu đối với đồ thị có hướng) có chứa một chu trình (hoặc đường đi) Euler được gọi là *đồ thị Euler (hoặc nửa Euler)*.

Ví dụ 7.3. Các dạng đồ thị Euler và không Euler vô hướng



Hình 7.4. (a) Đồ thị không nửa Euler, (b) Đồ thị nửa Euler, (c) Đồ thị Euler

Ví dụ 7.4. Các dạng đồ thị Euler và không Euler có hướng



Hình 7.5. (a) Đồ thị nửa Euler, (b) Đồ thị nửa Euler

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 hóc búa nổi tiếng thời đó về bảy cái cầu ở Konigsberg và đây là định lý đầu tiên của lý thuyết đồ thị.

Bổ đề 7.1. *Nếu bậc của mỗi đỉnh của đồ thị $\mathcal{G}$ không nhỏ hơn 2 thì $\mathcal{G}$ chứa chu trình đơn.*

Chứng minh. Nếu $\mathcal{G}$ có cạnh bội hoặc có khuyên thì khẳng định của bổ đề là hiển nhiên. Vì vậy giả sử $\mathcal{G}$ là một đơn đồ thị. Gọi v là một đỉnh nào đó của $\mathcal{G}$. Ta sẽ xây dựng theo quy nạp đường đi trong đó v_1 là đỉnh kề với v, còn với $i \geq 1$, chọn v_{i+1} là đỉnh

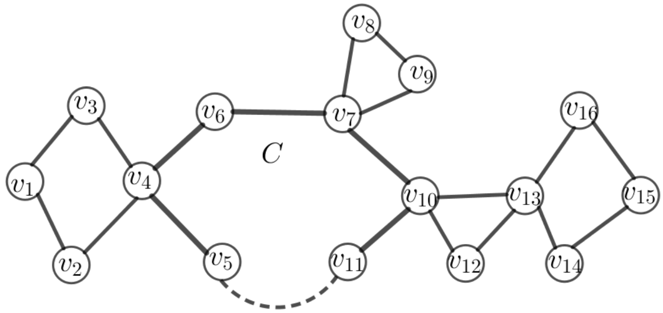
Hình 7.6. Xây dựng đường đi

kề với v_i và $v_{i+1} \neq v_{i-1}$ (có thể chọn như vậy vì $\deg(v_i) \geq 2$), $v_0 = v$. Do tập đỉnh của $\mathcal{G}$ là hữu hạn, nên sau một số hữu hạn bước ta phải quay lại một đỉnh đã xuất hiện trước đó. Gọi k là số nguyên dương đầu tiên để $v_k = v_i (0 \leq i < k)$. Khi đó, đường đi $v_i, v_{i+1}, ..., v_{k-1}, v_k (= v_i)$ là một chu trình đơn cần tìm.

Định lý 7.1. *Đồ thị (vô hướng) liên thông $\mathcal{G}$ là đồ thị Euler khi và chỉ khi mọi đỉnh của $\mathcal{G}$ đều có bậc chẵn.*

Chứng minh. a) *Điều kiện cần:* Giả sử $\mathcal{G}$ là đồ thị Euler, tức là tồn tại chu trình Euler P trong $\mathcal{G}$. Khi đó cứ mỗi lần chu trình P đi qua một đỉnh nào đó của $\mathcal{G}$ thì bậc của đỉnh đó tăng lên 2. Mặt khác, mỗi cạnh của đồ thị xuất hiện trong P đúng một lần. Do đó mỗi đỉnh của đồ thị đều có bậc chẵn.

b) *Điều kiện đủ:* Quy nạp theo số cạnh của $\mathcal{G}$. Do $\mathcal{G}$ liên thông và bậc của mọi đỉnh là chẵn nên mỗi đỉnh có bậc không nhỏ hơn 2. Từ đó theo Bổ đề 7.1, $\mathcal{G}$ phải chứa một chu trình đơn C. Nếu C đi qua tất cả các cạnh của $\mathcal{G}$ thì nó chính là chu trình Euler. Giả sử C không đi qua tất cả các cạnh của $\mathcal{G}$. Khi đó loại bỏ khỏi $\mathcal{G}$ các cạnh thuộc C, ta thu được một đồ thị mới H (không nhất thiết là liên thông). Số cạnh trong H nhỏ hơn trong $\mathcal{G}$ và rõ ràng 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. Do $\mathcal{G}$ liên thông nên mỗi thành phần trong H có ít nhất một đỉnh chung với chu trình C. Vì vậy, ta có thể xây dựng chu trình Euler trong $\mathcal{G}$ như sau: Bắt đầu từ một đỉnh nào đó của chu trình C, đi theo các cạnh của C chừng nào chưa gặp phải đỉnh không cô lập của H. Nếu gặp phải đỉnh như vậy thì ta đi theo chu trình Euler của thành phần liên thông của H chứa đỉnh đó. Sau đó lại tiếp tục đi theo cạnh của C cho đến khi gặp phải đỉnh không cô lập của H thì lại theo chu trình Euler của thành phần liên thông tương ứng trong H, ... Quá trình sẽ kết thúc khi ta trở về đỉnh



Hình 7.7. Xây dựng chu trình Euler

xuất phát, tức là thu được chu trình đi qua mỗi cạnh của đồ thị đúng một lần.

Định lý 7.2. *Đồ thị liên thông $\mathcal{G}$ là nửa Euler (mà không là Euler) khi và chỉ khi có đúng hai đỉnh bậc lẻ trong $\mathcal{G}$.*

Chứng minh. Nếu $\mathcal{G}$ là nửa Euler thì tồn tại một đường đi Euler trong $\mathcal{G}$ từ đỉnh u đến đỉnh v. Gọi $\mathcal{G}'$ là đồ thị thu được từ $\mathcal{G}$ bằng cách thêm vào cạnh 4. Khi đó$\mathcal{G}'$ là đồ thị Euler nên mọi đỉnh trong $\mathcal{G}'$ đều có bậc chẵn (kể cả u và v). Vì vậy u và v là hai đỉnh duy nhất trong $\mathcal{G}$ có bậc lẻ.

Đảo lại, nếu có đúng hai đỉnh bậc lẻ là u và v thì gọi $\mathcal{G}'$ là đồ thị thu được từ $\mathcal{G}$ bằng cách thêm vào cạnh (u, v). Khi đó mọi đỉnh của $\mathcal{G}'$ đều có bậc chẵn hay $\mathcal{G}'$ là đồ thị Euler. Bỏ cạnh (u, v) đã thêm vào ra khỏi chu trình Euler trong $\mathcal{G}'$ ta có được đường đi Euler từ u đến v trong $\mathcal{G}$ hay $\mathcal{G}$ là nửa Euler.

7.3.2. Thuật toán xây dựng chu trình Euler

Dựa trên chứng minh của Định lý 7.1, ta xây dựng thuật toán tìm chu trình Euler cho đồ thị liên thông không có đỉnh bậc lẻ như sau.

Thuật toán 7.4. Tìm chu trình Euler

Đầu vào: Đồ thị liên thông $G = (V, E)$ không có đỉnh bậc lẻ biểu diễn bởimảng các danh sách kề DK.
Đầu ra: Chu trình Euler với danh sách các đỉnh trong CE.
begin

```
S := ∅; CE := ∅;
  v := đỉnh tuỳ ý của đồ thị ;
  push v onto S ;
  while S ≠ ∅ do
      begin
          v := top(S);
          if DK(v) ≠ ∅ then
             begin
                 u := đỉnh đầu tiên trong danh sách DK[v] ;
                 push u onto S ;
                 DK[v] := DK[v] \ {u} ;
                 DK[u] := DK[u] \ {v} ;
                 // Xoá cạnh (v,u) trong đồ thị G
                 v := u
             end
          else
             begin
                 v := top(S) ;
                 push v onto CE
             end
      end
end .
```

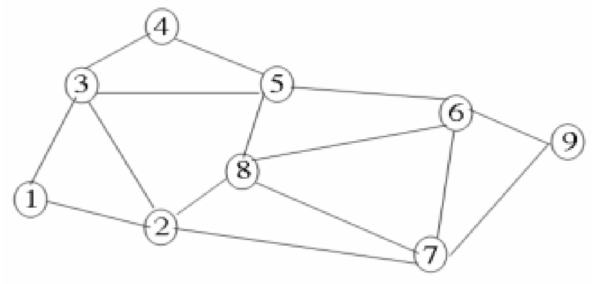
Ta thấy rằng, mỗi lần lặp của chu trình trong thuật toán hoặc là đặt đỉnh lên stack S và xoá cạnh hoặc chuyển đỉnh từ stack S sang stack CE. Số lần lặp của chu trình không vượt quá số cạnh m. Vậy độ phức tạp tổng thể của thuật toán là $O(m)$. Đây là một thuật toán tối ưu để tìm chu trình Euler.

Ví dụ 7.5. Áp dụng thuật toán trên cho đồ thị vô hướng với các đỉnh bậc chẵn dưới đây.

Ta nhận được chu trình Euler là $[1, 2, 3, 4, 5, 6, 7, 2, 8, 6, 9, 7, 8, 5, 3, 1]$.

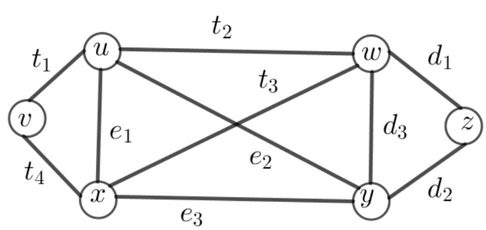
Chú ý 1. Ta có thể sửa đổi thuật toán tìm theo chiều sâu để xây dựng đường đóng các cạnh chưa dùng.

Ví dụ 7.6. Điều quan trọng của thuật toán là mở rộng những đường mòn đóng bằng ghép nối đường đóng thứ hai. Ta xét hai



Hình 7.8. Đồ thị vô hướng với các đỉnh bậc chẵn

đường đóng $T =< t_1, t_2, t_3, t_4 >$ và $D =< d_1, d_2, d_3 >$ như trong hình 7.9.



Hình 7.9. Thuật toán Euler

Khi đó $T' =< t_1, t_2, t_3, d_1, d_2, d_3, t_3, t_4 >$;

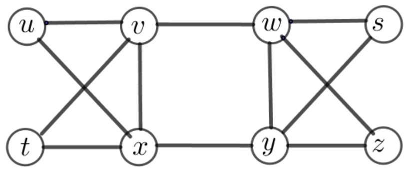
Bước lặp tiếp theo gép nối với $< e_1, e_2, e_3 >$ với T' và cho ra kết quả chu trình Euler của toàn đồ thị.

Chú ý 2. Một phương pháp khác tìm một chu trình Euler trong đồ thị liên thông $\mathcal{G}$ có bậc của mọi đỉnh là chẵn theo thuật toán Fleury sau đây.

Xuất phát từ một đỉnh bất kỳ của $\mathcal{G}$ và tuân theo hai quy tắc sau

1. Mỗi khi đi qua một cạnh nào thì xoá nó đi; sau đó xoá đỉnh cô lập (nếu có);
2. Không bao giờ đi qua một cầu, trừ phi không còn cách đi nào khác.

Ví dụ 7.7. Tìm chu trình Euler theo thuật toán Fleury đồ thị sau



Hình 7.10. Tìm chu trình Euler

Xuất phát từ u, ta có thể đi theo cạnh (u, v) hoặc (u, x), giả sử là (u, v) (xoá (u, v)). Từ v có thể đi qua một trong các cạnh $(v, w), (v, x), (v, t)$, giả sử (v, w) (xoá (v, w)). Tiếp tục, có thể đi theo một trong các cạnh $(w, s), (w, y), (w, z)$, giả sử(w, s) (xoá (w, s)). Đi theo cạnh (s, y) (xoá (s, y) và s). Vì (y, x) là cầu nên có thể đi theo một trong hai cạnh $(y, w), (y, z)$, giả sử (y, w) (xoá (y, w)). Đi theo (w, z) (xoá 4 và w) và theo (z, y) (xoá (z, y) và z). Tiếp tục đi theo cạnh (y, x) (xoá (y, x) và y). Vì (x, u) là cầu nên đi theo cạnh (x, v) hoặc 4, giả sử (x, v) (xoá (x, v)). Tiếp tục đi theo cạnh 4(xoá (v, t) và v), theo cạnh 4 (xoá cạnh(t, x) và 4), cuối cung đi theo cạnh (x, u) (xoá (x, u), x và u).

7.3.3. Bài toán người phát thư Trung Hoa

Một nhân viên đi từ Sở Bưu Điện, qua một số đường phố để phát thư, rồi quay về Sở. Người ấy phải đi qua các đường theo trình tự nào để đường đi là ngắn nhất?

Bài toán được nhà toán học Trung Hoa Guan nêu lên đầu tiên (1960), vì vậy thường được gọi là "bài toán người phát thư Trung Hoa". Ta xét bài toán ở một dạng đơn giản như sau.

Cho đồ thị liên thông $\mathcal{G}$. Một chu trình qua mọi cạnh của $\mathcal{G}$ gọi là một hành trình trong $\mathcal{G}$. Trong các hành trình đó, hãy tìm hành trình ngắn nhất, tức là qua ít cạnh nhất.

Rõ ràng rằng nếu $\mathcal{G}$ là đồ thị Euler (mọi đỉnh đều có bậc chẵn) thì chu trình Euler trong $\mathcal{G}$ (qua mỗi cạnh của $\mathcal{G}$ đúng một lần) là hành trình ngắn nhất cần tìm.

Chỉ còn phải xét trường hợp $\mathcal{G}$ có một số đỉnh bậc lẻ (số đỉnh bậc lẻ là một số chẵn). Khi đó, mọi hành trình trong $\mathcal{G}$ phải đi qua ít nhất hai lần một số cạnh nào đó.

Dễ thấy rằng một hành trình qua một cạnh (u, v) nào đó quá hai lần thì không phải là hành trình ngắn nhất trong $\mathcal{G}$. Vì vậy, ta chỉ cần xét những hành trình T đi qua hai lần một số cạnh nào đó của $\mathcal{G}$.

Ta quy ước xem mỗi hành trình T trong $\mathcal{G}$ là một hành trình trong đồ thị Euler $\mathcal{G}_T$, có được từ $\mathcal{G}$ bằng cách vẽ thêm một cạnh song song đối với những cạnh mà T đi qua hai lần. Bài toán đặt ra được đưa về bài toán sau:

Trong các đồ thị Euler $\mathcal{G}_T$, tìm đồ thị có số cạnh ít nhất (khi đó chu trình Euler trong đồ thị này là hành trình ngắn nhất).

Định lý 7.3 (Gooodman và Hedetniemi, 1973). *Nếu $\mathcal{G}$ là một đồ thị liên thông có q cạnh thì hành trình ngắn nhất trong $\mathcal{G}$ có chiều dài*

$$q + m(\mathcal{G}),$$

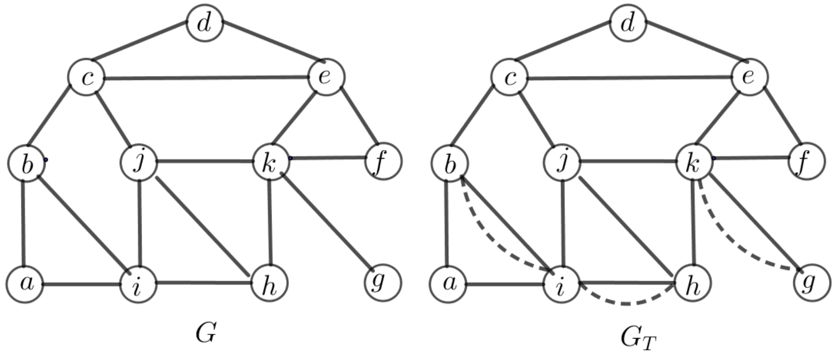
trong đó $m(\mathcal{G})$ là số cạnh mà hành trình đi qua hai lần và được xác định như sau:

1. *Gọi $V_0(\mathcal{G})$ là tập hợp các đỉnh bậc lẻ ($2k$ đỉnh) của $\mathcal{G}$. Ta phân $2k$ phần tử của $\mathcal{G}$ thành k cặp, mỗi tập hợp k cặp gọi là một phân hoạch cặp của $V_0(\mathcal{G})$.*
2. *Ta gọi độ dài đường đi ngắn nhất từ u đến v là khoảng cách $d(u, v)$. Đối với mọi phân hoạch cặp P_i, ta tính khoảng cách giữa hai đỉnh trong từng cặp, rồi tính tổng $d(P_i)$. Số $m(\mathcal{G})$bằng cực tiểu của các $d(P_i)$:*

$$m(\mathcal{G}) = \min d(Pi).$$

Ví dụ 7.8. Giải bài toán người phát thư Trung Hoa cho trong đồ thị sau

Lời giải. Tập hợp các đỉnh bậc lẻ $V_0(\mathcal{G}) = \{b, g, h, k\}$ và tập hợp



Hình 7.11. Bài toán người phát thư

các phân hoạch cặp là $P = P_1, P_2, P_3$, trong đó

$$P_1 = \{(b,g),(h,k)\} \to d(P_1) = d(b,g) + d(h,k) = 4 + 1 = 5,$$
$$P_2 = \{(b,h),(g,k)\} \to d(p_2) = d(b,h) + d(g,k) = 2 + 1 = 3,$$
$$P_3 = \{(b,k),(g,h)\} \to d(p_3) = d(b,k) + d(g,h) = 3 + 2 = 5.$$

$m(\mathcal{G}) = \min(d(P_1), d(P_2), d(P_3)) = 3.$

Do đó $\mathcal{G}_T$ có được từ $\mathcal{G}$ bằng cách thêm vào 3 cạnh: $(b,i),(i,h),(g,k)$ và $\mathcal{G}_T$ là đồ thị Euler. Vậy hành trình ngắn nhất cần tìm là đi theo chu trình Euler trong $\mathcal{G}_T$:

$$a, b, c, d, e, f, k, g, k, e, c, j, k, h, j, i, h, i, b, i, a.$$

7.3.4. Đường Euler trong đồ thị định hướng

Thuật ngữ về đường Euler trong đồ thị định hướng là đường mòn có hướng. Các đặc trưng của đồ thị cho hành trình Euler và đường Euler mở vẫn còn đúng trong đồ thị định hướng còn đúng với các đường định hướng này. Ta phát biểu các định lí cho đồ thị định hướng, còn chứng minh thì hoàn toàn tương tự.

Định lý 7.4. *Đồ thị có hướng liên thông yếu $\mathcal{G}$ là đồ thị Euler khi và chỉ khi mọi đỉnh của $\mathcal{G}$ đều có bậc vào bằng bậc ra.*

Chứng minh. Chứng minh tương tự như chứng minh của Định lý 7.1 và điều kiện đủ cũng cần có bổ đề dưới đây tương tự như ở Bổ đề 7.1.

Bổ đề 7.2. *Nếu bậc vào và bậc ra của mỗi đỉnh của đồ thị có hướng $\mathcal{G}$ không nhỏ hơn 1 thì $\mathcal{G}$ chứa chu trình đơn.*

Hệ quả 7.1. *Đồ thị có hướng liên thông yếu $\mathcal{G}$ là nửa Euler (mà không là Euler) khi và chỉ khi tồn tại hai đỉnh x và y sao cho*

$\deg_o(x) = \deg_t(x) + 1, \deg_t(y) = \deg_o(y) + 1, \deg_t(v) = \deg_o(v)$, $\forall v \in V, v \neq x, v \neq y$.

Chứng minh. Chứng minh tương tự như ở Hệ quả 7.1.

7.4 Đường đi và đồ thị Hamilton

Năm 1857, nhà toán học người Ailen là Hamilton(1805-1865) đưa ra trò chơi "đi vòng quanh thế giới" như sau.

Cho một hình thập nhị diện đều (đa diện đều có 12 mặt, 20 đỉnh và 30 cạnh), mỗi đỉnh của hình mang tên một thành phố nổi tiếng, mỗi cạnh của hình (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 trở về chỗ cũ.

Trước Hamilton, có thể là từ thời Euler, người ta đã biết đến một câu đố hóc búa về "đường đi của con mã trên bàn cờ". Trên bàn cờ, con mã chỉ có thể đi theo đường chéo của hình chữ nhật 2×3 hoặc 3×2 ô vuông. Giả sử bàn cờ có 8×8 ô vuông. Hãy tìm đường đi của con mã qua được tất cả các ô của bàn cờ, mỗi ô chỉ một lần rồi trở lại ô xuất phát.

Bài toán này được nhiều nhà toán học chú ý, đặc biệt là Euler, De Moivre, Vandermonde, ...

Hiện nay đã có nhiều lời giải và phương pháp giải cũng có rất nhiều, trong đó có quy tắc: mỗi lần bố trí con mã ta chọn vị trí mà tại vị trí này số ô chưa dùng tới do nó khống chế là ít nhất.

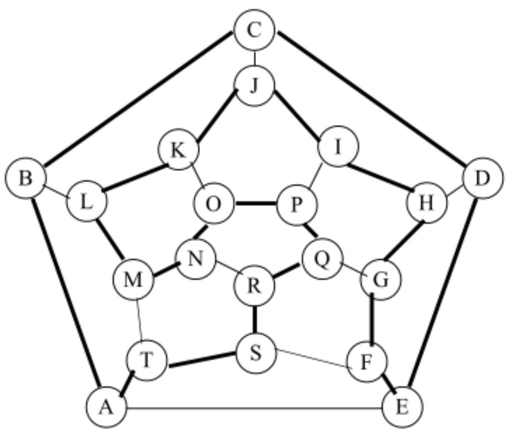
Một phương pháp khác dựa trên tính đối xứng của hai nửa bàn cờ. Ta tìm hành trình của con mã trên một nửa bàn cờ, rồi lấy đối xứng cho nửa bàn cờ còn lại, sau đó nối hành trình của hai nửa đã tìm lại với nhau.

Trò chơi và câu đố trên dẫn tới việc khảo sát một lớp đồ thị đặc biệt, đó là đồ thị Hamilton.

7.4.1. Định nghĩa và tính chất đồ thị Hamilton

Định nghĩa 7.3. Chu trình (t.ư. đường đi) sơ cấp chứa tất cả các đỉnh của đồ thị (vô hướng hoặc có hướng) $\mathcal{G}$ được gọi là chu trình (t.ư. đường đi) Hamilton. Một đồ thị có chứa một chu trình (t.ư. đường đi) Hamilton được gọi là đồ thị Hamilton (t.ư. nửa Hamilton).

Ví dụ 7.9. Đồ thị Hamilton (hình thập nhị diện đều biểu diễn trong mặt phẳng) với chu trình Hamilton A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, A (Hình 7.12 đường tô đậm).



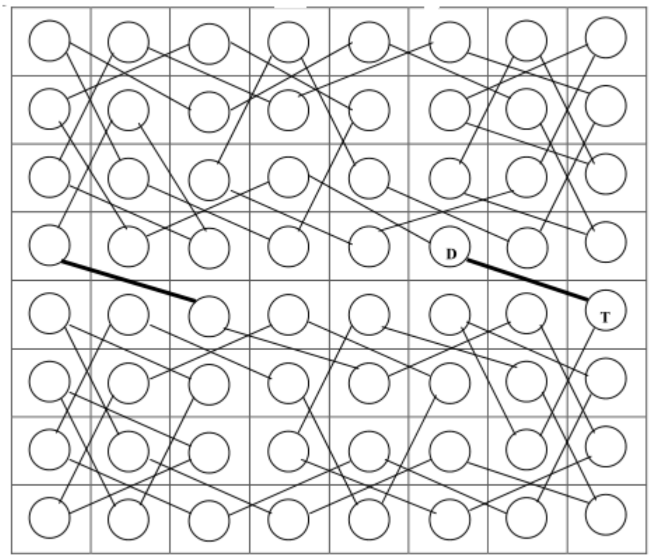
Hình 7.12. Hình thập nhị diện đều

Ví dụ 7.10. Trong một đợt thi đấu bóng bàn có $n(n \geq 2)$ đấu thủ tham gia. Mỗi đấu thủ gặp từng đấu thủ khác đúng một lần. Trong thi đấu bóng bàn chỉ có khả năng thắng hoặc thua.

Chứng minh rằng sau đợt thi đấu có thể xếp tất cả các đấu thủ đứng thành một hàng dọc, để người đứng sau thắng người đứng ngay trước anh (chị) ta.

Xét đồ thị có hướng $\mathcal{G}$ gồm n đỉnh sao cho mỗi đỉnh ứng với một đấu thủ và có một cung nối từ đỉnh u đến đỉnh v nếu đấu thủ ứng với u thắng đấu thủ ứng với v. Như vậy, đồ thị $\mathcal{G}$ có tính chất là với hai đỉnh phân biệt bất kỳ u và v, có một và chỉ một trong hai cung (u, v) hoặc (v, u), đồ thị như thế được gọi là đồ thị có hướng đầy đủ. Từ Định lý 7.5 dưới đây, $\mathcal{G}$ là một đồ thị nửa Hamilton. Khi đó đường đi Hamilton trong $\mathcal{G}$ cho ta sự sắp xếp cần tìm.

Ví dụ 7.11. Một lời giải về hành trình của con mã trên bàn cờ 8×8:



Hình 7.13. Hành trình của con mã

Đường đi Hamilton tương tự đường đi Euler trong cách phát biểu: Đường đi Euler qua mọi cạnh (cung) của đồ thị đúng một lần, đường đi Hamilton qua mọi đỉnh của đồ thị đúng một lần. 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 về tìm đường đi Hamilton và xác định đồ thị Hamilton lại khó hơn rất nhiều. Đường đi Hamilton và đồ thị Hamilton có nhiều ý nghĩa thực tiễn và đã được nghiên cứu nhiều, nhưng vẫn còn những khó khăn lớn chưa ai vượt qua được.

Người ta chỉ mới tìm được một vài điều kiện đủ để nhận biết một lớp rất nhỏ các đồ thị Hamilton và đồ thị nửa Hamilton. Sau đây là một vài kết quả.

Định lý 7.5 (Rédei). *Nếu $\mathcal{G}$ là một đồ thị có hướng đầy đủ thì $\mathcal{G}$ là đồ thị nửa Hamilton.*

Chứng minh. Giả sử $\mathcal{G} = (V, E)$ là đồ thị có hướng đầy đủ và $d = (v_1, v_2, ..., v_{k-1}, v_k)$là đường đi sơ cấp bất kỳ trong đồ thị $\mathcal{G}$.

– Nếu d đã đi qua tất cả các đỉnh của $\mathcal{G}$ thì nó là một đường đi Hamilton của $\mathcal{G}$.

– Nếu trong $\mathcal{G}$ còn có đỉnh nằm ngoài d, thì ta có thể bổ sung dần các đỉnh này vào d và cuối cùng nhận được đường đi Hamilton.

Thật vậy, giả sử v là đỉnh tuỳ ý không nằm trên d.

a) Nếu có cung nối v với v_1 thì bổ sung v vào đầu của đường đi d để được $d_1 = (v, v_1, v_2, ..., v_{k-1}, v_k)$.

b) Nếu tồn tại chỉ số $i (1 \le i \le k-1)$ mà từ v_i có cung nối tới v và từ v có cung nối tới v_{i+1} thì ta chen v vào giữa v_i và v_{i+1} để được đường đi sơ cấp $d_2 = (v_1, v_2, ..., v_i, v, v_{i+1}, ..., v_k)$.

c) Nếu cả hai khả năng trên đều không xảy ra nghĩa là với mọi $i (1 \le i \le k) v_i$ đều có cung đi tới v. Khi đó bổ sung v vào cuối của đường đi d và được đường đi $d_3 = (v_1, v_2, ..., v_{k-1}, v_k, v)$.

Nếu đồ thị $\mathcal{G}$ có n đỉnh thì sau $n - k$ bổ sung ta sẽ nhận được đường đi Hamilton.

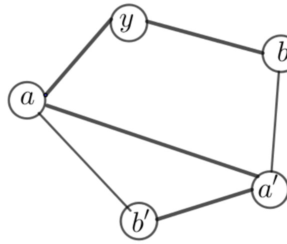
Định lý 7.6 (Dirac, 1952). *Nếu $\mathcal{G}$ là một đơn đồ thị có n đỉnh và mọi đỉnh của $\mathcal{G}$ đều có bậc không nhỏ hơn $\frac{n}{2}$ thì $\mathcal{G}$ là một đồ thị Hamilton.*

Chứng minh.

Gọi P là chu trình Hamilton $ayb...a$ trong $\mathcal{G}'$, trong đó a và b là các đỉnh của $\mathcal{G}$, còn y là một trong các đỉnh mới. Khi đó b không kề với a, vì nếu trái lại thì ta có thể bỏ đỉnh y và được chu trình $ab...a$, mâu thuẩn với giả thiết về tính chất nhỏ nhất của k.

Ngoài ra, nếu a' là một đỉnh kề nào đó của a (khác với y) và b' là đỉnh nối tiếp ngaya' trong chu trình P thì b' không thể là

Định lý được chứng minh bằng phản chứng. Giả sử $\mathcal{G}$ không có chu trình Hamilton. Ta thêm vào $\mathcal{G}$ một số đỉnh mới và nối mỗi đỉnh mới này với mọi đỉnh của $\mathcal{G}$, ta được đồ thị $\mathcal{G}'$. Giả sử $k(>0)$ là số tối thiểu các đỉnh cần thiết để $\mathcal{G}'$ chứa một chu trình Hamilton. Như vậy, $\mathcal{G}'$ có $n+k$ đỉnh.



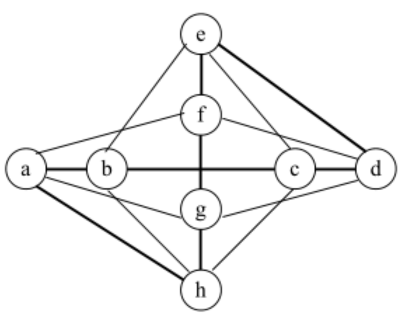
Hình 7.14. Chứng minh Định lý Dirac

đỉnh kề với b, vì nếu trái lại thì ta có thể thay P bởi chu trình $aa'...bb'...a$, trong đó không có y, mâu thuẫn với giả thiết về tính chất nhỏ nhất của k.

Như vậy, với mỗi đỉnh kề với a, ta có một đỉnh không kề với b, tức là số đỉnh không kề với b không thể ít hơn số đỉnh kề với a (số đỉnh kề với a không nhỏ hơn $\frac{n}{2}+k$).

Mặt khác, theo giả thiết số đỉnh kề với b cũng không nhỏ hơn $\frac{n}{2}+k$. Vì không có đỉnh nào vừa kề với b lại vừa không kề với b, nên số đỉnh của $\mathcal{G}'$ không ít hơn $2(\frac{n}{2}+k) = n+2k$, mâu thuẫn với giả thiết là số đỉnh của $\mathcal{G}'$ bằng $n+k(k>0)$. Định lý được chứng minh.

Ví dụ 7.12. Xét đồ thị $\mathcal{G}=(V,E)$, $V=\{a,b,c,d,e,f,g,h\}$,
$E=\{(a,b),(a,f),(a,g),(a,h),(b,e),(b,c),(b,h),(e,g),(e,c),(e,d),$
$(e,f),(f,d),(f,g),(g,d),(g,h),(h,c),(c,d)\}$



Hình 7.15. Chu trình Hamilton theo Định lý 7.6

Đồ thị $\mathcal{G}$ (Hình 7.15) có 8 đỉnh, đỉnh nào cũng có bậc 4, nên theo Định lý 7.6, $\mathcal{G}$ là đồ thị Hamilton.

Hệ quả 7.2. *Nếu $\mathcal{G}$ là đơn đồ thị có n đỉnh và mọi đỉnh của $\mathcal{G}$ đều có bậc không nhỏ hơn $\frac{n-1}{2}$ thì $\mathcal{G}$ là đồ thị nửa Hamilton.*

Chứng minh. Thêm vào $\mathcal{G}$ một đỉnh x và nối x với mọi đỉnh của $\mathcal{G}$ thì ta nhận được đơn đồ thị $\mathcal{G}'$ có $n+1$ đỉnh và mỗi đỉnh có bậc không nhỏ hơn $\frac{n+1}{2}$. Do đó theo Định lý 7.6, trong $\mathcal{G}'$ có một chu trình Hamilton. Bỏ x ra khỏi chu trình này, ta nhận được đường đi Hamilton trong $\mathcal{G}$.

7.4.2. Điều kiện đủ cho đồ thị là Hamilton

Định lý 7.7 (Ore, 1960). *Cho $\mathcal{G}$ là một đồ thị n đỉnh đơn với $n \geq 3$ sao cho $\deg(X) + \deg(Y) \geq n$ với mọi cặp hai đỉnh X và Y không cạnh nhau. Khi đó $\mathcal{G}$ là Hamilton.*

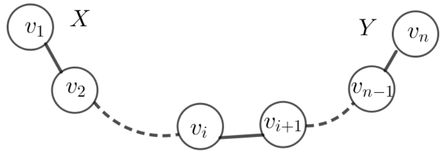
Chứng minh. Ta chứng minh bằng phản chứng, giả sử rằng định lí sai và cho $\mathcal{G}$ là đồ thị lớn nhất theo n có thể mà định lí sai. Nghĩa là $\mathcal{G}$ không có đường Hamilton và vẫn thỏa mãn các điều kiện của định lí, và ta thêm một cạnh nối hai đỉnh không kề nhau của $\mathcal{G}$ tạo thành một đồ thị Hamilton.

Cho X và Y là hai đỉnh không cạnh nhau của $\mathcal{G}$ (vì $\mathcal{G}$ không đầy đủ và $n \geq 3$). Để chứng minh có điều vô lí, ta cần chỉ ra $\deg(X) + \deg(Y) \leq n-1$ là đủ.

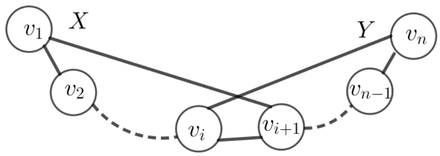
Thật vậy, từ đồ thị $\mathcal{G} + XY$ chứa chu trình Hamilton, $\mathcal{G}$ chứa đường dẫn Hamilton, mà các điểm cuối của nó là X và Y. Đặt $< X = v_1, v_2, ..., v_n = Y >$ là đường dẫn (hình 7.16)

Với mỗi $i = 2, 3, ..., n-1$ ít nhất một trong các cặp v_1, v_{i+1} và v_i, v_n không kề nhau, vì nếu ngược lại thì $v_1, v_2, ..., v_i, v_n, v_{n-1}, ..., v_{i+1}, v_1 >$ sẽ là chu trình Hamilton trong $\mathcal{G}$ (hình 7.17).

Điều này nghĩa là nếu (a_{ij}) là ma trận kề cho $\mathcal{G}$, thì $a_{1,i+1} +$



Hình 7.16. Đường dẫn Hamilton trong $\mathcal{G}$



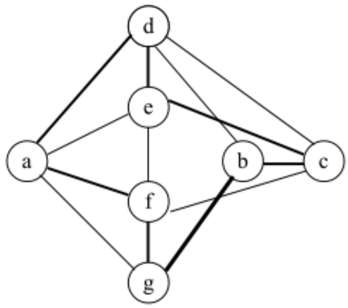
Hình 7.17. Đường dẫn Hamilton trong $\mathcal{G}$

$a_{i,n} \leq 1$ với $i = 2, 3, ..., n-2$. Như vậy,

$$\begin{aligned}
\deg(X) + \deg(Y) &= \sum_{i=2}^{n-1} a_{1,i} + \sum_{i=2}^{n-1} a_{i,n} \\
&= a_{1,2} + \sum_{i=3}^{n-1} a_{1,i} + \sum_{i=2}^{n-2} a_{i,n} + a_{n-1,n} \\
&= 1 + \sum_{i=3}^{n-1} a_{1,i} + \sum_{i=2}^{n-2} a_{i,n} + 1 \\
&= 2 + \sum_{i=3}^{n-1} (a_{1,i} + a_{i,n}) \\
&\leq 2 + n - 3 = n - 1.
\end{aligned}$$

Điều này cần chứng minh như chú ý trên.

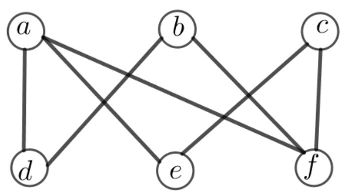
Ví dụ 7.13. Đồ thị $\mathcal{G}'$ (Hình 7.18) có 5 đỉnh bậc 4 và 2 đỉnh bậc 2 kề nhau nên tổng số bậc của hai đỉnh không kề nhau bất kỳ bằng 7 hoặc 8, nên theo Định lý 7.7, $\mathcal{G}'$ là đồ thị Hamilton.



Hình 7.18. Chu trình Hamilton theo Định lý 7.7.

Định lý 7.8. *Nếu $\mathcal{G}$ là đồ thị phân đôi với hai tập đỉnh là $V1, V_2$ có số đỉnh cùng bằng $n(n \geq 2)$ và bậc của mỗi đỉnh lớn hơn $\frac{n}{2}$ thì $\mathcal{G}$ là một đồ thị Hamilton.*

Ví dụ 7.14. Đồ thị phân đôi này có bậc của mỗi đỉnh bằng 2 hoặc $3(> 3/2)$, nên theo Định lý 7.8, nó là đồ thị Hamilton.



Hình 7.19. Chu trình Hamilton theo Định lý 7.7.

7.4.3. Bài toán sắp xếp chỗ ngồi

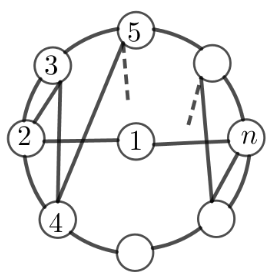
Có n đại biểu từ n nước đến dự hội nghị quốc tế. Mỗi ngày họp một lần ngồi quanh một bàn tròn. Hỏi phải bố trí bao nhiêu ngày và bố trí như thế nào sao cho trong mỗi ngày, mỗi người có hai người kế bên là bạn mới. Lưu ý rằng n người đều muốn làm quen với nhau.

Xét đồ thị gồm n đỉnh, mỗi đỉnh ứng với mỗi người dự hội nghị, hai đỉnh kề nhau khi hai đại biểu tương ứng muốn làm quen với nhau. Như vậy, ta có đồ thị đầy đủ K_n . Đồ thị này là

Hamilton và rõ ràng mỗi chu trình Hamilton là một cách sắp xếp như yêu cầu của bài toán. Bái toán trở thành tìm các chu trình Hamilton phân biệt của đồ thị đầy đủ K_n (hai chu trình Hamilton gọi là phân biệt nếu chúng không có cạnh chung).

Định lý 7.9. *Đồ thị đầy đủ K_n với n lẻ và $n \geq 3$ có đúng $\dfrac{n-1}{2}$ chu trình Hamilton phân biệt.*

Chứng minh. K_n có $\dfrac{n(n-1)}{2}$ cạnh và mỗi chu trình Hamilton có n cạnh, nên số chu trình Hamilton phân biệt nhiều nhất là $\dfrac{n-1}{2}$.(Hình 7.20)



Hình 7.20. Chu trình Hamilton.

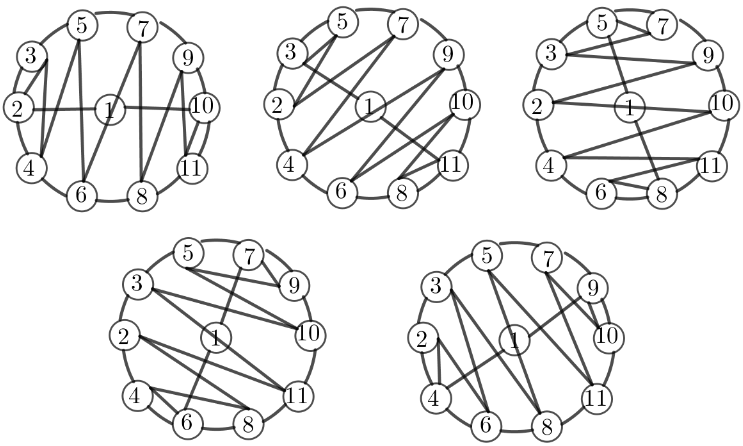
Giả sử các đỉnh của K_n là $1, 2, ..., n$. Đặt đỉnh 1 tại tâm của một đường tròn và các đỉnh $2, ..., n$ đặt cách đều nhau trên đường tròn (mỗi cung là $\dfrac{360^\circ}{n-1}$ sao cho đỉnh lẻ nằm ở nửa đường tròn trên và đỉnh chẵn nằm ở nửa đường tròn dưới. Ta có ngay chu trình Hamilton đầu tiên là $1, 2, ..., n, 1$. Các đỉnh được giữ cố định, xoay khung theo chiều kim đồng hồ với các góc quay:

$$\frac{360^\circ}{n-1}, 2.\frac{360^\circ}{n-1}, 3.\frac{360^\circ}{n-1}, ..., \frac{n-3}{2}.\frac{360^\circ}{n-1},$$

ta nhận được $\dfrac{n-3}{2}$ khung phân biệt với khung đầu tiên. Do đó ta có $\dfrac{n-1}{2}$ chu trình Hamilton phân biệt.

Ví dụ 7.15. Giải bài toán sắp xếp chỗ ngồi với $n = 11$.

Lời giải. Có $\dfrac{11-1}{2} = 5$ cách sắp xếp chỗ ngồi phân biệt như sau



Hình 7.21. Bài toán sắp xếp chỗ ngồi.

1	2	3	4	5	6	7	8	9	10	11	1
1	3	5	2	7	4	9	6	11	8	10	1
1	5	7	3	9	2	11	4	10	6	8	1
1	7	9	5	11	3	10	2	8	4	6	1
1	9	11	7	10	5	8	3	6	2	4	1

7.4.4. Đồ thị không phải là Hamilton

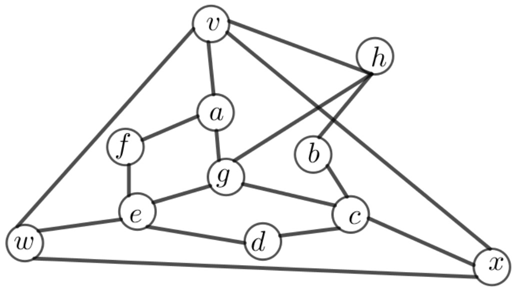
Các quy tắc sau đây như các định lí là cơ sở chỉ ra rằng mọi chu trình Hamilton phải chứa đúng hai cạnh gắn liền với mỗi đỉnh.

Quy tắc 1: Nếu đỉnh V có bậc 2, thì cả hai cạnh gắn với nó phải là một phần của chu trình Hamilton bất kì.

Quy tắc 2: Trong quá trình xây dựng chu trình Hamilton, không có chu trình có thể tạo thành cho tới khi tất cả các đỉnh đã được ghé qua.

Quy tắc 3: Nếu trong quá trình xây dựng chu trình Hamilton hai trong số các cạnh gắn liền với 1 đỉnh V được chỉ ra, thì tất cả các cạnh gắn liền khác có thể bỏ đi được.

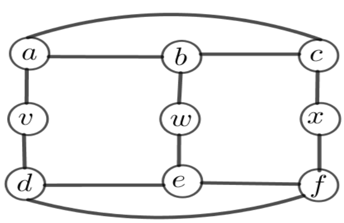
Ví dụ 7.16. Chứng minh rằng đồ thị dưới đây không phải là đồ thị Hamilton (hình 7.22)



Hình 7.22. Đồ thị không là đồ thị Hamilton

Chứng minh. Quy tắc 1 áp dụng cho các đỉnh b, d, f suy ra các cạnh bh, bc, dc, de, ef và fa phải nằm trên mọi chu trình Hamilton. Theo nguyên tắc 3 áp dụng cho đỉnh c và e loại trừ các cạnh cg, cx, eg và ew. Kết quả còn lại của đồ thị lại sử dụng quy tắc 1 suy ra các cạnh wv, vx và xw phải nằm trên mọi chu trình Hamilton. Nhưng điều đó tạo ra một 3-chu trình, mà nó vô lí với quy tắc 2. Điều đó kéo theo tồn tại hai chu trình con tách biệt, mà nó lại trái với quy tắc 2. □

Ví dụ 7.17. Chứng minh rằng đồ thị sau không phải là Hamilton (hình 7.23)



Hình 7.23. Đồ thị không là đồ thị Hamilton

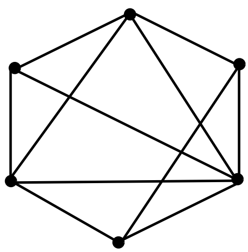
Chứng minh. Quy tắc 1 áp dụng cho v, w, x kéo theo tất cả sáu cạnh đứng phải là một phần chu trình Hamilton bất kì. Quy tắc

1 và 3 áp dụng cho đỉnh b kéo theo đúng một trong các cạnh ab và bc là một Phần của chu trình Hamilton. Nếu ab nằm trên chu trình và bc không thì theo quy tắc 3 áp dụng cho đỉnh a kéo theo ac không nằm trên chu trình. Nhưng khi đó cx chỉ là cạnh trên chu trình này mà nó gắn liền với c. Như vậy theo quy tắc 1 không có chu trình tồn tại trong trường hợp này, vô lí. Theo tính đối xứng, tương tự cho kết quả với nếu bc trên chu trình còn ab thì không. □

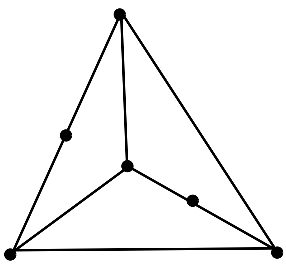
7.5 Bài tập

▷ **7.1.** Hãy cho một đồ thị nào đó. Duyệt theo chiều sâu, rồi duyệt theo chiều rộng trên đồ thị này.

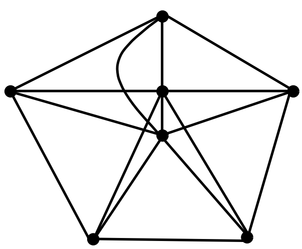
▷ **7.2.** Duyệt theo chiều sâu, rồi duyệt theo chiều rộng trên các đồ thị sau đây.



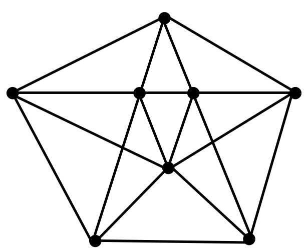
(a)



(b)



(c)



(d)

▷ **7.3.** Đồ thị mà kết quả duyệt theo chiều sâu và duyệt theo chiều rộng luôn trùng nhau, có những tính chất gì?

▷ **7.4.** Với giá trị nào của n các đồ thị sau đây có chu trình Euler ?
a) Kn, b) Cn, c) Wn, d) Qn.

▷ **7.5.** Với giá trị nào của m và n các đồ thị phân đôi đầy đủ $K_{m,n}$ có

a) chu trình Euler ? b) đường đi Euler ?

▷ **7.6.** Với giá trị nào của m và n các đồ thị phân đôi đầy đủ $K_{m,n}$ có chu trình Hamilton ?

▷ **7.7.** Chứng minh rằng đồ thị lập phương Q_n là một đồ thị Hamilton. Vẽ cây liệt kê tất cả các chu trình Hamilton của đồ thị lập phương Q_3 .

▷ **7.8.** Trong một cuộc họp có 15 người mỗi ngày ngồi với nhau quanh một bàn tròn một lần. Hỏi có bao nhiêu cách sắp xếp sao cho mỗi lần ngồi họp, mỗi người có hai người bên cạnh là bạn mới, và sắp xếp như thế nào ?

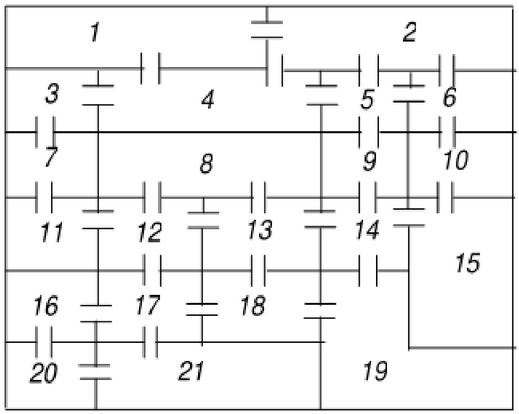
▷ **7.9.** Hiệu trưởng mời $2n(n \geq 2)$ sinh viên giỏi đến dự tiệc. Mỗi sinh viên giỏi quen ít nhất n sinh viên giỏi khác đến dự tiệc. Chứng minh rằng luôn luôn có thể xếp tất cả các sinh viên giỏi ngồi xung quanh một bàn tròn, để mỗi người ngồi giữa hai người mà sinh viên đó quen.

▷ **7.10.** Một ông vua đã xây dựng một lâu đài để cất báu vật. Người ta tìm thấy sơ đồ của lâu đài (hình sau) với lời dặn: muốn tìm báu vật, chỉ cần từ một trong các phòng bên ngoài cùng (số 1, 2, 6, 10, ...), đi qua tất cả các cửa phòng, mỗi cửa chỉ một lần; báu vật được giấu sau cửa cuối cùng.

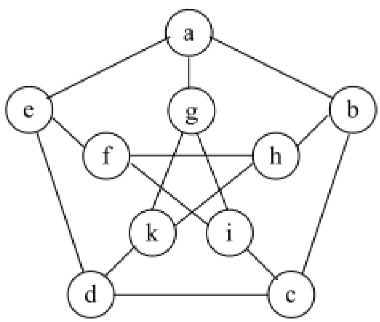
Hãy tìm nơi giấu báu vật

▷ **7.11.** Đồ thị cho trong hình sau gọi là đồ thị Peterson P (Hình 7.25).

a) Tìm một đường đi Hamilton trong P.



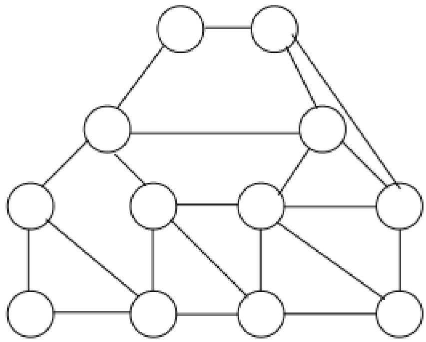
Hình 7.24. Lâu đài bí mật



Hình 7.25. Đồ thị Peterson P

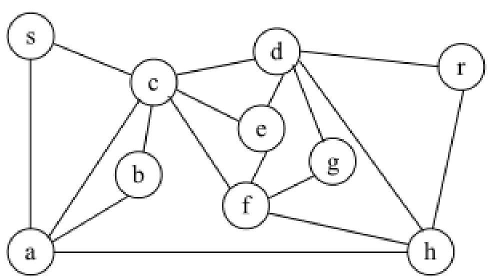
b) Chứng minh rằng$P \setminus \{v\}$, với v là một đỉnh bất kỳ của P, là một đồ thị Hamilton.

▷ **7.12.** Giải bài toán người phát thư Trung Hoa với đồ thị cho trong hình sau:



Hình 7.26. Bài toán người phát thư P

▷ **7.13.** Chứng minh rằng đồ thị $\mathcal{G}$ cho trong hình sau có đường đi Hamilton (từ s đến r) nhưng không có chu trình Hamilton.



Hình 7.27. Không có chu trình

▷ **7.14.** Cho thí dụ về:

1) Đồ thị có một chu trình vừa là chu trình Euler vừa là chu trình Hamilton;

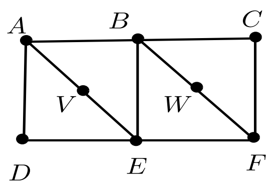
2) Đồ thị có một chu trình Euler và một chu trình Hamilton, nhưng hai chu trình đó không trùng nhau;

3) Đồ thị có 6 đỉnh, là đồ thị Hamilton, nhưng không phải là đồ thị Euler;

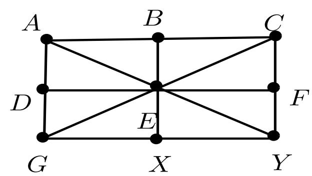
4) Đồ thị có 6 đỉnh, là đồ thị Euler, nhưng không phải là đồ thị Hamilton.

▷ **7.15.** Chứng minh rằng con mã không thể đi qua tất cả các ô của một bàn cờ có 4×4 hoặc 5×5 ô vuông, mỗi ô chỉ một lần, rồi trở về chỗ cũ.

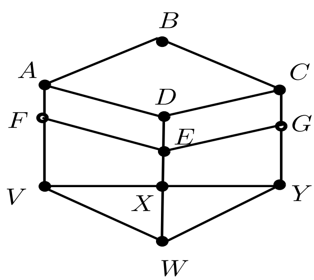
▷ **7.16.** Chứng minh rằng các đồ thị sau đây có chu trình Hamilton hoặc không phải là đồ thị Hamilton.



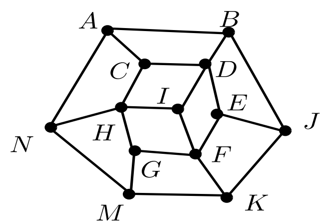
(a)



(b)



(c)



(d)

Chương

8 BÀI TOÁN TỐI ƯU TRÊN ĐỒ THỊ

8.1 Đồ thị có trọng số và bài toán đường đi ngắn nhất

8.1.1. Mở đầu

Trong đời sống, chúng ta thường gặp những tình huống như sau: để đi từ địa điểm A đến địa điểm B trong thành phố, có nhiều đường đi, nhiều cách đi; có lúc ta chọn đường đi ngắn nhất (theo nghĩa cự ly), có lúc lại cần chọn đường đi nhanh nhất (theo nghĩa thời gian) và có lúc phải cân nhắc để chọn đường đi rẻ tiền nhất (theo nghĩa chi phí), v.v...

Có thể coi sơ đồ của đường đi từ A đến B trong thành phố là một đồ thị, với đỉnh là các giao lộ (A và B coi như giao lộ), cạnh là đoạn đường nối hai giao lộ. Trên mỗi cạnh của đồ thị này, ta gán một số dương, ứng với chiều dài của đoạn đường, thời gian đi đoạn đường hoặc cước phí vận chuyển trên đoạn đường đó, ...

Định nghĩa 8.1. Đồ thị có trọng số là đồ thị $G = (V, E)$ mà mỗi cạnh (hoặc cung) $e \in E$ được gán bởi một số thực $c(e)$, gọi là *trọng số* của cạnh (hoặc cung) e.

Trong phần này, trọng số của mỗi cạnh được xét là một số dương và còn gọi là chiều dài của cạnh đó.

Định nghĩa 8.2. Mỗi đường đi từ đỉnh u đến đỉnh v, có chiều dài là $c(u, v)$, bằng tổng chiều dài các cạnh mà nó đi qua. *Khoảng cách* $d(u, v)$ giữa hai đỉnh u và v là chiều dài đường đi ngắn nhất (theo nghĩa $c(u, v)$ nhỏ nhất) trong các đường đi từ u đến v.

Có thể xem một đồ thị G bất kỳ là một đồ thị có trọng số mà mọi cạnh đều có chiều dài 1. Khi đó, khoảng cách $d(u, v)$ giữa hai đỉnh u và v là chiều dài của đường đi từ u đến v ngắn nhất, tức là đường đi qua ít cạnh nhất.

8.1.2. Bài toán tìm đường đi ngắn nhất

Bài toán: *Cho đồ thị $G = (V, E)$ và hai đỉnh a, b. Tìm đường đi ngắn nhất (nếu có) đi từ đỉnh a đến đỉnh b trong đồ thị G.*

Ý nghĩa thực tế: Bài toán này giúp chúng ta chọn các hành trình tiết kiệm nhất (quãng đường, thời gian, chi phí ...) trong giao thông, lập lịch thi công các công trình một cách tối ưu, xử lý trong truyền tin ...

Thuật toán duyệt đồ thị theo chiều rộng đã cho ta lời giải của bài toán này. Song ta có thêm thuật toán sau đây.

Thuật toán 8.1. Đường đi ngắn nhất duyệt theo chiều rộng

Đầu vào: Biểu diễn mảng DK các danh sách kề và a, b của $\mathcal{G}$.
Đầu ra: Danh sách các đỉnh đường đi ngắn nhất của đồ thị $\mathcal{G}$.

Bước 1. Lần lượt gán nhãn cho các đỉnh của đồ thị, mỗi đỉnh không quá một lần, như sau:
- Đỉnh a được gán nhãn là số 0.
- Những đỉnh kề với đỉnh a được gán số 1.
- Những đỉnh kề với đỉnh đã được gán nhãn số 1, được gán số 2.
....
- Tương tự, những đỉnh kề với đỉnh đã được gán số i được gán nhãn là số $i + 1$.
....
Thực hiện cho đến khi gán được nhãn cho đỉnh b hoặc không gán nhãn được nữa.

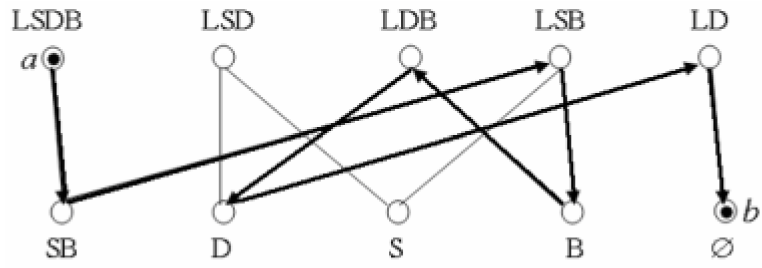
Bước 2. Nếu đỉnh b được gán nhãn nào đó là k thì kết luận có đường đi ngắn nhất từ đỉnh a tới đỉnh b với độ dài k, ngược lại thì trả lời là không có.

Bước 3. Khôi phục đường đi: Nếu ở bước 2. chỉ ra b được gán nhãn k nào đó thì ta đingược lại theo quy tắc sau đây: Nếu đỉnh y được gán nhãn j với $j \geq 1$ thì sẽ cóđỉnh x được gãn nhãn $j - 1$ sao cho có cạnh đi từ x tới y. Đi ngược lại cho đến khigặp đỉnh a, ta nhận được đường đi ngắn nhất cần tìm.

Ví dụ 8.1. (*Bài toán con sói, con dê và cái bắp cải*) Một con sói, một con dê và một cái bắp cải đang ở bờ sông. Người lái đò phải đưa chúng sang sông. Nhưng thuyền quá bé nên mỗi chuyến chỉ chở được một "hành khách" thôi. Vì những lý do mà ai cũng biết, không thể bỏ mặc sói với dê hoặc dê với bắp cải mà không có người trông. Vậy người lái đò phải xử trí thế nào mà vẫn đưa được sói, dê và bắp cải sang bên kia sông.

Lời giải. Xây dựng đồ thị vô hướng với các đỉnh thể hiện các hành khách còn lại bên phía xuất phát tại mỗi thời điểm khác nhau. Cạnh nối hai đỉnh thể hiện một chuyến đò qua sông.

Bài toán đưa về việc tìm đường đi ngắn nhất từ đỉnh a đến đỉnh b trên đồ thị. Đường đi như thế được chỉ ra bởi các mũi tên ở hình trên.



Hình 8.1. Hành trình qua sông của sói, dê và bắp cải

8.1.3. Bài toán Đường đi có trọng số bé nhất

Với bài toán đường đi tổng quát, ta xét các đồ thị có trọng số. Ta thường ký hiệu đồ thị có trọng số là (G, c). Độ dài của đường đi trong đồ thị có trọng số bằng tổng các trọng số của các cạnh trên đường đi đó.

Bài toán: *Cho đồ thị có trọng số (G, c) và hai đỉnh a, b thuộc G. Hãy tìm đường đi có trọng số bé nhất (nếu có) đi từ đỉnh a đến đỉnh b.*

Độ dài đường đi ngắn nhất từ đi đỉnh a đến đỉnh b còn được gọi là khoảng cách từ đỉnh a đến đỉnh b trong đồ thị. Nếu không có đường đi từ a đến b thì đặt khoảng cách bằng ∞.

Năm 1959 E. W. Dijkstra đưa ra một thuật toán rất hiệu quả để giải bài toán đường đi ngắn nhất.

Thuật toán thực hiện việc gán và giảm giá trị của nhãn $l(i)$ tại mỗi đỉnh i của đồ thị G như sau:

Thuật toán 8.2. Tìm đường đi ngắn nhất (E. W. Dijkstra)

Đầu vào: Biểu diễn mảng DK các danh sách kề và a, b của $\mathcal{G}$.
Đầu ra: Danh sách các đỉnh đường đi ngắn nhất của đồ thị $\mathcal{G}$.

Bước 1.Với đỉnh xuất phát a, gán nhãn $l(a) := 0$.

Bước 2. Nếu có cạnh (i, j) mà đỉnh i đã được gán nhãn và đỉnh j chưa được gánnhãn hoặc đỉnh j đã được gán nhãn nhưng $l(i) + c(i, j) < l(j)$ thì giảm nhãn $l(j) := l(i) + c(i, j)$.

Bước 3. Lặp lại bước 2. cho đến khi không gán hoặc giảm nhãn được nữa.

Định lý 8.1. *Tại mỗi đỉnh b giá trị nhãn $l(b)$ cuối cùng (nếu có) chính là độ dài của đường đi ngắn nhất từ đỉnh a đến đỉnh b.*

Chứng minh. Sau khi đã thực hiện xong thuật toán trên, nếu giá trị nhãn $l(b)$ xác định thì ta có đường đi từ đỉnh a tới đỉnh b.

Ta khôi phục đường đi từ a đến b như sau:

Xuất phát từ đỉnh b, tìm cạnh có đỉnh cuối là b và đỉnh đầu là i sao cho:

$$l(i) + c(i,b) = l(b).$$

Đỉnh i như thế chắc chắn phải tồn tại vì xảy ra đẳng thức ở lần gán hoặc giảm giá trị nhãn $l(j)$ cuối cùng. Cứ tiếp tục như thế cho đến khi gặp đỉnh a.

Giả sử ta nhận được dãy các cạnh:

$$(a,a_1),(a_1,a_2),...,(a_{k-1},b)$$

mà trên đó

$$\begin{aligned} l(a) + c(a,a_1) &= l(a_1) \\ l(a_1) + c(a_1,a_2) &= l(a_2) \\ &\cdots\cdots \\ l(a_{k-1}) + c(a_{k-1},b) &= l(b). \end{aligned}$$

Cộng từng vế và khử các giá trị chung ở cả hai vế ta có:

$$c(a,a_1) + c(a_1,a_2) + ... + c(a_{k-1},b) = l(b).$$

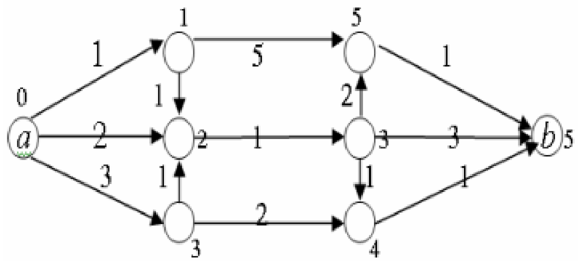
Vậy giá trị nhãn $l(b)$ chính là độ dài đường đi nói trên.

Bất kỳ đường đi nào khác từ đỉnh a đến đỉnh b cũng có các hệ thức tương tự nhưng có dấu $\geq$.

Vậy nhãn $l(b)$ là độ dài của đường đi ngắn nhất.

Ví dụ 8.2. Xét đồ thị có trọng số sau đây:

Độ dài đường đi ngắn nhất từ đỉnh a đến đỉnh b là 5.



Hình 8.2. Đồ thị có trọng số

Để đơn giản việc tính toán, ta xây dựng ma trận trọng số C :

$$C[i,j] = \begin{cases} c(i,j) & \text{nếu } (i,j) \in E \\ \infty & \text{nếu } (i,j) \notin E \\ 0 & \text{nếu } i = j. \end{cases}$$

Khi đó, thuật toán Dijkstra được trình bày chi tiết hơn như sau

Thuật toán 8.3. Dijkstra

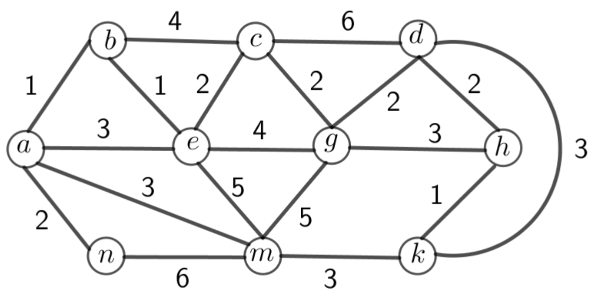
Đầu vào: Biểu diễn mảng C các trọng số và a của $\mathcal{G}$.
Đầu ra: Danh sách các đỉnh đường đi ngắn nhất của đồ thị $\mathcal{G}$.

```
procedure DIJKSTRA(a) ;
 begin
 for j ∈ V do
   begin
    L[j] := C[a, j]; Truoc[j] := a;
   end;
  T := V \ {a};
  while T ≠ ∅ do
    begin
      chọn đỉnh i ∈ T mà L[i] = min{L[j]|j ∈ T};
      T := T i;
      for j ∈ T do
           if L[j] > L[i] + C[i, j] then
           begin
                L[j] := L[i] + C[i, j];
                Truoc[j] := i;
             end;
```

```
    end ;
end ;
```

Biến mảng *Truoc* dùng để khôi phục đường đi.

Ví dụ 8.3. Tìm khoảng cách $d(a, v)$ từ a đến mọi đỉnh v và tìm đường đi ngắn nhất từ a đến v cho trong đồ thị G sau.



Hình 8.3. Thuật toán Dijkstra

Lời giải.

$L(a)$	$L(b)$	$L(c)$	$L(d)$	$L(e)$	$L(g)$	$L(h)$	$L(k)$	$L(m)$	$L(n)$
0	∞	∞	∞	∞	∞	∞	∞	∞	∞
–	1	∞	∞	3	∞	∞	∞	3	2
–	–	5	∞	2	∞	∞	∞	3	2
–	–	5	∞	2	∞	∞	∞	3	–
–	–	4	∞	–	6	∞	∞	3	–
–	–	4	∞	–	6	∞	6	–	–
–	–	–	10	–	6	∞	6	–	–
–	–	–	8	–	–	9	6	–	–
–	–	–	8	–	–	7	–	–	–
–	–	–	8	–	–	–	–	–	–

Hình 8.4. Lời giải theo thuật toán Dijkstra

Định lý 8.2. *Thuật toán Dijkstra tìm được đường đi ngắn nhất từ một đỉnh cho trước đến một đỉnh tuỳ ý trong đơn đồ thị vô hướng liên thông có trọng số.*

Chứng minh. Định lý được chứng minh bằng quy nạp. Tại bước k ta có giả thiết quy nạp là

(i) Nhãn của đỉnh v không thuộc S là độ dài của đường đi ngắn nhất từ đỉnh a tới đỉnh này;

(ii) Nhãn của đỉnh v trong S là độ dài của đường đi ngắn nhất từ đỉnh a tới đỉnh này và đường đi này chỉ chứa các đỉnh (ngoài chính đỉnh này) không thuộc S.

Khi $k = 0$, tức là khi chưa có bước lặp nào được thực hiện, $S = V \setminus \{a\}$, vì thế độ dài của đường đi ngắn nhất từ a tới các đỉnh khác a là ∞ và độ dài của đường đi ngắn nhất từ a tới chính nó bằng 0 (ở đây, chúng ta cho phép đường đi không có cạnh). Do đó bước cơ sở là đúng.

Giả sử giả thiết quy nạp là đúng với bước k. Gọi v là đỉnh lấy ra khỏi S ở bước lặp $k+1$, vì vậy v là đỉnh thuộc S ở cuối bước k có nhãn nhỏ nhất (nếu có nhiều đỉnh có nhãn nhỏ nhất thì có thể chọn một đỉnh nào đó làm v). Từ giả thiết quy nạp ta thấy rằng trước khi vào vòng lặp thứ $k+1$, các đỉnh không thuộc S đã được gán nhãn bằng độ dài của đường đi ngắn nhất từ a. Đỉnh v cũng vậy phải được gán nhãn bằng độ dài của đường đi ngắn nhất từ a. Nếu điều này không xảy ra thì ở cuối bước lặp thứ k sẽ có đường đi với độ dài nhỏ hơn $L_k(v)$ chứa cả đỉnh thuộc S (vì $L_k(v)$ là độ dài của đường đi ngắn nhất từ a tới v chứa chỉ các đỉnh không thuộc S sau bước lặp thứ k). Gọi u là đỉnh đầu tiên của đường đi này thuộc S. Đó là đường đi với độ dài nhỏ hơn $L_k(v)$ từ a tới u chứa chỉ các đỉnh không thuộc S. Điều này trái với cách chọn v. Do đó (i) vẫn còn đúng ở cuối bước lặp $k+1$.

Gọi u là đỉnh thuộc S sau bước $k+1$. Đường đi ngắn nhất từ a tới u chứa chỉ các đỉnh không thuộc S sẽ hoặc là chứa v hoặc là không. Nếu nó không chứa v thì theo giả thiết quy nạp độ dài của nó là L k (v). Nếu nó chứa v thì nó sẽ tạo thành đường đi từ a tới v với độ dài có thể ngắn nhất và chứa chỉ các đỉnh không thuộc S khác v, kết thúc bằng cạnh từ v tới u. Khi đó độ dài của nó sẽ là $L_k(v) + m(v, u)$. Điều đó chứng tỏ (ii) là đúng vì

$L_{k+1}(u) = min(L_k(u), L_k(v) + m(v, u))$.

Mệnh đề 8.1. *Thuật toán Dijkstra tìm đường đi ngắn nhất từ một đỉnh cho trước đến một đỉnh tuỳ ý trong đơn đồ thị vô hướng liên thông có trọng số có độ phức tạp là $O(n^2)$.*

Chứng minh. Thuật toán dùng không quá $n-1$ bước lặp. Trong mỗi bước lặp, dùng không hơn $2(n-1)$ phép cộng và phép so sánh để sửa đổi nhãn của các đỉnh. Ngoài ra, một đỉnh thuộc S_k có nhãn nhỏ nhất nhờ không quá $n-1$ phép so sánh. Do đó thuật toán có độ phức tạp $O(n^2)$.

8.1.4. Thuật toán Floyd

Cho $G = (V, E)$ là một đồ thị có hướng, có trọng số. Để tìm đường đi ngắn nhất giữa mọi cặp đỉnh của G, ta có thể áp dụng thuật toán Dijkstra nhiều lần hoặc áp dụng thuật toán Floyd được trình bày dưới đây.

Giả sử $V = \{v_1, v_2, ..., v_n\}$ và có ma trận trọng số là $W = W_0$. Thuật toán Floyd xây dựng dãy các ma trận vuông cấp n là $W_k (0 \leq k \leq n)$ như sau:

Thuật toán 8.4. Thuật toán Floyd

```
procedure Xac_dinh W_n
  for i := 1 to n
    for j := 1 to n
      W[i,j] := m(v_i, v_j) //W[i,j] là phần tử dòng i cột j của ma
trận W_0
      for k := 1 to n
        if W[i,k] + W[k,j] < W[i,j] then
                                  W[i,j] := W[i,k] + W[k,j]
//W[i,j] là phần tử dòng i cột j của ma trận W_k
```

Định lý 8.3. *Thuật toán Floyd cho ta ma trận $W^* = W_n$ là ma trận khoảng cách nhỏ nhất của đồ thị G.*

Chứng minh. Ta chứng minh bằng quy nạp theo k mệnh đề sau:

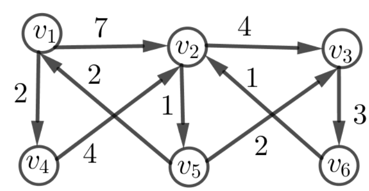
$W_k[i,j]$ là chiều dài đường đi ngắn nhất trong những đường đi nối đỉnh v_i với đỉnh v_j đi qua các đỉnh trung gian trong $\{v_1, v_2, ..., v_k\}$.

Trước hết mệnh đề hiển nhiên đúng với $k = 0$.

Giả sử mệnh đề đúng với $k - 1$.

Xét $W_k[i,j]$. Có hai trường hợp:

1. Trong các đường đi chiều dài ngắn nhất nối v_i với v_j và đi qua các đỉnh trung gian trong $\{v_1, v_2, ..., v_k\}$, có một đường đi γ sao cho $v_k \in \gamma$. Khi đó γ cũng là đường đi ngắn nhất nối v_i với v_j đi qua các đỉnh trung gian trong $\{v_1, v_2, ..., v_{k-1}\}$, nên theo giả thiết quy nạp,
$$W_{k-1}[i,j] = \text{ chiều dài } \gamma \le W_{k-1}[i,k] + W_{k-1}[k,j].$$
Do đó theo định nghĩa của W k thì $W_k[i,j] = W_{k-1}[i,j]$.
2. Mọi đường đi chiều dài ngắn nhất nối v_i với v_j và đi qua các đỉnh trung gian trong $\{v_1, v_2, ..., v_k\}$, đều chứa v_k . Gọi $\gamma = v_i ... v_k ... v_j$ là một đường đi ngắn nhất như thế thì $v_1 ... v_k$ và $v_k ... v_j$ cũng là những đường đi ngắn nhất đi qua các đỉnh trung gian trong $\{v_1, v_2, ..., v_{k-1}\}$ và
$$\begin{aligned} W_{k-1}[i,k] + W_{k-1}[k,j] &= \text{ chiều dài } (v_1 ... v_k) + \text{ chiều dài } (v_k ... v_j) \\ &= \text{ chiều dài } \gamma < W_{k-1}[i,j]. \end{aligned}$$
Do đó theo định nghĩa của W_k thì ta có
$$W_k[i,j] = W_{k-1}[i,k] + W_{k-1}[k,j].$$



Hình 8.5. Lời giải theo thuật toán Floyd

Ví dụ 8.4.

Lời giải. Áp dụng thuật toán Floyd, ta tìm được (các ô trống là

∞).

$$W = W_0 = \begin{pmatrix} \infty & 7 & \infty & 2 & \infty & \infty \\ \infty & \infty & 4 & \infty & 1 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & \infty & \infty & \infty & \infty \\ 2 & \infty & 2 & \infty & \infty & \infty \\ \infty & 1 & \infty & \infty & \infty & \infty \end{pmatrix}$$

$$W_1 = \begin{pmatrix} \infty & 7 & \infty & 2 & \infty & \infty \\ \infty & \infty & 4 & \infty & 1 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & \infty & \infty & \infty & \infty \\ 2 & 9 & 2 & 4 & \infty & \infty \\ \infty & 1 & \infty & \infty & \infty & \infty \end{pmatrix}, W_2 = \begin{pmatrix} \infty & 7 & 11 & 2 & 8 & \infty \\ \infty & \infty & 4 & \infty & 1 & \infty \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & 8 & \infty & 5 & \infty \\ 2 & 9 & 2 & 4 & 10 & \infty \\ \infty & 1 & 5 & \infty & 2 & \infty \end{pmatrix}$$

$$W_3 = \begin{pmatrix} \infty & 7 & 11 & 2 & 8 & 14 \\ \infty & \infty & 4 & \infty & 1 & 7 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & 8 & \infty & 5 & 11 \\ 2 & 9 & 2 & 4 & 10 & 5 \\ \infty & 1 & 5 & \infty & 2 & 8 \end{pmatrix}, W_4 = \begin{pmatrix} \infty & 6 & 10 & 2 & 7 & 13 \\ \infty & \infty & 4 & \infty & 1 & 7 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ \infty & 4 & 8 & \infty & 5 & 11 \\ 2 & 8 & 2 & 4 & 9 & 5 \\ \infty & 1 & 5 & \infty & 2 & 8 \end{pmatrix}$$

$$W_5 = \begin{pmatrix} 9 & 6 & 9 & 2 & 7 & 12 \\ 3 & 9 & 3 & 5 & 1 & 6 \\ \infty & \infty & \infty & \infty & \infty & 3 \\ 7 & 4 & 7 & 9 & 5 & 10 \\ 2 & 8 & 2 & 4 & 9 & 5 \\ 4 & 1 & 4 & 6 & 2 & 7 \end{pmatrix}, W_6 = \begin{pmatrix} 9 & 6 & 9 & 2 & 7 & 12 \\ 3 & 7 & 3 & 5 & 1 & 6 \\ 7 & 4 & 7 & 9 & 5 & 3 \\ 7 & 4 & 7 & 9 & 5 & 10 \\ 2 & 6 & 2 & 4 & 7 & 5 \\ 4 & 1 & 4 & 6 & 2 & 7 \end{pmatrix}$$

Thuật toán Floyd có thể áp dụng cho đồ thị vô hướng cũng như đồ thị có hướng. Ta chỉ cần thay mỗi cạnh vô hướng (u, v) bằng một cặp cạnh có hướng (u, v) và (v, u) với $m(u, v) = m(v, u)$. Tuy nhiên, trong trường hợp này, các phần tử trên đường chéo của ma trận W cần đặt bằng 0.

Đồ thị có hướng G là liên thông mạnh khi và chỉ khi mọi phần tử nằm trên đường chéo trong ma trận trọng số ngắn nhất W^* đều hữu hạn.

8.2 Bài toán luồng cực đại

8.2.1. Luồng vận tải

Định nghĩa 8.3. *Mạng vận tải* là một đồ thị có hướng, không có khuyên và có trọng số $G = (V, E)$ với $V = \{v_0, v_1, ..., v_n\}$ thoả mãn

1. Mỗi cung $e \in E$ có trọng số $m(e)$ là một số nguyên không âm và được gọi là khả năng thông qua của cung e.
2. Có một và chỉ một đỉnh v_0 không có cung đi vào, tức là $\deg_t(v_0) = 0$. Đỉnh v_0 được gọi là lối vào hay đỉnh phát của mạng.
3. Có một và chỉ một đỉnh v_n không có cung đi ra, tức là $\deg_o(v_n) = 0$. Đỉnh v_n được gọi là lối ra hay đỉnh thu của mạng.

Định nghĩa 8.4. Để định lượng khai thác, tức là xác định lượng vật chất chuyển qua mạng vận tải $G = (V, E)$, người ta đưa ra khái niệm luồng vận tải và nó được định nghĩa như sau.

Hàm φ xác định trên tập cung E và nhận giá trị nguyên được gọi là luồng vận tải của mạng vận tải G nếu φ thoả mãn

1) $\varphi(e) \geq 0, \forall e \in E$.

2) $\sum\limits_{e\in\Gamma^-(v)} \varphi(e) = \sum\limits_{e\in\Gamma^+(v)} \varphi(e), v \neq v_0, v \neq v_n,$

$\Gamma^-(v) = \{e \in E | e \text{ có đỉnh cuối là } v\}$

$\Gamma^+(v) = \{e \in E | e \text{ có đỉnh đầu là } v\}$

3) $\varphi(e) \leq c(e), \forall e \in E$.

Ta xem $\varphi(e)$ như là lượng hàng chuyển trên cung $e = (u, v)$ từ đỉnh u đến đỉnh v và không vượt quá khả năng thông qua của cung này. Ngoài ra, từ điều kiện 2) ta thấy rằng nếu v không phải là lối vào v_0 hay lối ra v_n, thì lượng hàng chuyển tới v bằng lượng hàng chuyển khỏi v.

Từ quan hệ 2) suy ra

4) $\sum_{e\in\Gamma^-(v_0)} \varphi(e) = \sum_{e\in\Gamma^+(v_0)} \varphi(e) =: \varphi_{v_n}$.

Đại lượng φ_{v_n} (ta còn ký hiệu là φ_n) được gọi là luồng qua mạng, hay cường độ luồng tại điểm v_n hay giá trị của luồng φ. Bài toán đặt ra ở đây là tìm φ để φ_{v_n} đạt giá trị lớn nhất, tức là tìm giá trị lớn nhất của luồng.

Định nghĩa 8.5. Cho mạng vận tải $G = (V, E)$ và $A \subset V$. Ký hiệu

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

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

Đối với tập cung M tuỳ ý, đại lượng $\varphi(M) = \sum_{e\in M} \varphi(e)$ được gọi là *luồng của tập cung* M.

Từ điều kiện 2) dễ dàng suy ra hệ quả sau.

Hệ quả 8.1. *Cho φ là luồng của mạng vận tải $G = (V, E)$ và $A \subset V \setminus \{v_0, v_n\}$. Khi đó*

$$\varphi(\Gamma^-(A)) = \varphi(\Gamma^+(A)).$$

8.2.2. Bài toán luồng cực đại

Cho mạng vận tải $G = (V, E)$. Hãy tìm luồng φ để đạt φ_{v_n} max trên mạng G.

Nguyên lý của các thuật toán giải bài toán tìm luồng cực đại là như sau.

Định nghĩa 8.6. Cho $A \subset V$ là tập con tuỳ ý không chứa lối vào v_0 và chứa lối ra v_n. Tập $\Gamma^-(A)$ được gọi là một thiết diện của mạng vận tải G. Đại lượng $m(\Gamma^-(A)) = \sum_{e\in\Gamma^-(A)} c(e)$ được gọi là khả năng thông qua của thiết diện $\Gamma^-(A)$.

Từ định nghĩa thiết diện và khả năng thông qua của nó ta nhận thấy rằng: mỗi đơn vị hàng hoá được chuyển từ v_0 đến v_n ít nhất cũng phải một lần qua một cung nào đó của thiết diện

$\Gamma^-(A)$. Vì vậy, dù luồng φ và thiết diện $\Gamma^-(A)$ như thế nào đi nữa cũng vẫn thoả mãn quan hệ

$$\varphi_n \le m(\Gamma^-(A)).$$

Do đó, nếu đối với luồng φ và thiết diện W mà có:

$$\varphi_n = m(W)$$

thì chắc chắn rằng luồng φ đạt giá trị lớn nhất và thiết diện W có khả năng thông qua nhỏ nhất.

Định nghĩa 8.7. Cung e trong mạng vận tải G với luồng vận tải φ được goi là *cung bão hoà* nếu $\varphi(e) = m(e)$.

Luồng φ của mạng vận tải G được gọi là luồng đầy nếu mỗi đường đi từ v_0 đến v_n đều chứa ít nhất một cung bão hoà.

Từ định nghĩa trên ta thấy rằng, nếu luồng φ trong mạng vận tải G chưa đầy thì nhất định tìm được đường đi α từ lối vào v_0 đến lối ra v_n không chứa cung bão hoà. Khi đó ta nâng luồng φ thành φ' như sau

$$\varphi'(e) = \begin{cases} \varphi(e) + 1 & \text{khi } e \in \alpha \\ \varphi(e) & \text{khi } e \notin \alpha \end{cases}$$

Khi đó φ' cũng là một luồng, mà giá trị của nó là

$$\varphi'(e) = \varphi_n 1 > \varphi_n.$$

Như vậy, đối với mỗi luồng không đầy ta có thể nâng giá trị của nó và nâng cho tới khi nhận được một luồng đầy.

Tuy vậy, thực tế cho thấy rằng có thể có một luồng đầy, nhưng vẫn chưa đạt tới giá trị cực đại. Bởi vậy, cần phải dùng thuật toán Ford-Fulkerson để tìm giá trị cực đại của luồng.

Thuật toán Ford-Fulkerson

Để tìm luồng cực đại của mạng vận tải G, ta xuất phát từ luồng tuỳ ý φ của G, rồi nâng luồng lên đầy, sau đó áp dụng thuật toán Ford-Fulkerson hoặc ta có thể áp dụng thuật toán Ford-Fulkerson trực tiếp đối với luồng φ.

Thuật toán gồm 3 bước

Bước 1 (đánh dấu ở đỉnh của mạng). Lối vào v_0 được đánh dấu bằng 0.

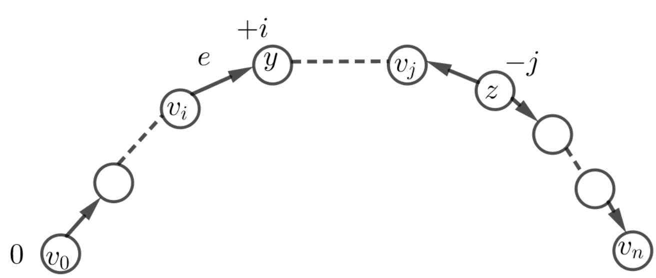
1. Nếu đỉnh v_i đã được đánh dấu thì ta dùng chỉ số $+i$ để đánh dấu cho mọi đỉnh y chưa được đánh dấu mà $(v_i, y) \in E$ và cung này chưa bão hoà $(\varphi(v_i, y) < m(v_i, y))$.
2. Nếu đỉnh v_i đã được đánh dấu thì ta dùng chỉ số $-i$ để đánh dấu cho mọi đỉnh z chưa được đánh dấu mà $(z, v_i) \in E$ và luồng của cung này dương $(\varphi(z, v_i) > 0)$.

Nếu với phương pháp này ta đánh dấu được tới lối ra v_n thì trong G tồn tại giữa v_0 và v_n một xích α, mọi đỉnh đều khác nhau và được đánh dấu theo chỉ số của đỉnh liền trước nó (chỉ sai khác nhau về dấu). Khi đó chắc chắn ta nâng được giá trị của luồng.

Bước 2 (nâng giá trị của luồng). Để nâng giá trị của luồng φ, ta đặt

$$\varphi'(e) = \begin{cases} \varphi(e) & \text{nếu } e \notin \alpha, \\ \varphi(e) + 1 & \text{nếu } e \in \alpha \text{ được định hướng theo} \\ & \qquad \text{chiều của xích } \alpha \text{ đi từ } v_0 \text{ đến } v_n, \\ \varphi(e) - 1 & \text{nếu } e \in \alpha \text{ được định hướng ngược} \\ & \qquad \text{với chiều của xích } \alpha \text{ đi từ } v_0 \text{ đến } v_n. \end{cases}$$

φ' thoả mãn các điều kiện về luồng, nên φ' là một luồng và



Hình 8.6. Nâng giá trị luồng

ta có

$$\varphi'_n = \varphi_n + 1.$$

Như vậy, ta đã nâng được luồng lên một đơn vị. Sau đó lặp lại một vòng mới. Vì khả năng thông qua của các cung đều hữu hạn, nên quá trình phải dừng lại sau một số hữu hạn bước.

Bước 3. Nếu với luồng φ^0 bằng phương pháp trên ta không thể nâng giá trị của luồng lên nữa, nghĩa là ta không thể đánh dấu được đỉnh v_n, thì ta nói rằng quá trình nâng luồng kết thúc và φ^0 đã đạt giá trị cực đại, đồng thời gọi φ^0 là luồng kết thúc.

Khi mạng vận tải $G = (V, E)$ đạt tới luồng φ^0, thì bước tiếp theo ta không thể đánh dấu được tới lối ra v_n. Trên cơ sở hiện trạng được đánh dấu tại bước này, ta sẽ chứng minh rằng luồng φ^0 đã đạt được giá trị cực đại.

Bổ đề 8.1. *Cho luồng φ của mạng vận tải $G = (V, E)$ và $A \subset V$, chứa lối ra v_n và không chứa lối vào v_0. Khi đó*

$$\varphi_{v_n} = \varphi(\Gamma^-(A)) - \varphi(\Gamma^+(A)).$$

Chứng minh. Đặt $A_1 = A \setminus \{v_n\}$. Theo Hệ quả 8.1, ta có

$$\varphi(\Gamma^-(A_1)) = \varphi(\Gamma^+(A_1)). \tag{8.1}$$

Đặt $C_1 = \{(a, v_n) \in E | a \notin A\}$. Khi đó $\Gamma^-(A) = \Gamma^-(A_1) \cup C_1$ và $\Gamma^-(A_1) \cap C_1 = \emptyset$, nên

$$\varphi(\Gamma^-(A)) = \varphi(\Gamma^-(A_1)) + \varphi(C_1). \tag{8.2}$$

Đặt $C_2 = \{(b, v_n) \in E | b \in A_1\}$. Khi đó $\Gamma^+(A_1) = \Gamma^+(A) \cup C_2$ và $\Gamma^+(A_1) \cap C_2 = \emptyset$, nên

$$\varphi(\Gamma^+(A)) = \varphi(\Gamma^+(A_1)) + \varphi(C_2). \tag{8.3}$$

Ngoài ra, $\Gamma^-(v_n) = C_1 \cup C_2$ và $C_1 \cap C_2 = \emptyset$, nên

$$\varphi_{v_n} = \varphi(\Gamma^-(v_n)) = \varphi(C_1) + \varphi(C_2). \tag{8.4}$$

Từ (8.1), (8.2), (8.3) và (8.4), ta có

$$\varphi_{v_n} = \varphi(\Gamma^-(A)) - \varphi(\Gamma^+(A)).$$

Định lý 8.4 (Ford-Fulkerson). *Trong mạng vận tải $G = (V, E)$, giá trị lớn nhất của luồng bằng khả năng thông qua nhỏ nhất của thiết diện, nghĩa là*

$$\max_{\varphi} \varphi_{v_n} = \min_{A \subset V, v_0 \notin A, v_n \in A} m(\Gamma^-(A)).$$

Chứng minh. Giả sử trong mạng vận tải G, φ^0 là luồng cuối cùng, mà sau đó bằng phương pháp đánh dấu của thuật toán Ford-Fulkerson không đạt tới lối ra v_n. Trên cơ sở hiện trạng được đánh dấu lần cuối cùng này, ta dùng B để ký hiệu tập gồm các đỉnh của G không được đánh dấu. Khi đó $v_0 \in B$, $v_n \in B$. Do đó $\Gamma^-(B)$ là một thiết diện của mạng vận tải G và theo Bổ đề 8.1, ta có

$$\varphi^0_{v_n} = \varphi^0(\Gamma^-(B)) - \varphi^0(\Gamma^+(B)). \tag{8.5}$$

Đối với mỗi cung $e = (u, v) \in \Gamma^-(B)$ thì $u \notin B$ và $v \in B$, tức là u được đánh dấu và v không được đánh dấu, nên theo nguyên tắc đánh dấu thứ nhất, e đã là cung bão hoà

$$\varphi^0(e) = m(e).$$

Do đó,

$$\varphi^0(\Gamma^-(B)) = \sum_{e \in \Gamma^-(B)} \varphi^0(e) = \sum_{e \in \Gamma^-(B)} m(e) = \varphi^0(\Gamma^+(B)). \tag{8.6}$$

Đối với mỗi cung $e = (s, t) \in \Gamma^+(B)$ thì $s \in B$ và $t \notin B$, tức là s được đánh dấu và t được đánh dấu, nên theo nguyên tắc đánh dấu thứ hai

$$\varphi^0(e) = 0.$$

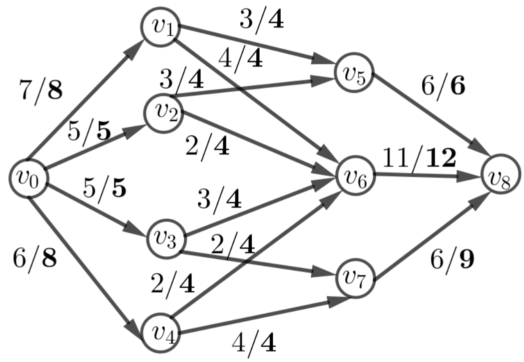
Do đó,

$$\varphi^0(\Gamma^+(B)) = \sum_{e \in \Gamma^+(B)} \varphi^0(e) = 0. \tag{8.7}$$

Từ (8.5), (8.6) và (8.7) ta suy ra

$$\varphi^0_{v_n} = m(\Gamma^-(B)).$$

Vì vậy, $\varphi^0_{v_n}$ là giá trị lớn nhất của luồng đạt được, còn $m(\Gamma^-(B))$ là giá trị nhỏ nhất trong các khả năng thông qua của các thiết diện thuộc mạng vận tải G.



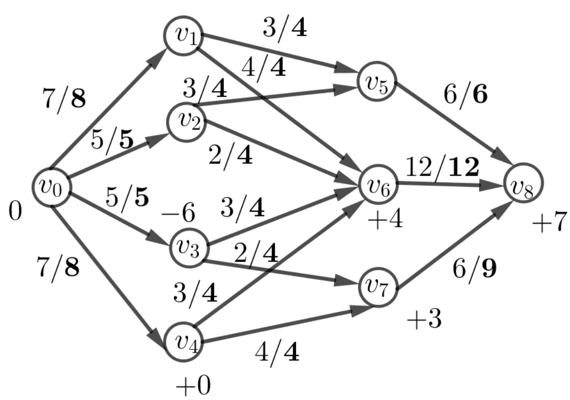
Hình 8.7. Luồng φ

Ví dụ 8.5. Cho mạng vận tải như Hình 8.7 với khả năng thông qua được đặt dưới /, luồng được ghi trên dấu /. Tìm luồng cực đại của mạng này.

Lời giải. Luồng φ có đường đi $(v_0, v_4), (v_4, v_6), (v_6, v_8)$ gồm các cung chưa bão hoà nên nó chưa đầy. Do đó có thể nâng luồng của các cung này lên một đơn vị, để được φ^1 (Hình 8.8).

Do mỗi đường xuất phát từ v_0 đến v_8 đều chứa ít nhất một cung bão hoà, nên luồng φ^1 là luồng đầy. Song nó chưa phải là luồng cực đại.

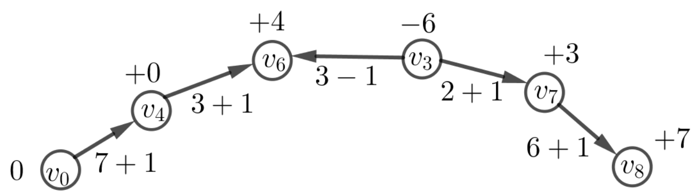
Áp dụng thuật toán Ford-Fulkerson để nâng luồng φ^1.



Hình 8.8. Luồng φ^1

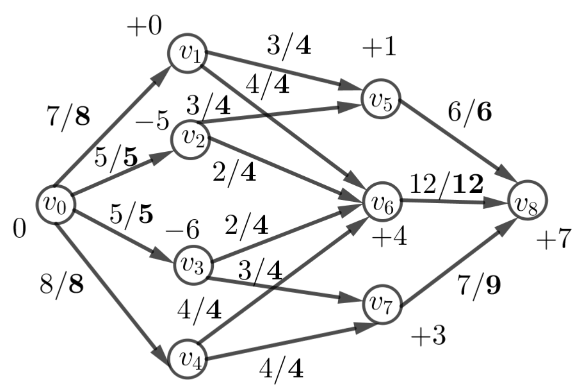
Xét xích $\alpha = (v_0, v_4, v_6, v_3, v_7, v_8)$. Quá trình đánh dấu từ v_0

đến v_8 để có thể nâng luồng φ^1 lên một đơn vị bằng cách biến đổi luồng tại các cung thuộc xích α được đánh dấu. Sau đó ta có luồng φ^2 (Hình 8.10).



Hình 8.9. Xích α

Xét xích $\beta = (v_0, v_1, v_5, v_2, v_6, v_3, v_7, v_8)$. Quá trình đánh dấu từ v_0 đến v_8 để có thể nâng luồng φ^2 lên một đơn vị bằng cách biến đổi luồng tại các cung thuộc xích β được đánh dấu. Sau đó ta có luồng φ^3 (Hình 8.12).



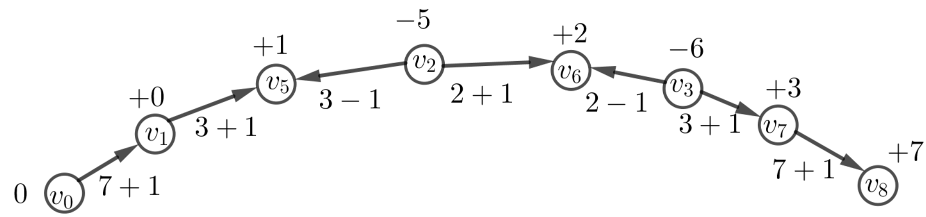
Hình 8.10. Luồng φ^2

Tiếp theo ta chỉ có thể đánh dấu được đỉnh v_0 nên quá trình nâng luồng kết thúc và ta được giá trị của luồng cực đại là

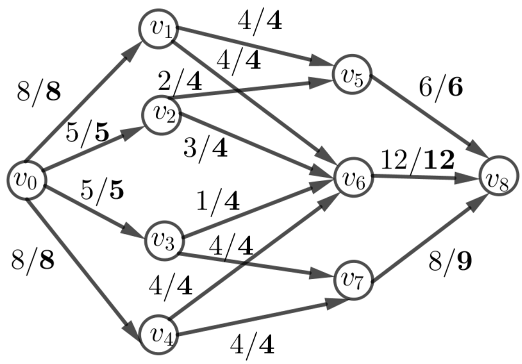
$$\varphi^3_{v_8} = 6 + 12 + 8 = 26.$$

Mặt khác, thiết diện nhỏ nhất $\Gamma^-(B)$ với $B = \{v_1, v_2, ..., v_8\}$ là

$$\Gamma^-(B) = \{(v_0, v_1), (v_0, v_2), (v_0, v_3), (v_0, v_4)\}.$$



Hình 8.11. Xích β



Hình 8.12. Luồng φ^3

8.3 Một số ứng dụng luồng lớn nhất

Bài toán luồng lớn nhất có rất nhiều ứng dụng trong việc giải quyết các bài toán khác nhau của lý thuyết đồ thị.

8.3.1. Bài toán luồng nhỏ nhất

Ngược lại với bài toán luồng lớn nhất, chúng ta xét bài toán sau đây: *Biton* : Cho mạng (G, c). Tìm luồng t qua mạng có giá trị tz nhỏ nhất và thoả mãn điều kiện a′) thay cho điều kiện a) như sau

a′) $\forall e \in E, t(e) \geq c(e)$.

Thuật toán 8.5. Tìm luồng bé nhất

+ Ta dùng phương pháp cải tiến luồng giống như phương pháp giải bài toánluồng lớn nhất.

+ Xuất phát từ một luồng t nào đó thoả mãn điều kiện c), ta dùng phươngpháp sau đây để giảm giá trị của luồng t.

Bước 1: *Đánh dấu các đỉnh của mạng.*

- Đầu tiên đánh dấu cho đỉnh thu z số 0.
- Nếu đỉnh y đã được đánh dấu, có cạnh (x, y) với đỉnh đầu chưa được đánh dấu và $t((x, y)) > c((x, y))$ thì đánh dấu cho đỉnh x là $+y$.
- Nếu đỉnh x đã được đánh dấu, có cạnh (x, y) thì đánh dấu cho đỉnh y là $-x$.

Với cách đánh dấu này mà đi tới được đỉnh phát x_0 thì ta đã tìm được mộtđường đi vô hướng từ z tới x_0 được đánh dấu.

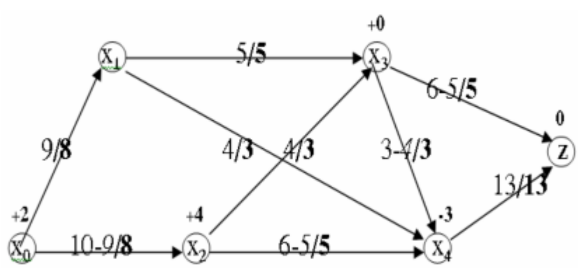
Bước 2: *Giảm luồng.*

- Bây giờ ta có thể giảm luồng đi 1 bằng cách chọn luồng mới t′ như sau
- Nếu cạnh e không thuộc đường đi trên thì giữ nguyên luồng, nghĩa là

$$t'(e) := t(e)$$

- Nếu cạnh e thuộc đường đi này và cùng chiều với chiều từ x_0 tới z thì đặt$t'(e) := t(e) - 1$ (vì trên cạnh đó $t(e) > c(e)$) còn nếu cạnh e ngược chiều thì đặt$t'(e) := t(e) + 1$.
- Lặp lại quá trình giảm luồng trên cho đến khi không đánh dấu được tới đỉnhphát x_0 . Khi đó luồng nhận được có giá trị nhỏ nhất.

Ví dụ 8.6. Xét mạng vận tải sau đây.



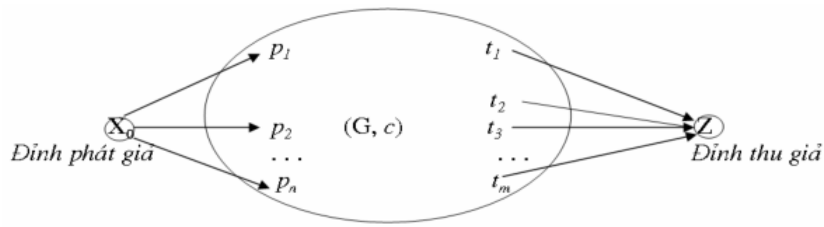
Hình 8.13. Mạng vận tải và luồng đã giảm

Luồng cũ có giá trị là $tz = 19$. Luồng mới sau khi cải tiến có giá trị là $tz' = 18$ và là luồng nhỏ nhất.

8.3.2. Bài toán luồng trên mạng có nhiều đỉnh phát và đỉnh thu

Giả sử (G, c) là một mạng vận tải với n đỉnh phát: $p_1, p_2, .., p_n$ và m đỉnh thu: $q_1, q_2, .., q_m$.

Bài toán tìm luồng lớn nhất từ nhiều đỉnh phát tới nhiều đỉnh thu có thể đưa về bài toán luồng lớn nhất từ một đỉnh phát tới một đỉnh thu bằng cách thêm vào một *đỉnh phát giả* X_0 , một đỉnh thu giả Z, các cạnh nối X_0 với tất cả các đỉnh phát và các cạnh nối tất cả các đỉnh thu với Z.



Hình 8.14. Mạng vận tải có nhiều đỉnh phát và nhiều đỉnh thu

Khả năng thông qua của các cạnh mới như sau:

- Nếu lượng phát của đỉnh p_i bị hạn chế bởi l_i thì đặt $c(X_0, p_i) = l_i$, còn nếu không bị hạn chế thì đặt bằng ∞.

- Tương tự như thế, giới hạn của lượng thu của đỉnh t_j sẽ là khả năng thông qua của cạnh (t_j, Z).

8.3.3. Bài toán tìm cặp ghép lớn nhất của đồ thị hai phần

Bài toán này là một dạng đặc biệt của bài toán mạng với nhiều đỉnh phát và nhiều đỉnh thu. Ta đưa bài toán này về bài toán luồng lớn nhất qua mạng.

Giả sử đồ thị $G = (V_1, V_2, F)$ là đồ thị hai phần. Ta xây dựng mạng vận tải như sau:

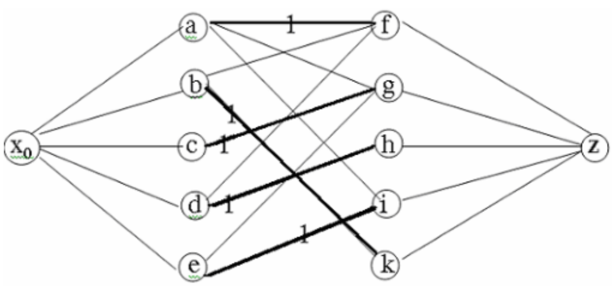
Các đỉnh của mạng là các đỉnh của đồ thị G và thêm vào đỉnh phát x_0 và đỉnh thu z. Mạng sẽ gồm tất cả các cạnh của G có hướng từ V_1 sang V_2 . Ngoài ra còn nối x_0 với tất cả các đỉnh trong V_1 và nối tất cả các đỉnh trong V_2 với z. Trên mọi cạnh e của mạng đều đặt $c(e) = 1$.

Khi đó mỗi luồng t qua mạng sẽ ứng với một cặp ghép W của G mà:

$$e \in W \Leftrightarrow t(e) = 1.$$

Ngược lại, mỗi cặp ghép W sẽ ứng với một luồng t qua mạng của G cũng theo quy tắc trên. Vậy tz đạt lớn mhất khi W có nhiều cạnh nhất.

Ví dụ 8.7. Từ một đồ thị hai phần gồm tập đỉnh a. b, c, d, e, f, g, h, i, k ta xây dựng mạng vận tải như sau



Hình 8.15. Mạng vận tải trên đồ thị hai phần

8.3.4. Bài toán vận tải thông qua các cạnh và các đỉnh

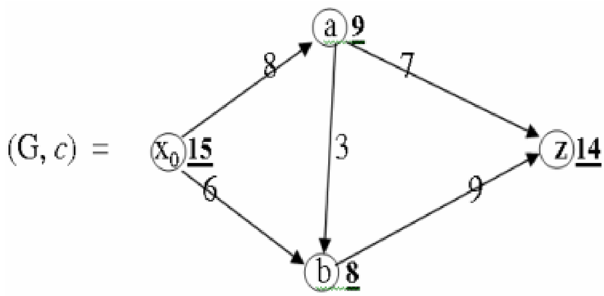
Giả sử trong đồ thị G, ngoài khả năng thông qua của các cạnh thì với mỗi đỉnh $x \in V$ còn có khả năng thông qua của đỉnh là $d(x)$ và đòi hỏi tổng luồng đi vào đỉnh x không được vượt quá $d(x)$, nghĩa là:

$$t(W^-(x)) \leq d(x).$$

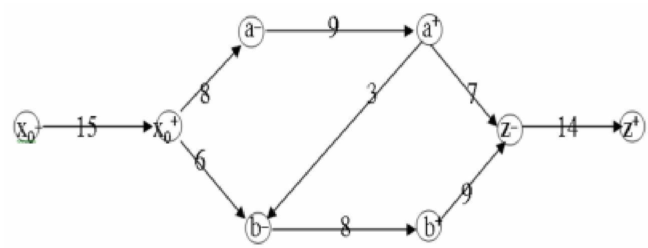
Hãy tìm luồng lớn nhất giữa x_0 và z trong mạng này.

Để đưa bài toán này về bài toán luồng lớn nhất, chúng ta xây dựng mạng G' sao cho: Mỗi đỉnh x trong G tương ứng với hai đỉnh x^- và x^+ trong G', cạnh (x^-, x^+) thuộc G' và $c((x^-, x^+)) = d(x)$. Mỗi cạnh (x, y) trong G ứng với cạnh (x^+, y^-) trong G'.

Ví dụ 8.8. Xét mạng vận tải sau đây



Hình 8.16. Mạng vận tải với khả năng thông qua cạnh và đỉnh



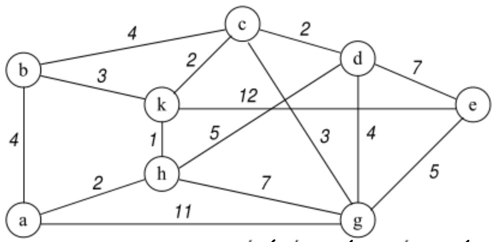
Hình 8.17. Mạng vận tải tương ứng

Xây dựng mạng (G', c) như sau

Do luồng đi vào đỉnh x^- phải đi qua cạnh (x^-, x^+) với khả năng thông qua $d(x)$ nên luồng lớn nhất trong G' sẽ bằng luồng lớn nhất trong G và thoả mãn các điều kiện về khả năng thông qua của các cạnh và các đỉnh.

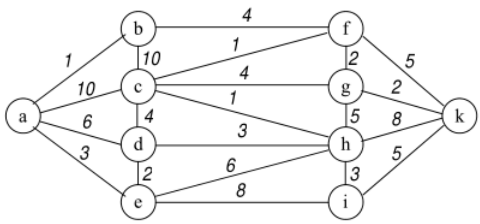
8.4 Bài tập

▷ **8.1.** Dùng thuật toán Dijkstra tìm đường đi ngắn nhất từ đỉnh *a* đến các đỉnh khác trong đồ thị sau



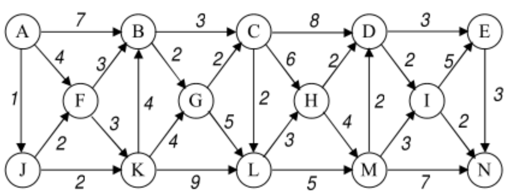
Hình 8.18. Tìm đường đi ngắn nhất từ *a* đến các đỉnh

▷ **8.2.** Dùng thuật toán Dijkstra tìm đường đi ngắn nhất từ đỉnh *a* đến các đỉnh khác trong đồ thị sau



Hình 8.19. Tìm đường đi ngắn nhất từ *a* đến các đỉnh

▷ **8.3.** Cho đồ thị có trọng số như hình dưới đây. Hãy tìm đường đi ngắn nhất từ đỉnh A đến đỉnh N.

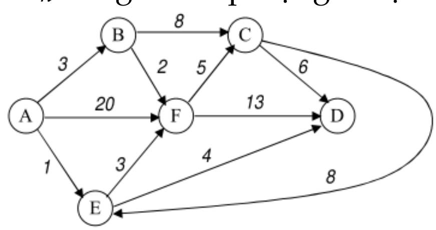


Hình 8.20. Tìm đường đi ngắn nhất từ A đến N

▷ **8.4.** Tìm đường đi ngắn nhất từ B đến các đỉnh khác của đồ thị có ma trận trọng số là

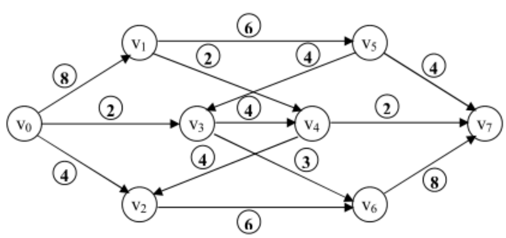
$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \end{array}\begin{array}{c} \begin{array}{ccccccc} A & B & C & D & E & F & G \end{array} \\ \begin{pmatrix} \infty & 3 & 6 & \infty & \infty & \infty & \infty \\ 3 & \infty & 2 & 4 & \infty & \infty & \infty \\ 6 & 2 & \infty & 1 & 4 & 2 & \infty \\ \infty & 4 & 1 & \infty & 2 & \infty & 4 \\ \infty & \infty & 4 & 2 & \infty & 2 & 1 \\ \infty & \infty & 2 & \infty & 2 & \infty & 4 \\ \infty & \infty & \infty & 4 & 1 & 4 & \infty \end{pmatrix} \end{array}$$

▷ **8.5.** Tìm $W^* = W_n$ bằng cách áp dụng thuật toán Floyd vào đồ thị sau



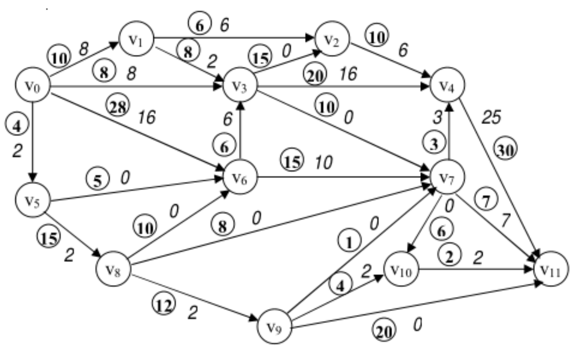
Hình 8.21. Tìm ma trận khoảng cách nhỏ nhất của đồ thị G

▷ **8.6.** Giải bài toán mạng vận tải sau bằng thuật toán Ford-Fulkerson với luồng vận tải khởi đầu bằng 0.



Hình 8.22. Tìm luồng vận tải tối ưu

▷ **8.7.** Giải bài toán mạng vận tải sau bằng thuật toán Ford-Fulkerson với luồng vận tải khởi đầu được cho kèm theo.



Hình 8.23. Tìm luồng vận tải tối ưu

Chương

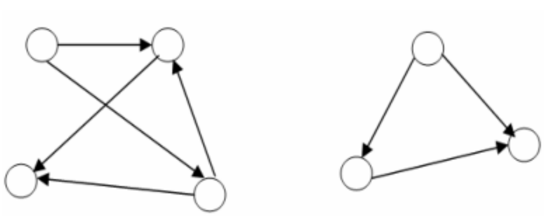
9 ĐỒ THỊ PHẲNG VÀ TÔ MÀU ĐỒ THỊ

9.1 Chu số của đồ thị

Cho đồ thị $G = (V, E)$ có n đỉnh, m cạnh, p thành phần liên thông.

Định nghĩa 9.1. Đại lượng $c = m - n + p$ được gọi là chu số của đồ thị G.

Ví dụ 9.1. Xét đồ thị sau đây



Hình 9.1. Đồ thị định hướng không liên thông

Đồ thị trên có $n = 7, m = 8$ và $p = 2$. Vậy chu số $c = 8 - 7 + 2 = 3$.

Trước hết, ta xét các tính chất của chu số.

Định lý 9.1. *Nếu thêm một cạnh mới vào đồ thị G thì chu số tăng thêm 1 hoặc không thay đổi.*

Chứng minh. Giả sử thêm cạnh mới (a, b) vào đồ thị G. Khi đó m tăng thêm 1.

i) Nếu hai đỉnh a, b thuộc cùng một mảng liên thông trong G thì n, p không đổi, do vậy chu số tăng thêm 1.

ii) Nếu hai đỉnh a, b nằm ở hai mảng liên thông khác nhau trong G thì p giảm 1, do vậy chu số không đổi.

Hệ quả 9.1. *Chu số của đồ thị là số nguyên không âm.*

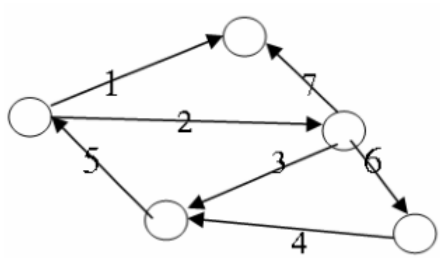
Chứng minh. Thật vậy, đồ thị G được xây dựng từ đồ thị G_0 gồm n đỉnh và không có cạnh nào cả. Sau đó, lần lượt thêm các cạnh vào đồ thị G_0 để được đồ thị G.

Chu số của G_0 là $c = 0 - n + n = 0$. Quá trình thêm cạnh không làm giảm chu số. Vậy chu số của G lớn hơn hoặc bằng chu số của $G_0 = 0$.

Bây giờ, ta đi tìm ý nghĩa của chu số.

Ta đánh số các cạnh của đồ thị G theo một thứ tự nào đó: $1, 2, ..., m$. Với mỗi chu trình vô hướng trong đồ thị G ta chọn một chiều thuận và biểu diễn nó bằng một vectơ m chiều $(q_1, q_2, ..., q_m)$ mà q_i là số lần xuất hiện của cạnh thứ i trong chu trình theo chiều thuận trừ đi số lần xuất hiện của cạnh đó trong chu trình theo chiều ngược.

Ví dụ 9.2. Xét đồ thị định hướng sau đây.



Hình 9.2. Đánh số các cạnh của đồ thị

Đồ thị có 7 cạnh, được đánh số như hình vẽ. Với chu trình vô hướng $[e_1, e_2, e_7]$ ta chọn chiều thuận là chiều $e_1e_2e_7$ khi đó vectơ tương ứng sẽ là $(-1, 1, 0, 0, 0, 0, 1)$.

Do vây, ta có thể đồng nhất mỗi chu trình vô hướng với một vectơ biểu diễn nó.

Định nghĩa 9.2. Các chu trình vô hướng $t_1, t_2, ..., t_k$ được gọi là *độc lập tuyến tính* nếu các vectơ tương ứng với chúng lập thành một hệ độc lập tuyến tính.

Hệ chu trình đơn vô hướng $t_1, t_2, ..., t_k$ được gọi là *độc lập tuyến tính cực đại* nếu nó là độc lập tuyến tính và mỗi chu trình vô hướng của đồ thị đều có thể biểu diễn tuyến tính qua các chu trình của hệ.

Định lý 9.2. *Chu số của đồ thị bằng số các chu trình đơn vô hướng độc lập cực đại trong đồ thị đó.*

Chứng minh. Quy nạp theo số cạnh m của đồ thị.

- Nếu $m = 0$ thì chu số bằng 0, đồ thị không có chu trình đơn nào.

- $(m) \Rightarrow (m+1)$: Giả sử đồ thị G' có n đỉnh, $m+1$ cạnh, p mảng liên thông. Có thể xem G' được xây dựng từ đồ thị G gồm m cạnh và bổ sung thêm một cạnh mới $e = (a, b)$. Đánh số cạnh e là cạnh thứ $m+1$ của đồ thị G'.

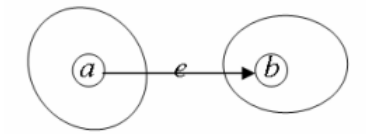
Theo giả thiết quy nạp, chu số của đồ thị G là $c(G) = m - n + p =$ số chu trình đơn vô hướng độc lập cực đại trong G. Ký hiệu các chu trình đó là: $(T) = t_1, t_2, ..., t_c$

Hiển nhiên, mỗi chu trình trong G' không chứa e đều có thể biểu diễn tuyến tính qua hệ các chu trình (T).

Ta xét hai trường hợp:

1) Hai đỉnh a, b của cạnh e nằm trong hai mảng liên thông khác nhau của G. Vì số cạnh tăng 1 nhưng số mảng liên thông bị giảm 1 nên chu số của G' vẫn bằng chu số của G.

Mặt khác, mỗi chu trình trong G' chứa e có tính chất sau đây: số lần e xuất hiện trong chu trình theo chiều thuận bằng số lần e



Hình 9.3. Hai mảng liên thông

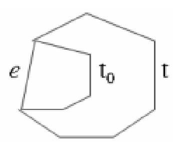
xuất hiện trong chu trình theo chiều ngược vì cạnh e là cầu nối duy nhất giữa hai mảng liên thông này của G. Do đó, thành phần thứ $m+1$ của vectơ biểu diễn chu trình này bằng 0, và chu trình này vẫn có thể biểu diễn qua hệ (T). Suy ra hệ (T) cũng chính là hệ chu trình đơn vô hướng độc lập cực đại của G'.

2) Hai đỉnh a, b của cạnh e thuộc cùng một mảng liên thông của G.

Khi đó chu số $c(G') = c(G) + 1$. Chọn một đường đi đơn vô hướng trong G nối a với b rồi ghép thêm cạnh e ta được một chu trình đơn vô hướng trong G'. Ký hiệu chu trình này là t_0

Xét hệ $(T') = t_0, (T) = t_0, t_1, t_2, ..., t_c$ gồm $c(G) + 1$ chu trình đơn vô hướng trong G'.

Hệ (T') là độc lập tuyến tính vì (T) độc lập tuyến tính và t_0 không thể biểu diễn được qua (T), vì toạ độ thứ $m+1$ của vectơ biểu diễn t_0 bằng 1, còn của các vectơ biểu diễn các chu trình trong (T) thì bằng 0.



Hình 9.4. Hai chu trình chung một cạnh

Giả sử t là một chu trình nào đó của G' chứa e. Chọn chiều của t sao cho chu trình tổng $t + t_0$ không chứa e. Vậy thì chu trình tổng $t + t_0$ có thể biểu diễn tuyến tính qua hệ (T). Do đó, chu trình t cũng có thể biểu diễn tuyến tính qua hệ (T). Vậy (T') là hệ chu trình đơn vô hướng độc lập cực đại của G'. ?

Đồ thị có chu số bằng 0 được gọi là *đồ thị phi chu trình*. Lớp đồ thị phi chu trình là lớp đặc biệt nhưng hay gặp trong thực tế ứng dụng. Trước hết ta chỉ ra một đặc trưng của lớp đồ thị này như sau.

Định lý 9.3. *Đồ thị định hướng $G = (V, E)$ là phi chu trình khi và chỉ khi các đỉnh của nó luôn có thể đánh số để sao cho mỗi cạnh (i, j) của đồ thị đều thoả mãn $i < j$.*

Chứng minh. a) Nếu có thể đánh số các đỉnh như trên thì hiển nhiên đồ thị không có chu trình.

b) Để chứng minh điều ngược lại, ta xây dựng thuật toán sau đây để đánh số các đỉnh của đồ thị định hướng phi chu trình.

Thuật toán dựa trên một tính chất rất đơn giản: Trong một đồ thị định hướng không rỗng phi chu trình tuỳ ý, luôn tồn tại đỉnh mà không có một cạnh nào đi vào đỉnh đó. Trước hết, thuật toán tính bậc vào cho các đỉnh của đồ thị.

Những đỉnh có bậc vào bằng 0 sẽ được đưa vào stack (ngăn xếp – LIFO).

Đánh số cho đỉnh đang ở đỉnh stack, loại bỏ đỉnh này khỏi stack và giảm bậc vào cho các đỉnh kề với đỉnh này. Nếu có đỉnh mà bậc vào đã giảm hết thì nạp nó lên đỉnh của stack.

Tiếp tục quá trình đánh số tăng dần, loại đỉnh, giảm bậc vào ... cho đến khi stack trở thành rỗng. Và ta đã đánh số xong tất cả các đỉnh của đồ thị. ?

Dựa vào chứng minh của định lý trên, ta xây dựng thuật toán đánh số các đỉnh cho đồ thị định hướng phi chu trình như sau.

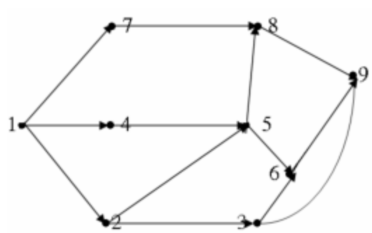
Thuật toán 9.1. Đánh số các đỉnh của đồ thị phi chu trình

Đầu vào: Biểu diễn mảng *DK* các danh sách kề của đồ thị phi chu trình *G*.
Đầu ra: Mảng *SO* các số nguyên với $SO[v]$ là số đánh trên đỉnh v.
begin
  for $v \in V$ **do** $BAC_V[v] := 0$; //$BAC\_V[v]$ *chứa bậc vào của đỉnh* v
  for $u \in V$ **do**
    for $v \in DK[u]$ **do** $BAC_V[v] := BAC_V[v] + 1$;
  $S := \emptyset$;

for $v \in V$ **do**
    if $BAC_V[v] = 0$ **then** push v onto S ;
$k := 0$;
while $S \neq \emptyset$ **do**
    begin $u := top(S); pop(S)$;
      $k := k + 1; SO[u] := k$;
      for $v \in DK[u]$ **do**
          begin $BAC_V[v] := BAC_V[v] - 1$;
            if $BAC_V[v] = 0$ **then** push v onto S
        end
      end
end .

Độ phức tạp của thuật toán là O(m+n).

Ví dụ 9.3. Áp dụng thuật toán trên để đánh số các đỉnh cho đồ thị phi chu trình sau.



Hình 9.5. Các đỉnh của đồ thị phi chu trình đã được đánh số

Việc đánh số các đỉnh trên đồ thị định hướng phi chu trình có nhiều ứng dụng trong sơ đồ PERT, phương pháp đường tới hạn CPM ...

9.2 Sắc số của đồ thị

Khái niệm sắc số liên quan đến bài toán tô màu đồ thị như sau:

Hãy tô màu các đỉnh của một đồ thị đã cho, sao cho hai đỉnh kề nhau phải được tô bằng hai màu khác nhau.

Ta nói rằng, đồ thị G tô được bằng k màu nếu tồn tại hàm:

$$m : V \to \{0, 1, 2, ..., k-1\}$$

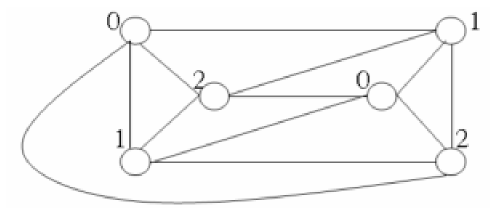
sao cho, nếu hai đỉnh x và y kề nhau thì $m(x) \neq m(y)$.

Dễ thấy rằng, đồ thị G tô màu được khi và chỉ khi nó không có đỉnh nút.

Định nghĩa 9.3. *Sắc số* của một đồ thị chính là số màu ít nhất dùng để tô các đỉnh của đồ thị đó.

Ta ký hiệu số s là sắc số của đồ thị G. Hiển nhiên $s \leq n$, số màu không vượt quá số đỉnh của đồ thị.

Ví dụ 9.4. Hãy tô màu đồ thị sau đây.



Hình 9.6. Tô màu các đỉnh đồ thị

Đồ thị trên có sắc số bằng 3.

Nhận xét: Mỗi cách tô màu m cho đồ thị G sẽ ứng với một cách phân hoạch tập đỉnh V thành các tập ổn định trong không giao nhau, 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.

Định lý 9.4. *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 chứng minh bằng quy nạp theo số n.

$n = 1$: Chu trình gồm 3 đỉnh, mà hai đỉnh bất kỳ đều kề nhau. Vậy ta phải dùng đúng 3 màu để tô các đỉnh.

$(n) \Rightarrow (n+1)$: Giả sử α là một chu trình có độ dài $2(n+1) + 1 = 2n + 3$ với dãy các đỉnh là $[x_1, x_2, ..., x_{2n+1}, x_{2n+2}, x_{2n+3}]$.

Nối x_1 với x_{2n+1} ta được một chu trình α' có độ dài $2n+1$. Theo giả thiết quy nạp, chu trình α' có sắc số bằng 3. Lấy màu của x_1 tô cho x_{2n+2}, còn màu của x_{2n+1} tô cho x_{2n+3}. Chu trình α đã được tô màu mà không phải thêm màu mới. Vậy chu trình α có sắc số bằng 3.

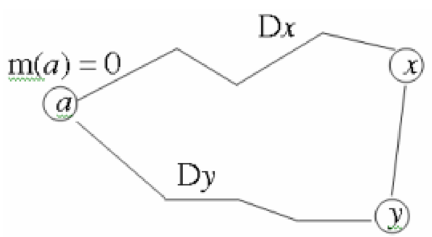
Định lý 9.5. *Đồ thị đầy đủ n đỉnh K_n có sắc số bằng n.*

Dưới đây là một tiêu chuẩn đơn giản để kiểm tra xem một đồ thị có hai sắc (sắc số bằng 2) hay không.

Định lý 9.6. *Giả sử đồ thị G có ít nhất một cạnh. Đồ thị G là hai sắc khi và chỉ khi G không có chu trình đơn vô hướng độ dài lẻ.*

Chứng minh. Giả sử G là đồ thị hai sắc. Theo Định lý 9.4 thì G không thể có chu trình đơn vô hướng độ dài lẻ.

Ngược lại, giả sử G không có chu trình đơn vô hướng độ dài lẻ. Không mất tính tổng quát có thể xem G là liên thông. Chọn một đỉnh a nào đó trong đồ thị. Đặt $m(a) = 0$.



Hình 9.7. Cách xây dựng hàm tô màu

Với $x \neq a$ ta ký hiệu $d(x)$ là độ dài đường đi vô hướng ngắn nhất nối a với x.

Đặt $m(x) = d(x) \mod 2$. Ta sẽ chứng minh m là hàm màu của G.

Giả sử x, y kề nhau. Lấy Dx là đường đi vô hướng ngắn nhất nối a với x có độ dài $d(x)$, và Dy là đường đi vô hướng ngắn nhất nối a với y có độ dài $d(y)$. Chu trình đơn $[Dx, (x,y), Dy]$ có độ dài $d(x) + d(y) + 1$ phải là một số chẵn.

Vậy thì $d(x) + d(y)$ là một số lẻ, có nghĩa là $d(x)$ và $d(y)$ khác nhau tính chẵn lẻ. Do vậy: $m(x) \neq m(y)$.

Hàm tô màu m có hai giá trị, vậy sắc số ≤ 2. G có ít nhất một cạnh nên sắc số của nó bằng 2.

Từ định lý trên chúng ta có hệ quả sau đây.

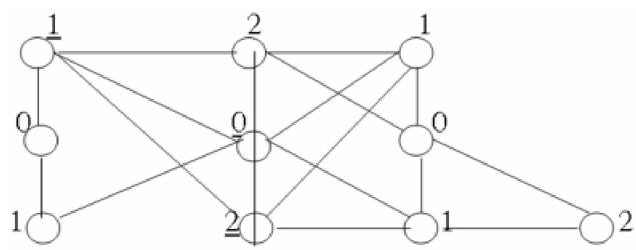
Hệ quả 9.2. *Tất cả các chu trình độ dài chẵn đều có sắc số bằng 2.*

Thuật toán 9.2. Tô màu đồ thị không có đỉnh nút

1. Liệt kê các đỉnh $x_1, x_2, ..., x_n$ của đồ thị theo thứ tự giảm dần của bậc:$r(x_1) \geq r(x_2) \geq ... \geq r(x_n)$ để làm giảm các phép kiểm tra ở bước dưới.
2. Tô màu 0 cho đỉnh x_1 (đỉnh có bậc lớn nhất) cùng các đỉnh không kề với $x_1$và không kề với các đỉnh đã tô màu 0.
3. Lặp lại thủ tục tô màu $i+1$ giống như thủ tục tô màu i cho đến khi tô màuhết các đỉnh của đồ thị.

Số màu đã dùng chính là sắc số của đồ thị.

Ví dụ 9.5. Tô màu đồ thị sau đây.

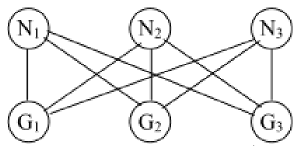


Hình 9.8. Tô màu một đồ thị

Định lý 9.7. *Nếu bậc lớn nhất của các đỉnh trong đồ thị G là r thì sắc số của đồ thị $G \leq r+1$.*

9.3 Đồ thị phẳng

Từ xa xưa đã lưu truyền một bài toán cổ "Ba nhà, ba giếng": Có ba nhà ở gần ba cái giếng, nhưng không có đường nối thẳng các nhà với nhau cũng như không có đường nối thẳng các giếng với nhau. Có lần bất hoà với nhau, họ tìm cách làm các đường



Hình 9.9. Bài toán 3 giếng nước

khác đến giếng sao cho các đường này đôi một không giao nhau. Họ có thực hiện được ý định đó không?

Bài toán này có thể được mô hình bằng đồ thị phân đôi đầy đủ K 3,3 . Câu hỏi ban đầu có thể diễn đạt như sau: Có thể vẽ $K_{3,3}$ trên một mặt phẳng sao cho không có hai cạnh nào cắt nhau? Trong chương này chúng ta sẽ nghiên cứu bài toán: có thể vẽ một đồ thị trên một mặt phẳng không có các cạnh nào cắt nhau không. Đặc biệt chúng ta sẽ trả lời bài toán ba nhà ba giếng. Thường có nhiều cách biểu diễn đồ thị. Khi nào có thể tìm được ít nhất một cách biểu diễn đồ thị không có cạnh cắt nhau?

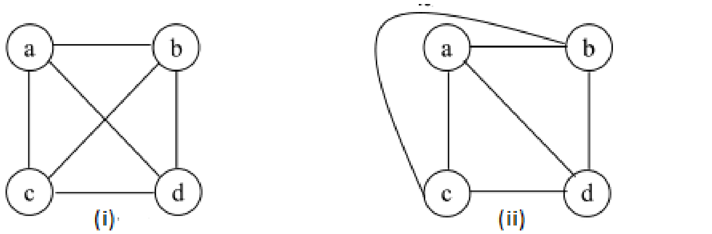
Định nghĩa 9.4. Một đồ thị được gọi là *phẳng* nếu nó có thể vẽ được trên một mặt phẳng mà không có các cạnh nào cắt nhau (ở một điểm không phải là điểm mút của các cạnh). Hình vẽ như thế gọi là một biểu diễn phẳng của đồ thị.

Một đồ thị có thể là phẳng ngay cả khi nó thường được vẽ với những cạnh cắt nhau, vì có thể vẽ nó bằng cách khác không có các cạnh cắt nhau.

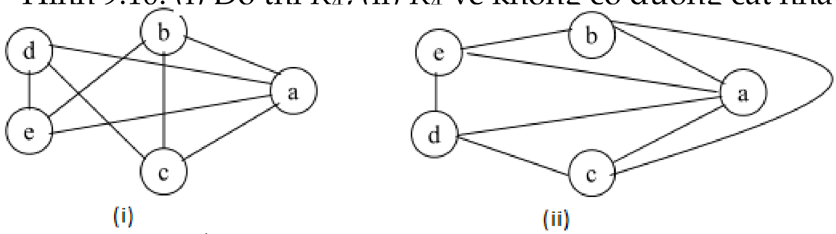
Ví dụ 9.6. 1) Một cây, một chu trình đơn là một đồ thị phẳng.

2) K_4 là đồ thị phẳng bởi vì có thể vẽ lại như hình bên không có đường cắt nhau. 3) Xét đồ thị G như trong hình a dưới đây. Có thể biểu diễn G một cách khác như trong hình *b*, trong đó bất kỳ hai cạnh nào cũng không cắt nhau. 4) Đồ thị đầy đủ K_5 là một thí dụ về đồ thị không phẳng (xem phần sau).

Định nghĩa 9.5. Cho G là một đồ thị phẳng. Mỗi phần mặt phẳng giới hạn bởi một chu trình đơn không chứa bên trong nó một chu



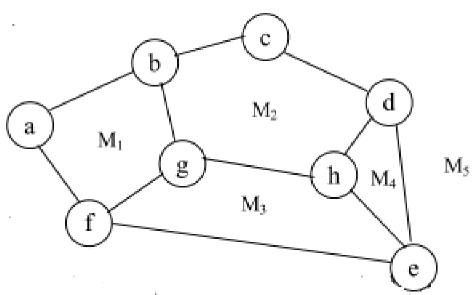
Hình 9.10. (i) Đồ thị K_4, (ii) K_4 vẽ không có đường cắt nhau



Hình 9.11. (i) Đồ thị có cạng giao nhau,
(ii) Có thể vẽ lại không có đường cắt nhau

trình đơn khác, gọi là *một miền* (hữu hạn) của đồ thị G. Chu trình giới hạn miền là *biên của miền*. Mỗi đồ thị phẳng liên thông có một *miền vô hạn* duy nhất (là phần mặt phẳng bên ngoài tất cả các miền hữu hạn). Số cạnh ít nhất tạo thành biên gọi là *đai* của G; trường hợp nếu G không có chu trình thì đai chính là số cạnh của G.

Ví dụ 9.7. Đồ thị phẳng ở hình 9.12 có 5 miền, M_5 là miền vô hạn, miền M_1 có biên $abgfa$, miền M_2 có biên là $bcdhgb$, ... Chu trình đơn $abcdhgfa$ không giới hạn một miền vì chứa bên trong nó chu trình đơn khác là $abgfa$.



Hình 9.12. Miền của đồ thị phẳng

Ký hiệu: h là số diện hữu hạn của một đồ thị phẳng.

Ta sẽ thấy rằng, hệ chu trình đơn độc lập cực đại sẽ chia đồ thị phẳng thành các diện hữu hạn. Thật vậy,

Định lý 9.8. *Số diện hữu hạn của một đa đồ thị phẳng G bằng chu số của đồ thị này.*

Chứng minh. Quy nạp theo số diện hữu hạn h của G.

$h = 1$: chỉ có một chu trình đơn duy nhất, đó chính là biên của diện này. Suy ra chu số bằng 1.

$(h-1) \Rightarrow (h)$: Giả sử đồ thị phẳng G với n đỉnh, m cạnh và p mảng liên thông có h diện. Lập đồ thị G' từ G bằng cách bớt đi cạnh e nào đó trên biên của một diện để số diện hữu hạn bớt đi 1. Khi đó, G' có $h-1$ diện. Theo giả thiết quy nạp, chu số của G' là $h-1 = (m-1) - n + p$ (p không đổi vì chỉ bớt đi một cạnh trên chu trình). Suy ra số diện hữu hạn của G là $h = m - n + p =$ chu số của G.

Hệ quả 9.3. *Nếu đa đồ thị phẳng G có n đỉnh, m cạnh, p mảng liên thông và h diện thì: $n - m + h = p + 1$ (công thức Euler tổng quát).*

Chứng minh. Số diện của đồ thị phẳng bằng số diện hữu hạn cộng thêm 1 (diện vô hạn) = chu số + 1. Vậy thì, $h = m - n + p + 1$. Do đó, $n - m + h = p + 1$.

Hệ quả 9.4. *Trong một đơn đồ thị phẳng có ít nhất một đỉnh có bậc không quá 5.*

Chứng minh. Không mất tính tổng quát có thể giả thiết rằng đơn đồ thị là liên thông. Trong đơn đồ thị phẳng mỗi diện hữu hạn được giới hạn bởi ít nhất 3 cạnh, mà mỗi cạnh thuộc nhiều nhất là hai diện nên

$$3h \leq 2m \Rightarrow h \leq \frac{2m}{3}.$$

Ta chứng minh bằng phản chứng. Giả sử mọi đỉnh của đồ thị G đều có bậc ít nhất là bằng 6. Khi đó, tổng tất cả các bậc của các đỉnh của trong $G = 2m \geq 6n$. Do vậy: $m \geq 3n$ hay $n \leq \dfrac{m}{3}$.

Theo công thức Euler thì $n - m + h = 1 + p = 2$. Ta có $2 \leq \frac{m}{3} - m + \frac{2m}{3} = 0$ Suy ra điều vô lý.

Các điều kiện cho tính phẳng của đồ thị.

Với điều kiện nào đảm bảo cho một đồ thị là phẳng. Để trả lời cho câu hỏi này ta dựa vào một số khái niệm được định nghĩa dưới đây. Trước hết, ta có ngay những kết quả hiển nhiên sau đây.

Định lý 9.9. *Giả sử G là một đồ thị và G' là đồ thị con của nó.*
Đồ thị G phẳng thì G' cũng phẳng.
Đồ thị G' không phẳng thì G cũng không phẳng.

Chứng minh. Hiển nhiên.

Ký hiệu: δ là độ dài của chu trình ngắn nhất hoặc là số cạnh của đồ thị G nếu nó không có chu trình. Số δ được gọi là *đai* của đồ thị.

Định lý 9.10. *Nếu đồ thị G là phẳng và đai của nó $\delta \geq 3$ thì*

$$m \leq \frac{\delta}{\delta - 2}(n - 2).$$

Chứng minh. Ta có: $h.\delta \leq 2m$. Do vậy theo công thức Euler thì $\delta.(m - n + 2) \leq 2m$. Từ đó suy ra điều phải chứng minh.

Hệ quả 9.5. *Đồ thị hai phần đầy đủ $K_{m,n}$ là đồ thị phẳng khi và chỉ khi $m \leq 2$ hoặc $n \leq 2$.*

9.4 Đồ thị không phẳng

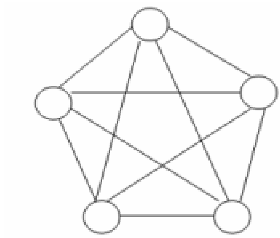
Định lý 9.11. *Đồ thị phân đôi đầy đủ $K_{3,3}$ là một đồ thị không phẳng.*

Chứng minh. Ta thấy $\delta = 4$, mà $m = 9 > \frac{4}{4-2}(6 - 2) = 8$. theo Định lý 9.10, đồ thị Hình 9.9 là không phẳng.

Như vậy định lý này cho ta lời giải của bài toán "Ba nhà ba giếng", nghĩa là không thể thực hiện được việc làm các đường khác đến giếng sao cho các đường này đôi một không giao nhau.

Định lý 9.12. *Đồ thị đầy đủ K_5 là một đồ thị không phẳng.*

Chứng minh. Đồ thị đầy đủ 5 đỉnh



Hình 9.13. Đồ thị đầy đủ 5 đỉnh (K_5)

Đồ thị này có đai $\delta = 3$. Vậy $m = 10 > \dfrac{3}{3-2}(5-2) = 9$. Do đó, đồ thị K_5 không phẳng. Từ đó suy ra, đồ thị đầy đủ K_n với $n \geq 5$ là không phẳng. Chú ý rằng, đồ thị đầy đủ K_n với $n \leq 4$ là đồ thị phẳng.

Định nghĩa 9.6. Từ đồ thị G' cho trước ta xây dựng đồ thị G bằng cách: Thêm vào G' các đỉnh mới và các cạnh mới. Đỉnh mới có thể nối với một đỉnh khác bằng một cạnh mới. Đỉnh mới cũng có thể đặt trên một cạnh cũ và chia cạnh này thành hai cạnh mới. Ta nói rằng, đồ thị G nhận được có chứa cấu hình G'. Hay đồ thị G' là một *cấu hình* của đồ thị G.

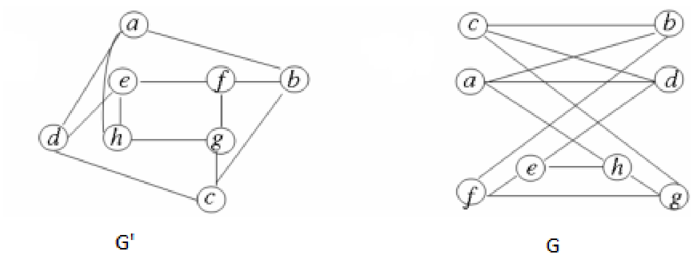
Chẳng hạn, đồ thị riêng của đồ thị là một cấu hình của đồ thị này.

Định lý 9.13 (Kuratowski). *Đồ thị là phẳng khi và chỉ khi nó không chứa cấu hình $K_{3,3}$ hoặc K_5.*

Ta có thể áp dụng định lý Kuratowski để xét tính chất phẳng của đồ thị.

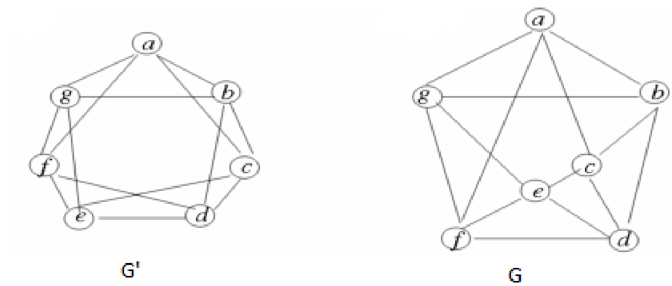
Ví dụ 9.8. Xét các đồ thị sau đây

Đồ thị này chứa cấu hình $K_{3,3}$. Do vậy, nó không phẳng.



Hình 9.14. Hai đồ thị đẳng hình chứa cấu hình $K_{3,3}$

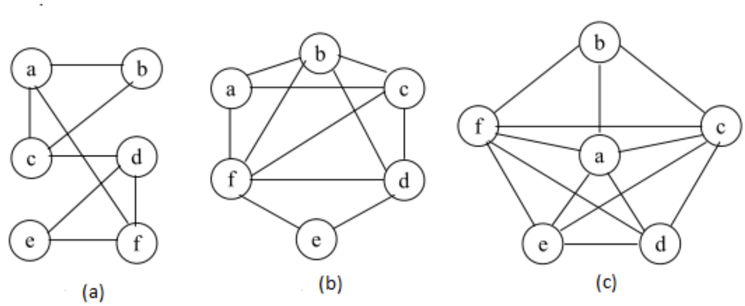
Ví dụ 9.9. Xét các đồ thị sau đây



Hình 9.15. Hai đồ thị đẳng hình chứa cấu hình K_5

Đồ thị này chứa cấu hình K_5. Vậy nó là không phẳng.

Ví dụ 9.10. Xét các hình sau



Hình 9.16. (a), (b) là đồ thị phẳng, (c) là đồ thị không phẳng

Đồ thị trong hình (a) và (b) là đồ thị phẳng. Các đồ thị này có 6 đỉnh, nhưng không chứa đồ thị con $K_{3,3}$ được vì có đỉnh bậc 2, trong khi tất cả các đỉnh của $K_{3,3}$ đều có bậc 3; cũng không thể chứa đồ thị con K_5 được vì có những đỉnh bậc nhỏ hơn 4, trong khi tất cả các đỉnh của K_5 đều có bậc 4.

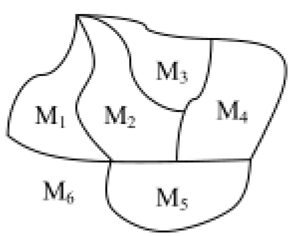
Đồ thị trong hình (c) là đồ thị không phẳng vì nếu xoá đỉnh b cùng các cạnh (b,a), (b,c), (b,f) ta được đồ thị con là K_5.

9.5 Tô mầu đồ thị

Định nghĩa 9.7. Mỗi bản đồ có thể coi là một đồ thị phẳng. Trong một bản đồ, ta coi hai miền có chung nhau một đường biên 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. Ta gọi một cách tô màu bản đồ như vậy là một *cách tô màu đúng*.

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 việc làm đó nói chung là không hợp lý. Nếu bản đồ có nhiều miền thì sẽ rất khó phân biệt những màu gần giống nhau. Do vậy người ta chỉ dùng một số màu cần thiết để tô bản đồ. Một bài toán được đặt ra là: xác định số màu tối thiểu cần có để tô màu đúng một bản đồ.

Ví dụ 9.11. Bản đồ trong hình bên có 6 miền, nhưng chỉ cần có 3 màu (vàng, đỏ, xanh) để tô đúng bản đồ này. Chẳng hạn, màu vàng được tô cho $M1$ và $M4$, màu đỏ được tô cho $M2$ và $M6$, màu xanh được tô cho $M3$ và $M5$



Hình 9.17. Tô màu bản đồ

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; 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. Rõ ràng mọi bản đồ trên mặt phẳng đều có đồ thị đối ngẫu phẳng. Bài toán tô màu các miền của bản đồ là tương đương với bài toán tô màu các đỉnh của đồ thị đối ngẫu sao cho không có hai đỉnh liền kề nhau có cùng một màu, mà ta gọi là tô màu đúng các đỉnh của đồ thị.

Mệnh đề 9.1. *Với mỗi số nguyên dương n, tồn tại một đồ thị không chứa K_3 và có sắc số bằng n.*

Chứng minh. Ta chứng minh mệnh đề bằng quy nạp theo n.

Trường hợp $n = 1$ là hiển nhiên.

Giả sử ta có đồ thị G_n với k_n đỉnh, không chứa K_3 và có sắc số là n. Ta xây dựng đồ thị G_{n+1} gồm n bản sao của G_n và thêm k_n^n đỉnh mới theo cách sau: mỗi bộ thứ tự $(v_1, v_2, \ldots, v_n)$, với v_i thuộc bản sao G_n thứ i, sẽ tương ứng với một đỉnh mới, đỉnh mới này được nối bằng n cạnh mới đến các đỉnh $v_1, v_2, \ldots, v_n$. Dễ thấy rằng G_{n+1} không chứa K_3 và có sắc số là $n + 1$.

Định lý 9.14 (Định lý 5 màu của Kempe-Heawood). *Mọi đồ thị phẳng không có đỉnh nút đều có sắc số không lớn hơn 5.*

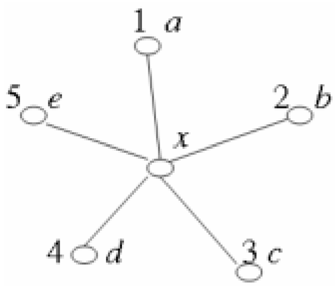
Chứng minh. Ta chứng minh định lý bằng quy nạp theo số đỉnh n của đồ thị. $n = 1, 2, 3, 4, 5$: Hiển nhiên đúng.

$(n - 1) \Rightarrow (n)$: Theo Hệ quả 9.4, đồ thị G có ít nhất một đỉnh x với bậc không quá 5. Xây dựng đồ thị G' từ đồ thị G bằng cách bỏ đỉnh x. Theo giả thiết quy nạp, đồ thị G' có sắc số không vượt quá 5.

Lấy một cách tô màu nào đấy của G'.

Nếu các đỉnh kề với đỉnh x được tô bằng ít hơn 5 màu thì vẫn còn thừa màu để tô cho x. Sắc số của G bằng sắc số của G' (không vượt quá 5).

Vậy ta chỉ cần xét trường hợp đỉnh x kề với 5 đỉnh và các đỉnh kề với x được đánh số thứ tự theo chiều kim đồng hồ và tô bằng 5 màu như Hình 9.18 dưới đây. Khi đó, ta phải đổi màu của các đỉnh trên để dành ra màu cho đỉnh x.



Hình 9.18. Năm đỉnh kề với 5 màu

Xét tất cả các đường đi trong G bắt đầu từ đỉnh a và gồm các đỉnh chỉ tô bằng màu 1 và màu 3. Trong các đường này nếu không có đường nào đi qua đỉnh c thì ta có thể tráo đổi màu 1 với màu 3 cho tất cả các đỉnh trên các đường đi ấy. Sau đó, ta tô màu 1 cho đỉnh x.

Ngược lại, nếu có một đường đi từ a đến c gồm toàn các đỉnh được tô bằng các màu 1 và màu 3 thì đường này cùng với hai cạnh (c, x) và (x, a) sẽ tạo thành một chu trình trong G. Do tính chất phẳng của đồ thị 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. Vậy ta lại có thể tráo đổi màu 2 với 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. Ta tô màu 2 cho đỉnh x. Định lý được chứng minh.

Định lý Bốn màu 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. Mặt khác, dùng phương pháp của Kempe, Heawood đã chứng minh được "bài toán năm màu" (tức là mọi bản đồ có thể tô đúng bằng 5 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. Cách chứng minh này đã gây ra nhiều cuộc tranh cãi vì máy tính đã đóng vai trò quan trọng biết bao. 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?

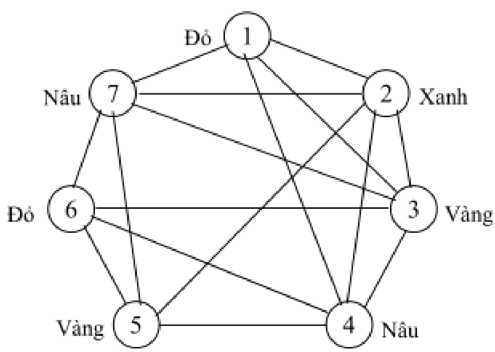
Định lý 9.15 (**Appel - Haken**). *Mọi đồ thị phẳng không có đỉnh nút đều có sắc số không quá 4.*

9.6 Ứng dụng tô mầu đồ thị

1) 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.*

Có thể giải bài toán lập lịch thi bằng mô hình đồ thị, với các đỉnh là các môn thi, 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. 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 9.19. Lập lịch thi

2) 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.

3) Các thanh ghi chỉ số: Trong các bộ dịch hiệu quả cao việc thực hiện các vòng lặp được tăng tốc khi các biến dùng thường xuyên được lưu tạm thời trong các thanh ghi chỉ số của bộ xử lý trung tâm (CPU) mà không phải ở trong bộ nhớ thông thường. Với một vòng lặp cho trước cần bao nhiêu thanh ghi chỉ số? Bài toán này có thể giải bằng mô hình tô màu đồ thị. Để xây dựng mô hình ta coi mỗi đỉnh của đồ thị là một biến trong vòng lặp. Giũa hai đỉnh có một cạnh nếu các biến biểu thị bằng các đỉnh này phải được lưu trong các thanh ghi chỉ số tại cùng thời điểm khi thực hiện vòng lặp. Như vậy số màu của đồ thị chính là số thanh ghi cần có vì những thanh ghi khác nhau được phân cho các biến khi các đỉnh biểu thị các biến này là liền kề trong đồ thị.

9.7 Bài tập

▷ **9.1.** Cho G là một đơn đồ thị phẳng liên thông có 10 mặt, tất cả các đỉnh đều có bậc 4. Tìm số đỉnh của đồ thị G.

▷ **9.2.** Cho G là một đơn đồ thị phẳng liên thông có 9 đỉnh, bậc các đỉnh là $2, 2, 2, 3, 3, 3, 4, 4, 5$. Tìm số cạnh và số mặt của G.

▷ **9.3.** Tìm số đỉnh, số cạnh và đai của: a) K_n; b) $K_{m,n}$.

▷ **9.4.** Chứng minh rằng

a) Kn là phẳng khi và chỉ khi $n \leq 4$.

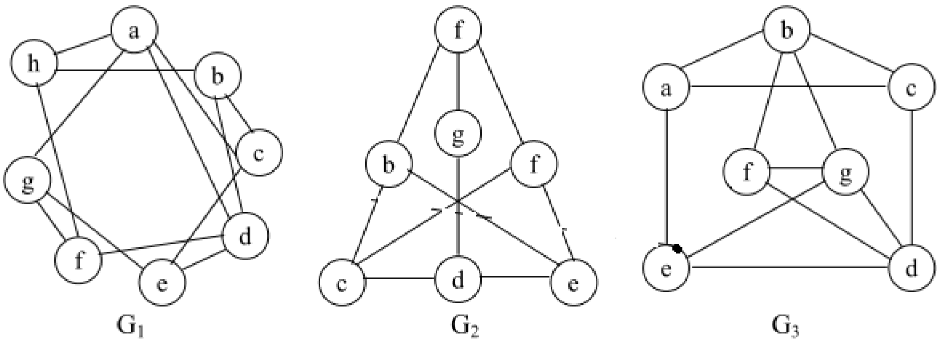
b) $K_{m,n}$ là phẳng khi và chỉ khi $m \leq 2$ hay $n \leq 2$.

▷ **9.5.** Đồ thị nào trong các đồ thị không phẳng sau đây có tính chất: Bỏ một đỉnh bất kỳ và các cạnh liên thuộc của nó tạo ra một đồ thị phẳng. a) $K5$; b) K_6; c) $K_{3,3}$.

▷ **9.6.** Cho G là một đơn đồ thị phẳng liên thông có n đỉnh và m cạnh, trong đó $n \geq 3$. Chứng minh rằng

$$m \leq 3n - 6.$$

▷ **9.7.** Trong các đồ thị ở hình dưới đây, đồ thị nào là phẳng, đồ thị nào không phẳng? Nếu đồ thị là phẳng thì có thể kẻ thêm ít nhất là bao nhiêu cạnh để được đồ thị không phẳng?



Hình 9.20. Đồ thị phẳng và không phẳng

▷ **9.8.** Chứng minh rằng đồ thị Peterson (Hình 7.26 Bài tập 7.11, Chương 7) là đồ thị không phẳng.

▷ **9.9.** Đa diện lồi có d mặt ($d \geq 5$), mà từ mỗi đỉnh có đúng 3 cạnh. Hai người chơi trò chơi như sau: mỗi người lần lượt tô đỏ một mặt trong các mặt còn lại. Người thắng là người tô được 3 mặt có chung một đỉnh. Chứng minh rằng tồn tại cách chơi mà người được tô trước luôn luôn thắng.

▷ **9.10.** Chứng minh rằng

a) Một đồ thị phẳng có thể tô đúng các đỉnh bằng hai màu khi và chỉ khi đó là đồ thị phân đôi.

b) Một đồ thị phẳng có thể tô đúng các miền bằng hai màu khi và chỉ khi đó là đồ thị Euler.

▷ **9.11.** Tìm sắc số của các đồ thị cho trong Hình 9.20.

▷ **9.12.** Tìm sắc số của các đồ thị $K_n, K_{m,n}, C_n$, và W_n.

▷ **9.13.** Khoa Toán có 6 hội đồng họp mỗi tháng một lần. Cần có bao nhiêu thời điểm họp khác nhau để đảm bảo rằng không ai bị xếp lịch họp hai hội đồng cùng một lúc, nếu các hội đồng là

$$H_1 = \{H, L, P\}, H_2 = \{L, M, T\}, H_3 = \{H, T, P\}.$$

▷ **9.14.** Một vườn bách thú muốn xây dựng chuồng tự nhiên để trưng bày các con thú. Không may, một số loại thú sẽ ăn thịt các con thú khác nếu có cơ hội. Có thể dùng mô hình đồ thị và tô màu đồ thị như thế nào để xác định số chuồng khác nhau cần có và cách nhốt các con thú vào các chuồng thú tự nhiên này?

▷ **9.15.** Chứng minh rằng một đơn đồ thị phẳng có 8 đỉnh và 13 cạnh không thể được tô đúng bằng hai màu.

▷ **9.16.** Chứng minh rằng nếu G là một đơn đồ thị phẳng có ít hơn 12 đỉnh thì tồn tại trong G một đỉnh có bậc ≤ 4. Từ đó hãy suy ra rằng đồ thị G có thể tô đúng bằng 4 màu.

Chương 10 CÂY VÀ ỨNG DỤNG

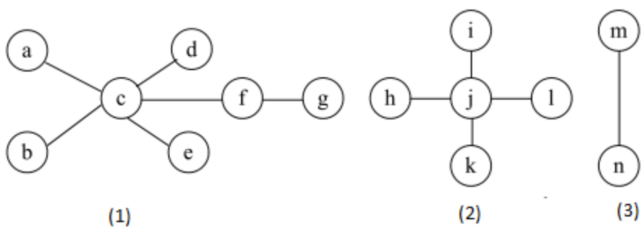
Một đồ thị liên thông và không có chu trình được gọi là cây. Cây đã được dùng từ năm 1857, khi nhà toán học Anh tên là Arthur Cayley dùng cây để xác định những dạng khác nhau của hợp chất hoá học. Từ đó cây đã được dùng để giải nhiều bài toán trong nhiều lĩnh vực khác nhau. Cây rất hay được sử dụng trong tin học. Chẳng hạn, người ta dùng cây để xây dựng các thuật toán rất có hiệu quả để định vị các phần tử trong một danh sách. Cây cũng dùng để xây dựng các mạng máy tính với chi phí rẻ nhất cho các đường điện thoại nối các máy phân tán. Cây cũng được dùng để tạo ra các mã có hiệu quả để lưu trữ và truyền dữ liệu. Dùng cây có thể mô hình các thủ tục mà để thi hành nó cần dùng một dãy các quyết định. Vì vậy cây đặc biệt có giá trị khi nghiên cứu các thuật toán sắp xếp.

10.1 Định nghĩa cây

Khái niệm cây được Cayley đưa ra đầu tiên vào năm 1857.

Định nghĩa 10.1. Giả sử $T = (V, E)$ là đồ thị vô hướng. Ta nói rằng đồ thị T là một cây nếu nó liên thông và không có chu trình.

Ví dụ 10.1. Đồ thị dưới đây là một số cây.



Hình 10.1. Một số cây

Kết quả dưới đây sẽ cho chúng ta một số tính chất lý thú và có thể dùng làm định nghĩa cho cây.

Định lý 10.1. *Với đồ thị vô hướng T có số đỉnh không ít hơn 2, các tính chất sau đây là tương đương*
(1) T là một cây.
(2) T không có chu trình và có n-1 cạnh.
(3) T liên thông và có n-1 cạnh.
(4) T không có chu trình, nhưng nếu thêm một cạnh nối hai đỉnh bất kỳ không kề nhau thì xuất hiện một chu trình.
(5) T liên thông, nhưng nếu bớt đi một cạnh bất kỳ thì sẽ mất tính liên thông.
(6) Mỗi cặp đỉnh được nối với nhau bằng đúng một đường đi đơn.

Chứng minh. Chú ý rằng đồ thị T không có chu trình khi và chỉ khi chu số của nó bằng 0, nghĩa là $m = n - p$.

(1)⇒ (2): Vì $p = 1$ và $m = n - p$ suy ra $m = n - 1$.

(2)⇒ (3): $m = n - p, m = n - 1$ cho nên $p = 1$.

(3)⇒ (4): $p = 1, m = n - 1$ suy ra $m = n - p$. Vậy thì chu số của đồ thị $T = 0$, đồ thị T không có chu trình. Thêm một cạnh vào thì m tăng thêm 1 còn n, p không đổi. Khi đó chu số $c = m - n + p = 1$. Đồ thị có một chu trình.

(4)⇒ (5): $c = 0$ nên $m = n - p$.

Giả sử ngược lại, đồ thị T không liên thông. Thế thì có ít nhất hai đỉnh a, b không liên thông. Khi thêm cạnh (a, b) vào đồ thị vẫn không làm xuất hiện chu trình. Mâu thuẫn với điều (4).

Vậy đồ thị phải liên thông, nghiã là $p = 1$. Suy ra $m = n - 1$.

Khi bớt đi một cạnh bất kỳ, đồ thị vẫn không có chu trình. Do đó $m - 1 = n - p'$. Thế thì $p' = 2$ và đồ thị mất tính liên thông.

(5)⇒ (6): Vì đồ thị T liên thông nên mỗi cặp đỉnh đều có đường đi đơn nối chúng. Giả sử cặp đỉnh a, b được nối bằng hai đường đi đơn khác nhau. Khi đó có cạnh e thuộc đường đi này nhưng không thuộc đường đi kia. Ta bỏ cạnh e này đi, đồ thị vẫn liên thông. Trái với điều (5).

(6)⇒ (1): Suy ra đồ thị T liên thông. Giả sử T có chu trình. Vậy thì giữa hai đỉnh của chu trình có thể nối bằng hai đường đơn khác nhau. Mâu thuẫn với điều 6).

10.2 Cây bao trùm của đồ thị

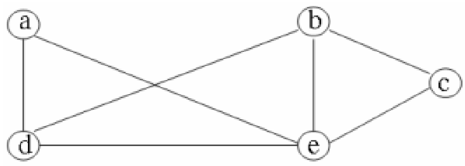
Giả sử G là một đồ thị vô hướng.

10.2.1. Cây bao trùm

Định nghĩa 10.2. Cây T được gọi là cây bao trùm của đồ thị G nếu T là một đồ thị riêng của G.

Ví dụ 10.2. Đồ thị G được cho như hình vẽ dưới đây. và một số cây bao trùm của G là

Cây bao trùm có nhiều ứng dụng trong các bài toán điều khiển giao thông, nối các mạng điện ...



Hình 10.2. Đồ thị có cây bao trùm

(1) (2)

Hình 10.3. Hai cây bao trùm của đồ thị trên

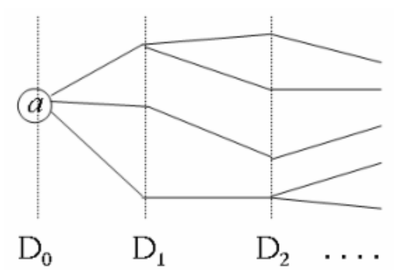
Định lý 10.2. *Đồ thị vô hướng G có cây bao trùm khi và chỉ khi G liên thông.*

Chứng minh. $\Rightarrow$ Hiển nhiên, vì cây bao trùm liên thông suy ra G liên thông.

$\Leftarrow$ Chọn a là một đỉnh bất kỳ trong đồ thị G.

Ký hiệu $d(x)$ là độ dài của đường đi ngắn nhất nối đỉnh a với đỉnh x. Lần lượt xây dựng các tập hợp

$$D_0 = \{a\}$$
$$D_1 = \{x | d(x) = i\} \text{ với } i \geq 1.$$



Hình 10.4. Cách xây dựng cây bao trùm

Chú ý: Mỗi đỉnh x thuộc $D_i (i \geq 1)$ đều có đỉnh y thuộc D_{i-1} sao cho (x, y) là một cạnh, vì nếu $< a, ..., y, x >$ là đường đi ngắn nhất nối a với x thì $< a, ..., y >$ là đường đi ngắn nhất nối a với y và $y \in D_{i-1}$.

Ta lập tập cạnh T như sau: Với mỗi đỉnh x của đồ thị G, thì $x \in D_i$ với $i \geq 1$, ta lấy một cạnh nào đó nối x với một đỉnh trong D_{i-1}. Tập cạnh này sẽ tạo nên một đồ thị riêng của G với n đỉnh và $n-1$ cạnh. Đồ thị riêng này liên thông vì mỗi đỉnh đều được nối với đỉnh a. Theo tính chất 3) của cây thì T là một cây. Do vậy, T là cây bao trùm của đồ thị G.

Định lý 10.3 (Cayley). *Số cây bao trùm của đồ thị vô hướng đầy đủ n đỉnh là n^{n-2} .*

Kết quả trên cho ta thấy số lượng cây bao trùm của một đồ thị nói chung là rất lớn.

Các thuật toán duyệt đồ thị theo chiều rộng và chiều sâu là những công cụ tốt để tìm cây bao trùm của đồ thị liên thông.

Thuật toán 10.1. Tìm cây bao trùm của đồ thị liên thông dùng phương pháp duyệt theo chiều sâu.

Đầu vào: Biểu diễn mảng DK các danh sách kề của đồ thị vô hướng G.
Đầu ra: Cây bao trùm (V, T) của đồ thị G.

```
procedure CBTS(v) ;
begin
    Duyet[v] := true;
    for u ∈ DK[v] do
    if !Duyet[u] then
    begin T := T ∪ {(v,u)}; CBTS(u) end ;
 end ;
 begin //Chương trình chính
    for u ∈ V do Duyet[u] := false;
    T := ∅;
    CBTS (z); //z là đỉnh tuỳ ý của đồ thị, sẽ trở thành gốc của cây
 end .
```

Tính đúng đắn của thuật toán được suy từ 3 tính chất sau đây

Khi ta thêm cạnh (v, u) vào tập cạnh T thì trong đồ thị (V, T)

đã có đường đi từ z tới v. Vậy thì thuật toán xây dựng lên đồ thị liên thông.

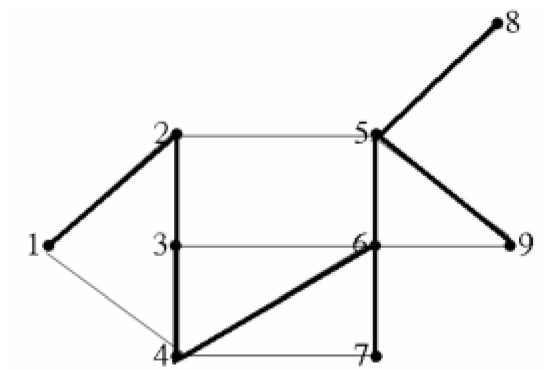
Mỗi cạnh mới (v, u) được thêm vào tập T có đỉnh v đã được duyệt và đỉnh u đang duyệt. Vậy đồ thị đang được xây dựng không có chu trình.

Theo tính chất của phép duyệt theo chiều sâu, thủ tục CBTS thăm tất cả các đỉnh của đồ thị liên thông G.

Do vậy, đồ thị do thuật toán xây dựng sẽ cho ta một cây bao trùm của đồ thị đã cho.

Hiển nhiên, độ phức tạp của thật toán là $O(n+m)$.

Ví dụ 10.3. Áp dụng thuật toán trên cho đồ thị (nét mảnh) ta nhận được cây bao trùm (nét đậm) như sau.



Hình 10.5. Cây bao trùm của đồ thị tìm theo phương pháp duyệt sâu

Một cách tương tự, ta áp dụng phép duyệt đồ thị theo chiều rộng để tìm cây bao trùm của đồ thị liên thông và nhận được thuật toán sau đây.

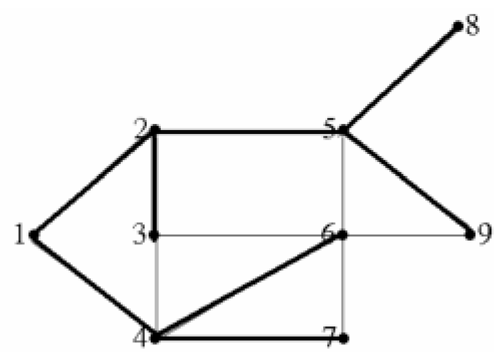
Thuật toán 10.2. Tìm cây bao trùm của đồ thị liên thông dùng phương pháp duyệt theo chiều rộng.

Đầu vào: Biểu diễn mảng DK các danh sách kề của đồ thị vô hướng G.
Đầu ra: Cây bao trùm (V, T) của đồ thị G.

```
procedure CBTR(v) ;
begin
    Q := ∅;
    enqueue v into Q;
    Duyet[v] := true;
    while Q ≠ ∅ do
    begin dequeue z from Q;
    for u ∈ DK[z] do
        if !Duyet[u] then
            begin enqueue u into Q ;
                Duyet[u] := true;
                T := T ∪ {(v,u)};
            end ;
      end ;
 end ;
 begin //Chương trình chính
   for u ∈ V do Duyet[u] := false;
   T := ∅;
   CBTR (z); //z là đỉnh tuỳ ý của đồ thị, sẽ trở thành gốc của cây
 end .
```

Hiển nhiên, độ phức tạp của thật toán là $O(n+m)$.

Ví dụ 10.4. Áp dụng thuật toán trên cho đồ thị (nét mảnh) ta nhận được cây bao trùm (nét đậm) như sau.



Hình 10.6. Cây bao trùm của đồ thị tìm theo phương pháp duyệt rộng

10.2.2. Một số ứng dụng của cây bao trùm

1) Kiểm tra tính liên thông của một đồ thị: Đồ thị là liên thông khi và chỉ khi nó có cây bao trùm.

2) Xây dựng hệ cơ sở của các chu trình.

Trước hết, giả thiết rằng đồ thị liên thông $G = (V, E)$ có n đỉnh và m cạnh. Trong trường hợp đồ thị không liên thông thì ta xét từng thành phần liên thông.

Để xây dựng hệ cơ sở các chu trình thuộc G ta tiến hành hai bước sau đây

1. Xây dựng cây bao trùm T của G.

   Giả sử trong quá trình xây dựng cây bao trùm T ta đã bỏ đi các cạnh: $e_1, e_2, ..., e_{m-n+1}$

2. Xây dựng hệ chu trình cơ sở:

   Lần lượt thêm vào cây T các cạnh $e_i (1 \leq i \leq m - n + 1)$, nghĩa là khôi phục lại cạnh e_i trong G, khi đó sẽ xuất hiện chu trình α_i - đây cũng là chu trình của đồ thị G. Sau đó lại xoá cạnh e_i và thêm cạnh e_{i+1} vào.

   Ta nhận được các chu trình tương ứng $\alpha_1, \alpha_2, ..., \alpha_{m-n+1}$.

   Hệ chu trình này độc lập vì:

   $\forall i \neq j$ thì α_i chứa e_i nhưng không chứa e_j , còn α_j chứa e_j nhưng không chứa e_i .

   Hơn nữa, số các chu trình này là $m - n + 1 = m - n + p =$ chu số của $G =$ số các chu trình độc lập cực đại.

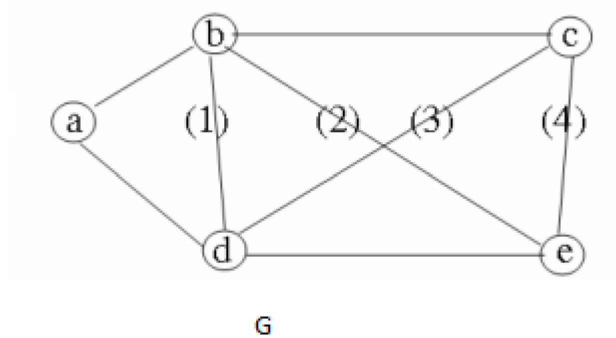
   Vậy hệ chu trình tìm được là một cơ sở của các chu trình trong đồ thị G.

Ví dụ 10.5. Xét đồ thị vô hướng sau đây $n = 5, m = 8, p = 1$. Vậy c(G)=4.

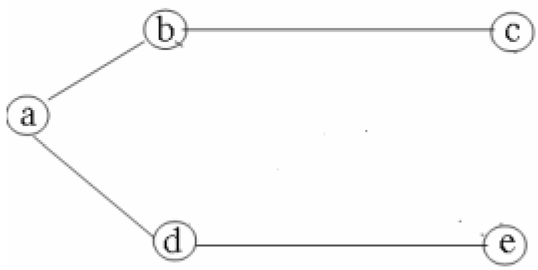
Một cây bao trùm T của G là

Ta nhận được một hệ chu trình cơ sở

$\alpha_1 = [a, b, d], \alpha_2 = [a, b, e, d], \alpha_3 = [a, b, c, d], \alpha_4 = [a, b, c, e, d]$.



Hình 10.7. Đồ thị và các cạnh bỏ đi



Hình 10.8. Một cây bao trùm của đồ thị trên

10.3 Cây bao trùm nhỏ nhất

Bây giờ ta xét bài toán tổng quát tìm cây bao trùm.

10.3.1. Bài toán cây bao trùm nhỏ nhất

Cho đồ thị vô hướng G với tập cạnh E và hàm trọng số $c : E \to N$. Hãy tìm cây bao trùm T của G sao cho tổng trọng số của các cạnh của T đạt giá trị nhỏ nhất.

Chẳng hạn như, xây dựng một hệ thống đường dây tải điện từ trạm phát điện đến các nơi tiêu thụ, nối các máy tính trong một mạng ... sao cho dây điện sử dụng là ít nhất.

10.3.2. Tìm cây bao trùm nhỏ nhất

Giả sử G là một đồ thị vô hướng liên thông và có trọng số. Khi đó, đồ thị G có cây bao trùm và sẽ có cây bao trùm nhỏ nhất.

Ta có thể dùng các thuật toán sau đây để tìm cây bao trùm nhỏ nhất của đồ thị G.

Thuật toán 10.3. Thuật toán Kruskal.

1) Chọn cạnh có trọng số bé nhất, ký hiệu là e_1 và đặt $W := \{e_1\}$.
2) Giả sử đã chọn được $W = \{e_1, e_2, ..., e_i\}$. Chọn e_{i+1} là cạnh có trọng số bé nhấttrong số các cạnh còn lại trong $E \setminus W$ sao cho $e_1, e_2, ..., e_i, e_{i+1}$ không chứa chutrình.
3) Đặt $W := W \cup \{e_{i+1}\}$.
   Lặp lại các bước 2) - 3) chừng nào còn có thể.

Tập cạnh W nhận được ở vòng lặp cuối cùng sẽ cho ta cây bao trùm nhỏ nhất.

Định lý 10.4. *Tập các cạnh W tìm được theo thuật toán Kruskal là cây bao trùm nhỏ nhất của đồ thị G.*

Chứng minh. Để đơn giản chứng minh, ta xét hai trường hợp sau đây:

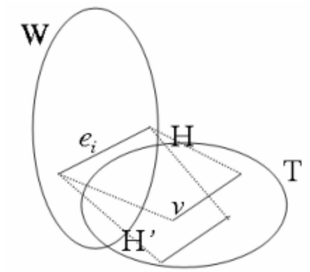
(i) Trước hết xét trường hợp G là đồ thị vô hướng đầy đủ có trọng số.

Tập cạnh W không có chu trình, nhưng nếu thêm một cạnh bất kỳ giữa hai đỉnh (cạnh này luôn có) sẽ tạo nên chu trình vì nếu không có thì quá trình lặp chưa kết thúc. Do đó theo tính chất 4) của cây, tập cạnh W là một cây.

Mặt khác, mỗi đỉnh của đồ thị G đều kề với tập cạnh W vì nếu không thì còn có thể thêm cạnh nữa vào W. Vậy trên W có n đỉnh và theo tính chất 3) của cây thì W có $n-1$ cạnh. Suy ra cây W là cây bao trùm của đồ thị G.

Bây giờ ta sẽ chứng minh rằng, W là cây bao trùm nhỏ nhất của đồ thị G. Giả sử T là một cây bao trùm nhỏ nhất nào đó của G. Ký hiệu e i là cạnh đầu tiên của W không thuộc T, vậy thì: $\{e_1, e_2, ..., e_{i-1}\} \subseteq T$ (Hình 10.9).

Theo tính chất 4) của cây, trong đồ thị $T \cup \{e_i\}$ có chu trình. Ký hiệu chu trình đó là H. Hiển nhiên, chu trình H phải chứa



Hình 10.9. Cách thay cạnh của T với W

cạnh e_i. H cũng không thể nằm trọn trong tập cạnh W vì trong W không có chu trình. Do vậy, có cạnh v trên chu trình này thuộc T nhưng không thuộc W.

Xét tập cạnh $T' = T \setminus \{v\} \cup \{e_i\}$.

- Tập cạnh T' không thể có chu trình. Vì nếu nó chứa chu trình H' nào đó thì H' phải chứa e_i và không chứa v. Khi đó, tập cạnh $(H' \setminus \{e_i\}) \cup (H \setminus \{e_i\})$ sẽ là một chu trình và chu trình này phải nằm trong cây T, trái với tính chất 2) của cây.

- Số cạnh của T' là $n - 1$. Vậy theo tính chất 2) của cây thì tập cạnh T' là một cây và là cây bao trùm của đồ thị G.

Hơn nữa, vì T là cây bao trùm nhỏ nhất nên: $c(e_i) \geq c(v)$. Cạnh v không thể tạo với tập cạnh W ở bước lặp $i - 1$ để có chu trình vì W nằm trong T. Nhưng nếu $c(e_i) > c(v)$ thì trong bước lặp i ta đã không chọn cạnh e_i. Vậy thì: $c(ei) = c(v)$ và trọng số của cây T bằng trọng số của cây T'. Ta nhận được cây bao trùm khác có trọng số không đổi nhưng có thêm một cạnh chung với W là e_i .

Tiếp tục quá trình này ta sẽ nhận được cây bao trùm có trọng số bằng trọng số của cây T và trùng với W. Suy ra, cây W cũng là cây bao trùm nhỏ nhất.

(ii) Bây giờ ta xét trường hợp G là đồ thị vô hướng liên thông có trọng số.

Ký hiệu: s là trọng số của đồ thị G.

Ta xây dựng đồ thị đầy đủ G' từ G như sau:

Nếu hai đỉnh trong G chưa kề nhau thì thêm một cạnh nối chúng với trọng số là $s+1$. Khi đó, nếu T là một cây bao trùm của G thì cũng là một cây bao trùm của G'.

Áp dụng thuật toán Kruskal cho đồ thị đầy đủ G' ta nhận được cây bao trùm nhỏ nhất W. Vậy thì, $s+1 >$ trọng số của $T \geq$ trọng số của W (vì W là cây bao trùm nhỏ nhất).

Vì trọng số của $W < s+1$, suy ra cây W chỉ chứa các cạnh trong G. Thế thì, thuật toán Kruskal áp dụng cho đồ thị G' đã không hề chọn các cạnh mới thêm vào mà chỉ chọn các cạnh thuộc G.

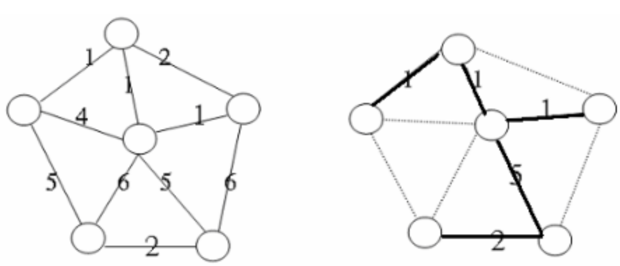
Vậy cây W cũng là cây bao trùm nhỏ nhất của G.

Thuật toán Kruskal được mô tả chi tiết như sau.

Thuật toán 10.4. Chi tiết thuật toán Kruskal.

```
procedure Kruskal;
 begin
   W := ∅; Z := E;
   while (|W| < n − 1)and(Z ≠ ∅) do
      begin
      chọn cạnh e có trọng số bé nhất trong Z;
      Z := Z \ {e};
      if W ∪ {e} không chứa chu trình then W := W ∪ {e}
      end;
   if |W| < n − 1 then writeln("Đồ thị không liên thông");
  end ;
```

Ví dụ 10.6. Đồ thị có trọng số và cây bao trùm nhỏ nhất của nó.



Hình 10.10. Đồ thị trọng số và một cây bao trùm nhỏ nhất

Prim đã cả tiến thuật toán Kruskal như sau: ở mỗi vòng lặp ta chọn cạnh có trọng số bé nhất trong số các cạnh kề với các cạnh đã chọn mà không tạo nên chu trình.

Thuật toán Prim được gọi là phương pháp lân cận gần nhất. Theo phương pháp này, bắt đầu từ một đỉnh nào đó a của đồ thị G ta nối nó với đỉnh "gần" nhất, chẳng hạn b. Nghĩa là, trong số các cạnh kề với a thì cạnh (a, b) được chọn có trọng số bé nhất. Tiếp theo, trong số các cạnh kề với đỉnh a hoặc đỉnh b ta chọn cạnh có trọng số bé nhất mà không tạo nên chu trình với cạnh (a, b). Cạnh này dẫn đến đỉnh thứ ba c ... Tiếp tục quá trình này cho đến khi nhận được cây gồm n đỉnh và $n - 1$ cạnh. Đó chính là cây bao trùm nhỏ nhất.

Thuật toán 10.5. Chi tiết thuật toán Prim.

```
procedure Prim;
 begin
    W := { cạnh có trọng số bé nhất };
    for  i := 1  to  n − 2  do
         begin
           e := cạnh có trọng số bé nhất kề với cạnh trong W và
                   nếu ghép nó vào W thì không tạo nên chu trình;
           W := W ∪ {e}
             end ;
end ;
```

Từ hai thuật toán đã trình bày ở trên, ta có thể khẳng định tính duy nhất của cây bao trùm nhỏ nhất trong trường hợp sau đây.

Định lý 10.5. *Nếu đồ thị vô hướng liên thông G có trọng số trên các cạnh khác nhau từng đôi thì cây bao trùm nhỏ nhất tồn tại và duy nhất.*

Chứng minh. Vì trong mỗi vòng lặp chỉ có duy nhất một cạnh được chọn.

10.4 Cây bao trùm lớn nhất

Trong các thuật toán Kruskal và Prim ta không ràng buộc về dấu của trọng số, nên có thể áp dụng cho đồ thị vô hướng với trọng số trên các cạnh có cùng dấu tuỳ ý.

Vì vậy để tìm cây bao trùm lớn nhất (tổng trọng số trên các cạnh của nó là lớn nhất) ta có hai cách sau đây:

1) Đổi thành dấu - cho các trọng số trên các cạnh. áp dụng một trong hai thuật toán đã trình bày ở trên để tìm cây bao trùm nhỏ nhất. Sau đó đổi dấu + trở lại, ta sẽ được cây bao trùm lớn nhất.

2) Sửa đổi trong các thuật toán: bước "chọn cạnh có trọng số bé nhất ... " được thay bằng "chọn cạnh có trọng số lớn nhất ... " còn các bước khác thì giữ nguyên. Khi thuật toán kết thúc, ta sẽ nhận được cây bao trùm lớn nhất.

10.5 Cây phân cấp

10.5.1. Định nghĩa cây phân cấp

Định nghĩa 10.3. *Cây phân cấp* là một cây, trong đó có một đỉnh đặc biệt gọi là gốc, giữa các đỉnh có mối quan hệ phân cấp "cha-con".

Số các con của một đỉnh trong cây phân cấp được gọi là *bậc* của đỉnh đó. Đỉnh không có con được gọi là *lá* của cây.

Thông thường, đỉnh không phải là lá được gọi là *đỉnh trong* của cây, còn lá được gọi là *đỉnh ngoài* của cây. Đỉnh gốc là đỉnh duy nhất không có cha.

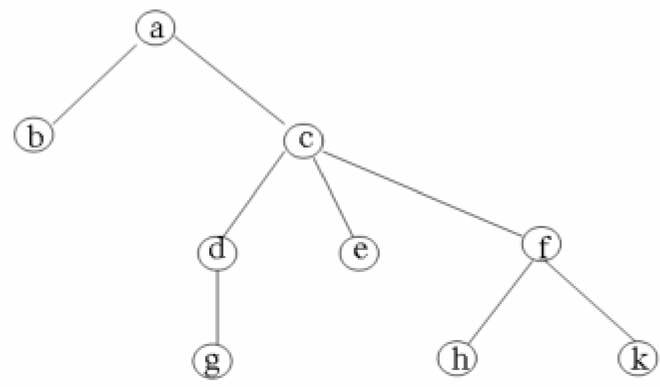
Ví dụ 10.7. Cây T dưới đây có đỉnh gốc a, các đỉnh lá b, g, e, h, k.

- *Mức* của đỉnh trong cây phân cấp:

Gốc của cây có mức là 0.

Nếu mức của đỉnh cha là i thì mức của các đỉnh con là $i + 1$.

- *Chiều cao* của cây là mức cao nhất của các đỉnh trong cây.



Hình 10.11. Cây phân cấp

Trong ví dụ trên, đỉnh gốc a có mức 0, các đỉnh b, c có mức 1, các đỉnh d, e, f có mức 2, các đỉnh g, h, k có mức 3. Cây có chiều cao là 3.

Cây phân cấp được áp dụng nhiều trong thực tế, chẳng hạn:

Mục lục của một cuốn sách để đọc giả tiện tra cứu.

Cấu trúc thư mục trên một ổ đĩa của máy tính để quản lý các tệp.

Sơ đồ tổ chức của một cơ quan để khách tiện liên hệ.

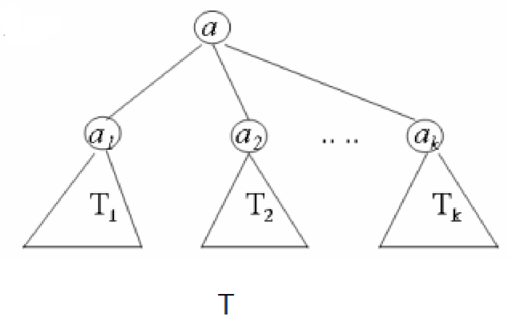
Để trình bày chặt chẽ các khái niệm khác và các phương pháp duyệt cây, ta đưa ra định nghĩa đệ quy cho cây phân cấp như sau.

Định nghĩa 10.4 (đệ quy). Tập rỗng là một cây phân cấp (cây rỗng).

Một đỉnh là một cây phân cấp.

Giả sử a là một đỉnh và $T_1, T_2, ..., T_k$ là các cây phân cấp với các gốc là $a_1, a_2, ..., a_k$ tương ứng. Cây T được xây dựng bằng cách cho đỉnh a làm "cha" của các đỉnh $a_1, a_2, ..., a_k$, sẽ là một cây phân cấp. Trong cây T này, đỉnh a là gốc và $T_1, T_2, ..., T_k$ là các cây con của gốc a.

- Đường đi trong cây phân cấp T là một dãy các đỉnh $< b_1, b_2, ..., b_m >$ mà b_i là "cha" của $b_{i+1}, 1 \leq i \leq m-1$.



Hình 10.12. Cây phân cấp tổng quát

Đường đi này đi từ đỉnh b_1 tới bm trong cây T. Như vậy, đường đi trong cây phân cấp chỉ đi từ đỉnh "tổ tiên" xuống các đỉnh "con cháu".

Định nghĩa 10.5. Cây phân cấp T với bậc cao nhất của các đỉnh trong T là m, được gọi là *cây m-phân*.

Định lý 10.6. *Giả sử T là một cây m-phân.*
Nếu cây T có chiều cao h thì cây có nhiều nhất m^h lá.
Nếu cây T có l lá thì cây có chiều cao $h \geq [\log_m l]$.

Chứng minh. 1) Chứng minh quy nạp theo chiều cao h.

$h = 1$: hiển nhiên.

$(h-1) \Rightarrow (h)$: Xét cây có chiều cao h. Bỏ gốc khỏi cây ta được một rừng gồm không quá m cây con, mỗi cây này có chiều cao $\leq h-1$. Theo giả thiết quy nạp thì mỗi cây con có nhiều nhất m^{h-1} lá. Mà lá của những cây con này cũng là lá của cây T. Vậy cây T có nhiều nhất là $m.m^{h-1} = m^h$ lá.

2) Vì số lá $l \leq mh$ suy ra $h \geq [\log_m l]$.

10.5.2. Các cách duyệt cây

Duyệt cây là cách đưa ra một danh sách tuyến tính liệt kê tất cả các đỉnh của cây, mỗi đỉnh một lần.

Ba cách duyệt cây hay được dùng là:

Duyệt theo thứ tự trước (pre-order search)

Duyệt theo thứ tự giữa (in-order search)

Duyệt theo thứ tự sau (post-order search)

Các cách duyệt cây này được định nghĩa đệ quy như sau.

Định nghĩa 10.6. - Nếu cây T rỗng thì cả ba cách duyệt đều cho danh sách rỗng.

- Nếu cây T chỉ có một đỉnh thì cả ba cách duyệt đều cho danh sách gồm chỉ một đỉnh này.

- Nếu cây T có gốc a và các cây con $T_1, T_2, ..., T_k$ thì

1. *Duyệt theo thứ tự trước* của cây T là danh sách bao gồm gốc a sau đó là các đỉnh của cây con T_1 được duyệt theo thứ tự trước, rồi đến các đỉnh của cây con T_2 được duyệt theo thứ tự trước .... cho đến các đỉnh của cây con T_k được duyệt theo thứ tự trước.
2. *Duyệt theo thứ tự giữa* của cây T là danh sách bao gồm các đỉnh của cây con T_1 được duyệt theo thứ tự giữa sau đó là gốc a, rồi đến các đỉnh của cây con T_2 được duyệt theo thứ tự giữa .... cho đến các đỉnh của cây con T_k được duyệt theo thứ tự giữa.
3. *Duyệt theo thứ tự sau* của cây T là danh sách bao 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 .... cho đến các đỉnh của cây con T_k được duyệt theo thứ tự sau và cuối cùng là đỉnh gốc a.

Ví dụ 10.8. Cây phân cấp T được cho như sau.

Duyệt theo thứ tự trước: 1, 2, 5, 6, 3, 4, 7, 9, 10, 8

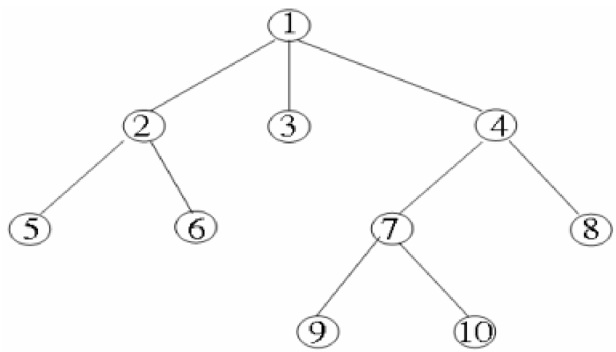
Duyệt theo thứ tự giữa: 5, 2, 6, 1, 3, 9, 7, 10, 4, 8

Duyệt theo thứ tự sau: 5, 6, 2, 3, 9, 10, 7, 8, 4, 1

Thủ tục duyệt một cây theo thứ tự giữa được mô tả như sau.

Thuật toán 10.6. Duyệt cây theo thứ tự giữa.

```
procedure DUYETCAYG( a ) ;
begin
```



Hình 10.13. Cây phân cấp và kết quả của 3 cách duyệt

if a là lá **then** write(a)
 else
 begin
  DUYETCAYG(con bên trái nhất của a) ;
  write(a);
  for mỗi một con c của a, trừ con
  bên trái nhất, từ trái sang phải **do**
  DUYETCAYG(c) ;
 end ;
end ;

10.6 Cây nhị phân

Cây nhị phân là cây phân cấp mà mỗi đỉnh của nó có không quá hai con. Các cây con của một đỉnh của cây nhị phân được phân biệt là cây con trái và cây con phải.

Định lý 10.7. *Số các cây nhị phân n đỉnh là* $c_n = \dfrac{1}{n+1}C_{2n}^{n}$.

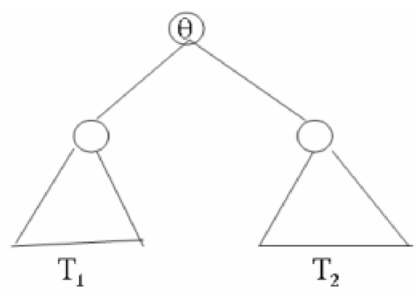
Dãy số $c_n, n = 0, 1, 2, \ldots$ do Catalan tìm ra và được gọi là dãy số Catalan.

Cây nhị phân có rất nhiều ứng dụng trong thực tế.

10.6.1. Cây biểu thức

Cây biểu thức là cây nhị phân mà mỗi đỉnh của nó được gán nhãn theo quy tắc sau: Các lá được gán các đại lượng còn các đỉnh trong được gán các dấu phép toán của một biểu thức nào đó.

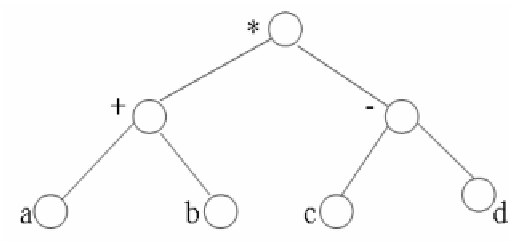
Cụ thể, nếu biểu thức $E = (E_1)\theta(E_2)$ và các cây biểu thức T_1, T_2 biểu diễn các biểu thức con E_1, E_2 thì cây biểu thức T biểu diễn E được xây dựng như sau



Hình 10.14. Cây biểu thức tổng quát

Ví dụ 10.9. Xét biểu thức sau đây $E = (a + b) * (c - d)$.

Cây biểu thức tương ứng Duyệt cây biểu thức trên theo thứ



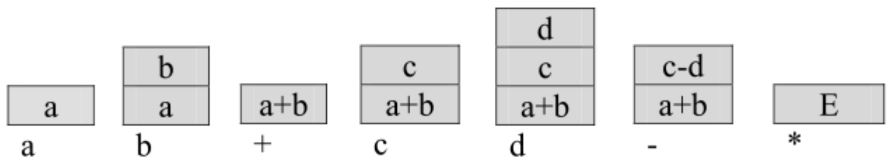
Hình 10.15. Cây biểu thức E

tự sau, ta được danh sách

$$a \quad b \quad + \quad c \quad d \quad - \quad *$$

Đó là dạng Ba lan ngược của biểu thức E, giúp máy tính tính giá trị của biểu thức E một cách thuận tiện nhờ một stack lưu giữ các đại lượng.

Mỗi khi đọc một phép toán thì máy thực hiện phép toán này với các đại lượng lấy ra từ đỉnh của stack, kết quả tính được lại đặt vào đỉnh của stack này. Cứ như thế cho đến khi thực hiện xong phép toán cuối cùng.



Hình 10.16. Dãy các stack phục vụ tính toán một biểu thức

10.6.2. Cây mã tiền tố

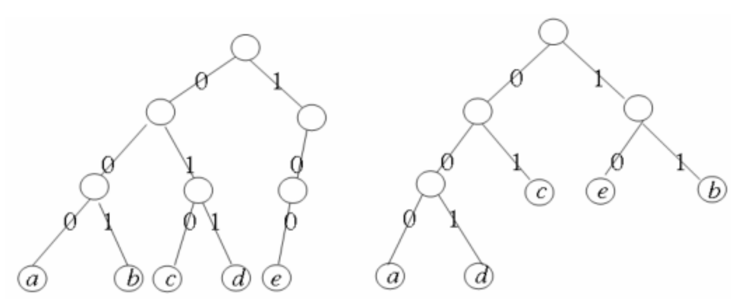
Cho một tập các ký hiệu. Hãy mã các ký hiệu này bằng dãy các chữ số 0, 1 thoả mãn tính chất tiền tố, nghĩa là không có mã của ký hiệu nào lại là tiền tố của mã của ký hiệu khác.

Để thực hiện công việc này ta xây dựng một cây nhị phân sao cho mỗi ký hiệu tương ứng với một lá, cạnh đi xuống con trái của một đỉnh được gán nhãn 0 còn cạnh đi xuống con phải được gán nhãn 1.

Khi đó, dãy các nhãn trên đường đi từ gốc đến lá sẽ cho mã tiền tố của ký hiệu tương ứng.

Cây nhị phân xây dựng như trên được gọi là *cây mã tiền tố.*

Ví dụ 10.10. Cây mã tiền tố



Hình 10.17. Các cây mã tiền tố

Các bộ mã tiền tố nhận được là

Ký hiệu	Bộ mã 1	Bộ mã 2
a	000	000
b	001	11
c	010	01
d	011	001
e	100	10

10.6.3. Cây mã Huffman

Giả sử ta có một bản tin là dãy các ký hiệu lấy trong một tập ký hiệu hữu hạn A. Biết rằng mỗi ký hiệu xuất hiện trong bản tin theo một tần suất đã biết.

Hãy xây dựng bộ mã tiền tố cho tập A sao cho độ dài chuỗi mã của bản tin là ngắn nhất.

Bộ mã tìm được mang tính tối ưu và được gọi là bộ mã Huffman.

Gọi d là số ký hiệu của bản tin, $\varphi(x)$ là tần suất xuất hiện của ký hiệu x trong bản tin. Mỗi cây nhị phân T với nhãn 0, 1 trên các cạnh và có số lá bằng số ký hiệu của tập A sẽ cho ta một bộ mã tiền tố cho tập ký hiệu A. Mức $\mu(x)$ của lá x chính là chiều dài mã của ký hiệu x. Khi đó, độ dài chuỗi mã của toàn bộ bản tin sẽ là

$$M = d \sum_{x \in A} \mu(x)\varphi(x).$$

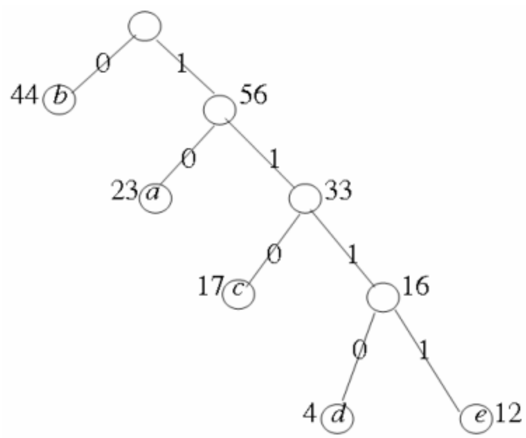
Cây mã tiền tố T là tối ưu khi độ dài M của mã bản tin đạt giá trị nhỏ nhất.

Ví dụ 10.11. Xét bản tin gồm 1000 ký hiệu trong tập ký hiệu $A = \{a, b, c, d, e\}$ với tần suất xuất hiện của các ký hiệu trong bản tin như sau

Ký hiệu	a	b	c	d	e
Tần suất %	23	44	17	4	12

Bộ mã 1 và bộ mã 2 chọn giống như trong Ví dụ 10.10. Bộ mã 3 được xây dựng dựa trên cây mã tiền tố (tối ưu) ở Hình 10.18. Độ dài của mã bản tin tương ứng như sau

Ký hiệu	Bộ mã 1	Bộ mã 2	Bộ mã 3
a	000	000	10
b	001	11	0
c	010	01	110
d	011	001	1110
e	100	10	1111
Độ dài mã bản tin	3000	2270	2050



Hình 10.18. Cây mã tiền tố tối ưu

Trong cây mã tiền tố ở hình vẽ trên, chúng ta đã xây dựng dựa trên nguyên lý: đỉnh lá nào có tần suất càng lớn thì đường đi từ gốc cây tới đỉnh lá đó càng ngắn.

Việc xây dựng bộ mã Huffman cho tập ký hiệu A được thực hiện thông qua xây dựng cây mã tiền tố tối ưu T. Thuật toán Huffman sẽ giúp chúng ta xây dựng cây này.

Giả sử tập A có l ký hiệu.

10.6.4. Thuật toán Huffman

Xây dựng rừng T có l cây, mỗi cây chỉ gồm một đỉnh tương ứng với một ký hiệu x trong A và được gán nhãn $\varphi(x)$.

Chọn hai cây trong T có gốc với nhãn nhỏ nhất. Thêm một đỉnh mới với nhãn là tổng các nhãn của hai gốc cây vừa chọn.

Nối đỉnh mới với hai gốc này bằng hai cạnh có nhãn 0, 1 để tạo thành một cây nhị phân.

Nếu T vẫn chưa phải là một cây thì lặp lại bước 2), ngược lại thì dừng. Cây mã tiền tố trong Hình 10.8 được xây dựng dựa trên thuật toán này.

Tính đúng đắn của Thuật toán Huffman được khẳng định qua định lý sau đây.

Định lý 10.8. *Khi thuật toán Huffman dừng thì cây mã tiền tố nhận được là tối ưu.*

Chứng minh. Ta chứng minh bằng quy nạp theo số ký hiệu l của tập A.

$l = 2$: Hiển nhiên.

$(l) \Rightarrow (l+1)$: Giả sử thuật toán đúng với các tập có l ký hiệu. Ta phải chứng minh nó cũng đúng với tập A có $l+1$ ký hiệu.

Giả sử tập $A = \{x_1, x_2, \ldots, x_{l-2}, x_{l-1}, x_l, x_{l+1}\}$. Không mất tính tổng quát, có thể giả thiết x_l, x_{l+1} có tần suất xuất hiện nhỏ nhất.

Ký hiệu H là cây mã tiền tố cho tập A theo thuật toán Huffman. Vì x_l, x_{l+1} có tần suất nhỏ nhất nên đã được chọn đầu tiên ở bước 2) và được thêm đỉnh mới y có tần xuất là $\varphi(xl) + \varphi(xl+1)$.

Hiển nhiên, theo cách xây dựng của thuật toán thì cây nhị phân $H' = H \setminus \{x_l, x_{l+1}\}$ là cây mã tiền tố của tập $A' = \{x_1, x_2, \ldots, x_{l-2}, x_{l-1}, y\}$ có l ký hiệu.

Theo giả thiết quy nạp, cây H' là cây mã tối ưu cho A'. Độ dài của mã bản tin theo cây H là

$$\begin{aligned} M &= d \sum_{x_i \in A} \mu(x_i)\varphi(x_i) \\ &= d\left[\sum_{i=1}^{l-1} \mu(x_i)\varphi(x_i) + \mu(y)\varphi(y)\right] + d(\varphi(x_l) + \varphi(x_{l+1})). \end{aligned}$$

đạt giá trị bé nhất. Cây mã tiền tố H là tối ưu.

Cây mã tiền tố tối ưu và thuật toán Huffman được ứng dụng rộng rãi trong mật mã học.

10.7 Bài tập

▷ **10.1.** Vẽ tất cả các cây (không đẳng cấu) có

a) 4 đỉnh b) 5 đỉnh c) 6 đỉnh

▷ **10.2.** Một cây có n_2 đỉnh bậc 2, n_3 đỉnh bậc 3, $\ldots$, n_k đỉnh bậc k. Hỏi có bao nhiêu đỉnh bậc 1?

▷ **10.3.** Tìm số tối đa các đỉnh của một cây m-phân có chiều cao h.

▷ **10.4.** Có thể tìm được một cây có 8 đỉnh và thoả điều kiện dưới đây hay không? Nếu có, vẽ cây đó ra, nếu không, giải thích tại sao

a) Mọi đỉnh đều có bậc 1.

b) Mọi đỉnh đều có bậc 2.

c) Có 6 đỉnh bậc 2 và 2 đỉnh bậc 1.

d) Có đỉnh bậc 7 và 7 đỉnh bậc 1.

▷ **10.5.** Chứng minh hoặc bác bỏ các mệnh đề sau đây.

a) Trong một cây, đỉnh nào cũng là đỉnh cắt.

b) Một cây có số đỉnh không nhỏ hơn 3 thì có nhiều đỉnh cắt hơn là cầu.

▷ **10.6.** Có bốn đội bóng đá A, B, C, D lọt vào vòng bán kết trong giải các đội mạnh khu vực. Có mấy dự đoán xếp hạng như sau

a) Đội B vô địch, đội D nhì.

b) Đội B nhì, đội C ba.

c) Đọi A nhì, đội C tư.

Biết rằng mỗi dự đoán trên đúng về một đội. Hãy cho biết kết quả xếp hạng của các đội.

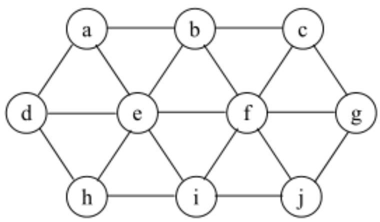
▷ **10.7.** Cây Fibonacci có gốc T_n được dịnh nghĩa bằng hồi quy như sau. T_1 và T_2 đều là cây có gốc chỉ gồm một đỉnh và với

$n = 3, 4, \ldots$ cây có gốc T n được xây dựng từ gốc với T_{n-1} như là cây con bên trái và T_{n-2} như là cây con bên phải.

a) Hãy vẽ 7 cây Fibonacci có gốc đầu tiên.

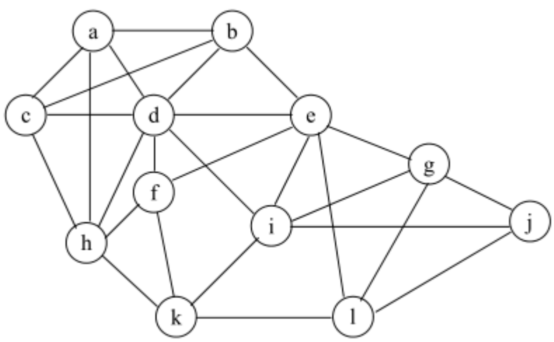
b) Cây Fibonacci T_n có bao nhiêu đỉnh, lá và bao nhiêu đỉnh trong. Chiều cao của nó bằng bao nhiêu?

▷ **10.8.** Hãy tìm cây bao trùm của đồ thị sau bằng cách xoá đi các cạnh trong các chu trình đơn.



Hình 10.19. Tìm cây bao trùm

▷ **10.9.** Hãy tìm cây bao trùm của đồ thị sau bằng cách xoá đi các cạnh trong các chu trình đơn.



Hình 10.20. Tìm cây bao trùm

▷ **10.10.** Hãy tìm cây bao trùm cho mỗi đồ thị sau.

a) K_5

b) $K_{4,4}$

c) $K_{1,6}$

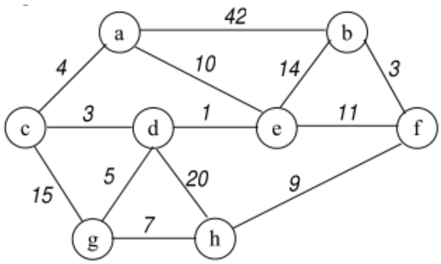
d) Q_3

e) C_5

f) W_5.

▷ **10.11.** Đồ thị K_n với$n = 3, 4, 5$ có bao nhiêu cây bao trùm không đẳng cấu?

▷ **10.12.** Tìm cây bao trùm nhỏ nhất của đồ thị sau theo thuật toán Kruskal và Prim.



Hình 10.21. Tìm cây bao trùm theo Kruskal và Prim

▷ **10.13.** Tìm cây khung nhỏ nhất bằng thuật toán Prim của đồ thị gồm các đỉnh A, B, C, D, E, F, H, I được cho bởi ma trận trọng số sau.

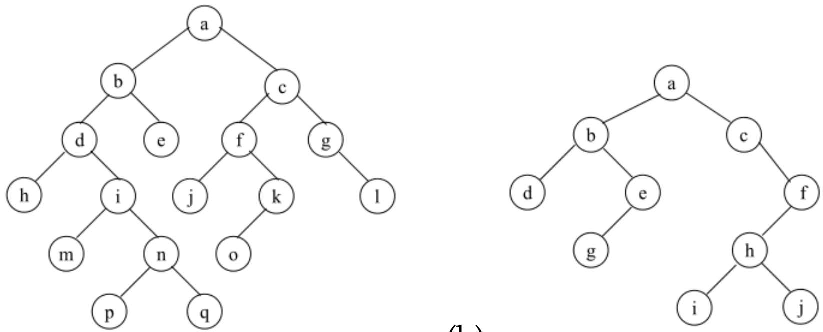
$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{array}
\begin{array}{c}
\begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\
\begin{pmatrix}
\infty & 16 & 15 & 23 & 19 & 18 & 32 & 20 \\
16 & \infty & 13 & 33 & 24 & 20 & 19 & 11 \\
15 & 13 & \infty & 13 & 29 & 21 & 20 & 19 \\
23 & 33 & 13 & \infty & 22 & 30 & 21 & 12 \\
19 & 24 & 29 & 22 & \infty & 34 & 23 & 21 \\
18 & 20 & 21 & 30 & 34 & \infty & 17 & 14 \\
32 & 19 & 20 & 21 & 23 & 17 & \infty & 18 \\
20 & 11 & 19 & 12 & 21 & 14 & 18 & \infty
\end{pmatrix}
\end{array}$$

Yêu cầu viết các kết quả trung gian trong từng bước lặp, kết quả cuối cùng cần đưa ra tập cạnh và độ dài của cây khung nhỏ nhất.

▷ **10.14.** Duyệt cây sau đây lần lượt bằng các thuật toán tiền thứ tự, trung thứ tự và hậu thứ tự.

▷ **10.15.** Viết các biểu thức sau đây theo ký pháp Ba Lan và ký pháp Ba Lan đảo.

a) $\dfrac{(A+B)(C+D)}{(A-B)C+D} + \dfrac{A^2+BD}{C^2-BD}$.



(a) (b)

Hình 10.22. Duyệt cây

b) $\left((a-b)^4 - \frac{c}{3} - 5d\right)^2 + (\frac{a-d}{3})^4 \frac{(3a+4b-2d)^3}{5}$.

TÀI LIỆU THAM KHẢO

[1] Đỗ Đức Giáo, Toán rời rạc ứng dụng trong tin học, NXBGD, 2008.

[2] Đỗ Đức Giáo, *Hướng dẫn giải bài tập toán rời rạc*, NXBGD, 2009.

[3] Nguyễn Đức Nghĩa, Nguyễn Tô Thành *Giáo trình toán rời rạc* NXB ĐHQGHN, 2009.

[4] Hoàng Chí Thành, *Lý thuyết đồ thị: Lý thuyết - Bài tập - Trắc nghiệm*, NXB ĐHQGHN, 2011.

[5] Kenneth H. Rosen, *Triển khai toán rời rạc với Maple*, NXB Giao thông Vận tải, 2008.

[6] Nguyễn Hữu Điển *Thực hành tính toán trong Maple*, NXB ĐHQG Hà Nội, 2015.

[7] N. H. Điển, *Một số vấn đề về thuật toán*, NXB GD, 2006.

[8] Tập thể Khoa Công nghệ - Thông tin, ĐH Huế, *Giáo trình Toán rời rạc*, ebook, 2003.

[9] Công nghệ thông tin, ĐH Hằng Hải, Hải Phòng *Bài giảng toán rời rạc*, ebook, 2010.

[10] Douglas B. West, *Introduction to graph theory*, Pearson Education (Singapore), 2001.

[11] C. Vasudev, *Graph theory with application*, New age internationa (P) limited, 2006.