

TOÁN RỜI RẠC






Back
Close











Back

Close



Chương 1

Lôgic mệnh đề

George Boole *Các định luật của tư duy* 1854.






Back
Close

Mệnh đề lôgic

Khái niệm mệnh đề và phủ định của mệnh đề

Định nghĩa 1.1. *Một mệnh đề (lôgic) là một khẳng định mà nội dung của nó là đúng hoặc là sai, chứ không thể vừa đúng vừa sai.*

Ví dụ.

1. Mưa bay, gió cuốn.
2. Cuốn sách này là của ai vậy?
3. $x + 3 = 7$.
4. Hà nội là thủ đô của Việt nam.
5. Tổng các góc của một tam giác bằng 100°.
6. $4 + 4 = 7$.

Giá trị chân lý của một một mệnh đề lôgic

Giá trị chân lý của mệnh đề lôgic là T (true) hoặc F (false).

Ví dụ.

1. p: "Hà nội là thủ đô của Việt nam."
2. q: "Tổng các góc của một tam giác bằng 100°."
3. r: "$4 + 4 = 7$."

Bảng 1.1: Bảng giá trị chân lý.

p	q	r
F	F	F

Mệnh đề phức hợp

Ví dụ.

1. Nếu x là số nguyên, thì x^2 cũng là số nguyên.
2. Trời vừa nắng vừa mưa.
3. Biển không phải là ao hồ.
4. Để được đi học nước ngoài, hoặc là bạn phải học giỏi hoặc là bạn phải có tiền tự túc.

Tính chất. Liên từ liên kết các mệnh đề đơn giản tạo nên mệnh đề phức hợp:

Ví dụ. "*Bạn* **không được** *đi xe máy,* **nếu** *bạn dưới 16 tuổi* **trừ phi** *đó là xe phân khối nhỏ* **hoặc** *khi bạn có giấy phép đặc biệt.*

Phủ định mệnh đề

Định nghĩa 1.2. *Cho trước mệnh đề lôgic p. Khi đó câu "không phải là p" cũng là một mệnh đề lôgic, được gọi là phủ định của p và được ký hiệu là $\bar{p}$ hoặc là $\neg p$. Nếu p đúng thì $\bar{p}$ sai và ngược lại.*

Ví dụ.

p: Ngày 20-11-2008 là ngày chủ nhật.

$\bar{p}$: Ngày 20-11-2008 không phải là ngày chủ nhật.

Bảng 1.2: Bảng giá trị chân lý mệnh đề phủ định

p	$\bar{p}$
T	F
F	T

Phép hội

Định nghĩa 1.3. *Cho trước hai mệnh đề lôgic p và q. Khi đó câu nói "p và q" cũng là một mệnh đề lôgic, ký hiệu $p \wedge q$. Hội của p và q chỉ đúng khi cả hai mệnh đề p và q đều đúng và sai trong các trường hợp còn lại.*

Ví dụ.

p: Bác Hồ sinh vào ngày 19-5.

q: Bác Hồ là Chủ tịch nước.

$p \wedge q$: Bác Hồ sinh vào ngày 19-5 và Bác Hồ là Chủ tịch nước.

Bảng 1.3: Bảng giá trị chân lý của phép hội

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F











Phép tuyển

Định nghĩa 1.4. *Cho trước hai mệnh đề lôgic p và q. Khi đó câu nói "p hoặc q" cũng là một mệnh đề lôgic và được ký hiệu là $p \vee q$. Tuyển của p và q chỉ sai khi cả p và q cùng sai và đúng trong các trường hợp còn lại.*

Ví dụ.

p: Hồ Xuân Hương sinh vào ngày 3-5.

q: Hồ Xuân Hương sinh vào ngày 9-5.

$p \vee q$: Hồ Xuân Hương sinh vào ngày 3-5 hoặc vào ngày 9-5.

Bảng 1.4: Bảng giá trị chân lý của phép tuyển

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F



Phép tuyển có loại

Định nghĩa 1.5. *Cho trước hai mệnh đề lôgic p và q. Khi đó câu nói "hoặc p hoặc q" cũng là một mệnh đề lôgic và được gọi là tuyển có loại của p và q và được ký hiệu là $p \oplus q$. Tuyển có loại của p và q chỉ đúng khi chỉ có đúng một trong p và q là đúng còn mệnh đề còn lại sai.*

Ví dụ.

p: Hồ Xuân Hương sinh vào ngày 3-5.

q: Hồ Xuân Hương sinh vào ngày 9-5.

$p \oplus q$: Hồ Xuân Hương hoặc sinh vào ngày 3-5 hoặc vào ngày 9-5.

Bảng 1.5: Bảng giá trị chân lý của phép tuyển có loại

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F



Phép kéo theo

Định nghĩa 1.6. *Cho trước hai mệnh đề lôgic p và q. Khi đó câu nói "nếu có p thì có q" cũng là một mệnh đề lôgic và được gọi là phép kéo theo của p và q và được ký hiệu là $p \to q$. Mệnh đề $p \to q$ chỉ sai nếu p đúng và q sai và đúng trong các trường hợp còn lại.*

Theo định nghĩa của ta, toán tử $\to$ có tên là phép kéo theo. Trong phép kéo theo pq, ta thường nói p là giả thiết và q là kết luận.

Ví dụ. Nếu gọi p là mệnh đề "*Tam giác ABC vuông tại đỉnh A*" và q là mệnh đề "$BC^2 = CA^2 + AB^2$" thì mệnh đề $p \to q$ là "*Nếu tam giác ABC vuông tại đỉnh A thì $BC^2 = CA^2 + AB^2$*".

Tính chất. Khi có mệnh đề kéo theo $p \to q$, thì mệnh đề $q \to p$ được gọi là mệnh đề đảo của $p \to q$ và mệnh đề $\bar{q} \to \bar{p}$ được gọi là mệnh đề phản đảo của nó.

Các cách phát biểu khác của phép kéo theo:

1. nếu p thì q,
2. p kéo theo q,
3. p là điều kiện đủ của q,



Bảng 1.6: Bảng giá trị chân lý của phép kéo theo

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

4. q là điều kiện cần của p,

5. có p thì có q,

6. từ p suy ra q, ...



Phép tương đương

1. p và q tương đương với nhau,
2. p nếu và chỉ nếu q,
3. p khi và chỉ khi q,
4. p tương đương với q, ...

Định nghĩa 1.7. *Cho trước hai mệnh đề lôgic p và q. Khi đó câu nói "p tương đương với q" cũng là một mệnh đề lôgic. Ta ký hiệu mệnh đề "p tương đương với q" bởi ký hiệu $p \leftrightarrow q$. Mệnh đề $p \leftrightarrow q$ chỉ đúng khi p và q cùng đúng hoặc cùng sai.*

Ví dụ.

p: Tam giác ABC là tam giác đều.

q: Tam giác ABC có ba cạnh bằng nhau.

$p \leftrightarrow q$: Tam giác ABC là tam giác đều khi và chỉ khi tam giác ABC có ba cạnh bằng nhau.

Bảng 1.7: Bảng giá trị chân lý của phép tương đương

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Phân tích mệnh đề lôgic phức hợp

Ví dụ.

Bạn không được đi xe máy, nếu bạn dưới 16 tuổi trừ phi đó là xe phân khối nhỏ hoặc khi bạn có giấy phép đặc biệt

1. p: *Bạn được đi xe máy*,
2. q: *Bạn dưới 16 tuổi*,
3. r: *Xe máy có phân khối nhỏ*,
4. s: *Bạn có giấy phép đặc biệt.*

Kết quả: $(q \wedge \neg(r \vee s)) \rightarrow \neg p$.

Biểu thức lôgic

Định nghĩa 1.8. *Một biểu thức lôgic là một biểu thức được tạo thành từ các biến lôgic cho trước bằng cách áp dụng các toán tử lôgic và các dấu ngoặc "(" và ")"một cách hình thức. Ta quy định toán tử* $\neg$ *được ưu tiên thực hiện trước, tiếp đó theo thứ tự là phép toán* $\wedge$, $\vee$, $p \to q$ *và* $p \leftrightarrow q$, *và từ trái sang phải cho các phép toán cùng ưu tiên, và đặc biệt nếu có ngoặc thì bắt đầu thực hiện từ dấu ngoặc trong cùng ra ngoài.*

Ví dụ. Trong biểu thức lôgic $((p \vee q) \to \bar{r}) \wedge p$ thì ta phải thực hiện phép phủ định r, sau đó thực hiện phép tuyển $p \vee q$, rồi tiếp đó là phép kéo theo $((p \vee q) \to \bar{r})$. Sau cùng ta thực hiện phép hội $\wedge$.








Một số biểu thức lôgic quan trọng

Định nghĩa 1.9. *Cho trước mệnh đề lôgic $p \to q$. Khi đó mệnh đề lôgic$q \to p$được gọi là mệnh đề lôgic đảo và mệnh đề lôgic $\neg q \to \neg p$ được gọi là mệnh đề lôgic phản đảo của mệnh đề lôgic $p \to q$ đã cho.*

Ví dụ.

p: Tam giác vuông có một góc nhọn 30°.

q: Tam giác có hai cạnh dài gấp đôi nhau.

$p \to q$: Nếu tam giác vuông có một góc nhọn 30°, thì nó có hai cạnh dài gấp đôi nhau.

$q \to p$: Nếu một tam giác có hai cạnh dài gấp đôi nhau, thì nó là tam giác vuông với một góc nhọn 30°.

$\neq q \to \neq p$: Nếu tam giác không có hai cạnh dài gấp đôi nhau, thì nó không phải là tam giác vuông với một góc nhọn 30°.











Bảng 1.8: Bảng giá trị chân lý của mệnh đề phản đảo và đảo.

p	q	$\bar{p}$	$\bar{q}$	$p \to q$	$\neg q \to \neg p$	$q \to p$
T	T	F	F	T	T	T
T	F	F	T	F	F	T
F	T	T	F	T	T	F
F	F	T	T	T	T	T

Tương đương lôgic

Định nghĩa 1.10. *Một biểu thức lôgic luôn có giá trị chân lý T (đúng) với bất cứ giá trị chân lý nào của các mệnh đề lôgic thành phần tạo nên nó được gọi là mệnh đề lôgic luôn đúng hoặc là hằng đúng. Ký hiệu $\mathbb{T}$.*

Một biểu thức lôgic luôn có giá trị chân lý F (sai) với bất cứ giá trị chân lý nào của các mệnh đề lôgic thành phần tạo nên nó được gọi là biểu thức lôgic luôn sai hoặc là hằng sai, ký hiệu $\mathbb{F}$.

Biểu thức lôgic không phải hằng đúng hoặc không hằng sai được gọi là tiếp liên.

Ví dụ.



$p \vee \neg p$ là $\mathbb{T}$.
$p \wedge \neg p$ là $\mathbb{F}$.
$p \to q$ là tiếp liên.

Bảng 1.9: Bảng giá trị chân lý của $p \vee \bar{p}$ và $p \wedge \neg p$

p	$p \vee \neg p$	$p \wedge \neg p$
T	T	F
F	T	F

Định nghĩa 1.11. *Các mệnh đề lôgic p và q được gọi là tương đương lôgic, nếu biểu thức lôgic $p \leftrightarrow q$ là mệnh đề lôgic hằng đúng. Khi ấy ta nóip và q là hai mệnh đề lôgic tương đương (bằng nhau) và ký hiệu $p \Leftrightarrow q$.*

Ví dụ. Ta có thể kiểm tra xem mệnh đề kép theo $p \to q$ và mệnh đề phản đảo của nó $\neg q \to \neg p$ có tương đương lôgic hay không thông qua bảng 1.10:

Tính chất. Dễ kiểm tra thấy rằng quan hệ tương đương lôgic là quan hệ có 3 tính chất: phản xạ, đối xứng và bắc cầu.

Bảng 1.10: Tương đương lôgic của phép kéo theo và phản đảo.

p	q	$p \to q$	$\neg q \to \neg p$	$(p \to q) \leftrightarrow (\neg q \to \neg p)$
T	T	T	T	T
T	F	F	F	T
F	T	T	T	T
F	F	T	T	T

1. Với mọi mệnh đề p ta luôn có $p \Leftrightarrow p$ (tính phản xạ).

2. Nếu $p \Leftrightarrow q$ thì $q \Leftrightarrow p$ (tính đối xứng).

3. Nếu $p \Leftrightarrow q$ và $q \Leftrightarrow r$ thì $p \Leftrightarrow r$ (tính bắc cầu).

Ngoài ra, nhờ sự tương đương lôgic của phép kéo theo $p \to q$ với biểu thức $\neg p \vee q$ (xem bảng 1.11) mà chúng ta có thể khử các phép kéo theo trong các biểu thức lôgic để từ một biểu thức lôgic cho trước ta thu được một biểu thức lôgic hoàn toàn không có phép kéo theo và phép tương đương.

Ví dụ. Chứng minh rằng $(\mathrm{p} \wedge \mathrm{q}) \to (\mathrm{p} \vee \mathrm{q})$ là hằng đúng.



Bảng 1.11: Khử phép kéo theo.

p	q	$p \to q$	$\neg p \vee \neg q$	$(p \to q) \leftrightarrow (\neg p \vee q)$
T	T	T	T	T
T	F	F	F	T
F	T	T	T	T
F	F	T	T	T

Chứng minh. Ta có

$$\begin{aligned}(p \wedge q) \to (p \vee q) &\Leftrightarrow (\neg p \vee q) \to (p \vee q)\\ &\Leftrightarrow \neg p \vee q \vee p \vee q\\ &\Leftrightarrow (\neg p \vee p) \vee (q \vee q)\\ &\Leftrightarrow \mathbb{T} \vee q\\ &\Leftrightarrow \mathbb{T}.\end{aligned}$$

1. $p \wedge \mathbb{T} \Leftrightarrow p$ (luật đồng nhất),
2. $p \vee \mathbb{F} \Leftrightarrow p$ (luật đồng nhất),

3. $p \wedge \mathbb{F} \Leftrightarrow \mathbb{F}$ (luật nuốt),
4. $p \vee \mathbb{T} \Leftrightarrow \mathbb{T}$ (luật nuốt),
5. $p \vee p \Leftrightarrow p$ (luật lũy đẳng),
6. $p \wedge p \Leftrightarrow p$ (luật lũy đẳng),
7. $\neg(\neg p) \Leftrightarrow p$ (luật phủ định kép),
8. $p \vee q \Leftrightarrow q \vee p$ (luật giao hoán),
9. $p \wedge q \Leftrightarrow q \wedge p$ (luật giao hoán),
10. $(p \vee q) \vee r \Leftrightarrow p \wedge (q \wedge r)$ (luật kết hợp),
11. $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$ (luật kết hợp),
12. $(p \vee q) \wedge r \Leftrightarrow (p \wedge r) \vee (q \wedge r)$ (luật phân phối),
13. $(p \wedge q) \vee r \Leftrightarrow (p \vee r) \wedge (q \vee r)$ (luật phân phối),
14. $\neg(p \vee q) \Leftrightarrow (\neg p \wedge \neg q)$ (luật De Morgan),
15. $\neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$ (luật De Morgan).

Ví dụ. $\neg(p \wedge q \wedge r) \Leftrightarrow \neg((p \wedge q) \wedge r) \Leftrightarrow \neg(p \wedge q) \vee \neg r \Leftrightarrow \neg p \vee \neg q \vee \neg r.$





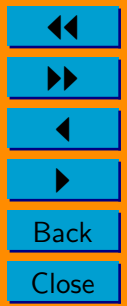


Các phép toán lôgic với các bit

Phép toán bit OR, AND, XOR

Máy tính dùng các bit để biểu diễn thông tin. Một bit có giá trị 0 hoặc 1. Nguồn gốc của cách gọi tên bit là do nhà thống kê học nổi tiếng người Anh John Tukey đưa ra năm 1946 với nguyên bản là binary digit (chữ số nhị phân). Do giá trị chân lý của các mệnh đề lôgic cũng chỉ là hai giá trị T và F, cho nên bit cũng có thể dùng để biểu diễn giá trị chân lý của các mệnh đề lôgic, trong đó giá trị 1 ứng với T và giá trị 0 ứng với F. Với cùng lý do đó mà bit cũng được dùng để biểu diễn các biến Boole (boolean variable) là các biến mà giá trị của nó là đúng hoặc sai.

T rong lập trình, chúng ta hay dùng các phép toán OR, AND, XOR. Đây là các phép toán lôgic tương ứng với các toán tử lôgic . Phép toán bit được tiến hành trên các số 0 và 1 tương tự như các phép toán với các giá trị chân lý T và F bằng cách thay T bởi 1 và F bởi giá trị 0. Để đơn giản, người ta cũng dùng các lôgic toán tử cho các phép toán OR, AND và XOR một cách tương ứng. Trong bảng 1.12, chúng ta có công thức thực



Bảng 1.12: Bảng tính cho các toán tử OR, AND, và XOR.

	OR	
$\vee$	0	1
0	0	1
1	1	1

	AND	
$\wedge$	0	1
0	0	0
1	0	1

	XOR	
$\oplus$	0	1
0	0	1
1	1	0

hiện phép toán bit với các bit 0 và 1.



Phép toán OR-bit, AND-bit, XOR-bit

Định nghĩa 1.12. *Thông tin trong máy tính thông thường được biểu diễn dưới dạng một dãy các bit (ta còn gọi là xâu bit hoặc là xâu nhị phân). Một xâu rỗng là một xâu không có bit nào cả. Số các bit tạo nên xâu nhị phân được gọi là độ dài của xâu nhị phân.*

Ví dụ. Xâu rỗng có đội dài 0, và 01110101 là một xâu nhị phân với chiều dài là 8.

Người ta mở rộng phép tính OR, AND và XOR cho các xâu bit có cùng độ dài. Các phép toán mở rộng này được gọi tên một cách tương ứng là các phép toán OR-bit, AND-bit và XOR-bit. Cách thực hiện các phép toán này với hai xâu bit cùng độ dài là áp dụng các phép toán OR, AND và XOR cho các bit tương ứng ở hai xâu. Để đơn giản, người ta cũng dùng ký hiệu lôgic toán tử để biểu diễn các phép toán OR-bit, AND-bit và XOR-bit. Ví dụ sau giải thích rõ cách thực hiện các phép toán này.

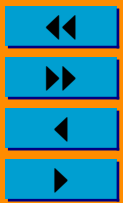
Ví dụ. Với hai xâu 1001 và 0111 ta có:

1. $1001 \vee 0111 = 1111$ (phép toán OR-bit),
2. $1001 \wedge 0111 = 0001$ (phép toán AND-bit),
3. $1001 \oplus 0111 = 1110$ (phép toán XOR-bit).



Chương 2

Lý thuyết tập hợp



Back
Close

Khái niệm tập hợp

Cantor (1845-1918): Một *tập hợp* là một tổng thể các đối tượng (được gọi là các phần tử của tập hợp) có cùng một tính chất chung nào đó.

Ví dụ. Tập hợp các sinh viên trong lớp học.

Tập hợp rỗng $\emptyset$.

Tập hợp các số tự nhiên $\mathbb{N}$

Tập hợp các số hữu tỷ $\mathbb{Q}$

Tập hợp các số thực $\mathbb{R}$.

Tập hợp các số nguyên $\mathbb{Z}$.

Tập hợp các số phức $\mathbb{C}$.

Phần tử của tập hợp: a, b, ...

Ví dụ. $1 \in \mathbb{Z}$, $\frac{1}{2} \notin \mathbb{Z}$, $\sqrt{2} \notin \mathbb{Z}$.

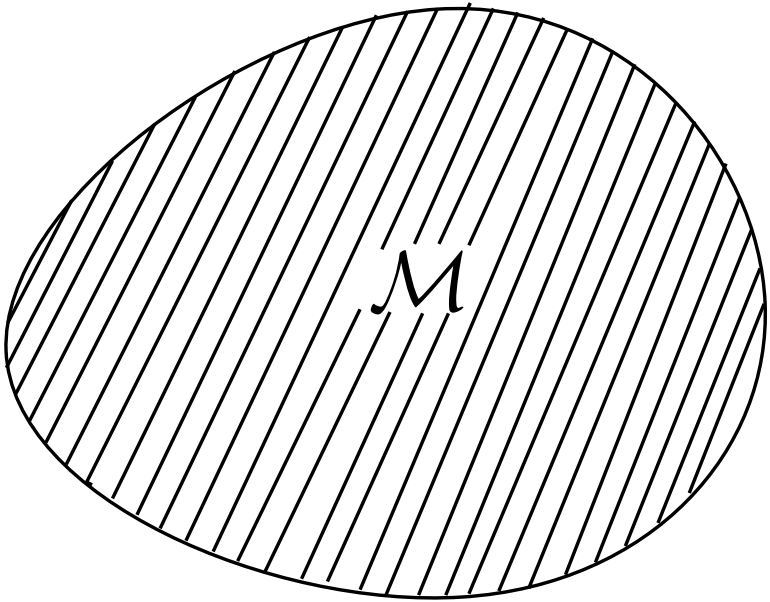












Hình 2.1: Biểu đồ Venn biểu diễn tập hợp $\mathcal{M}$.

Có ba cách biểu diễn tập hợp như sau:

a) **Biểu diễn tập hợp bằng liệt kê tất cả các phần tử của nó:**

$A = \{1, 2, 3\}$
$B = \{1, 2, 3, 4, 5, 6\}$.
b) **Biểu diễn tập hợp thông qua quy luật đơn giản.**

$A = \{0, 2, 4, \ldots.\}$
$B = \{1, 3, 5 \ldots\}$
$\mathbb{Z} = \{0, -1, 1, -2, 2 \ldots\}$
$\mathbb{Z} = \{\ldots - 2, -1, 0, 1, -2 \ldots\}$
$\mathbb{N} = \{0, 1, 2, \ldots\}$.
c) **Biểu diễn tập hợp thông qua quy tắc nhận biết.**

$$\mathcal{P} = \{p \,|p \text{ là số nguyên tố}\},$$
$$X = \{x \ : x^2 - 2x + 1 = 0\}.$$



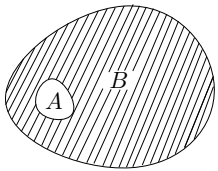





Tập hợp con và tập hợp bằng nhau

Tập hợp con

Định nghĩa 2.1. *Cho trước hai tập hợp A và B. Ta nói rằng tập hợp A là tập hợp con của tập hợp B, nếu như mỗi phần tử của tập hợp A là phần tử của tập hợp B, khi đó ta viết $A \subseteq B$.*



Hình 2.2: Tập hợp chứa nhau.

Trong trường hợp tập hợp A là tập hợp con của tập hợp B, ta còn nói rằng tập hợp B chứa tập hợp A. Trong trường hợp tập hợp A là tập hợp con của tập hợp B và tập hợp B không phải tập hợp con của tập hợp A thì ta nói rằng A là tập hợp con thật sự của tập hợp B, và ta viết $A \subset B$.

Biểu đồ Venn của tập hợp con bị chứa trong biểu đồ Venn của tập hợp chứa nó (Hình 2.2).

Ví dụ. Tập hợp các số tự nhiên $\mathbb{N}$ là tập hợp con thật sự của tập hợp các số nguyên $\mathbb{Z}$. Tập hợp các số thực $\mathbb{R}$ là tập hợp con thật sự của tập hợp

các số phức $\mathbb{C}$. Tập hợp các số hữu tỷ $\mathbb{Q}$ là tập hợp con thật sự của tập hợp các số thực $\mathbb{R}$. Quan hệ giữa các tập hợp này có thể viết gọn lại là $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$.

Với mỗi n, tập hợp số tự nhiên $\mathbb{N}$ có một tập hợp con đặc biệt gồm các số tự nhiên đầu tiên từ 0 tới n, được ký hiệu là $\overline{1, n}$.

Tập hợp $\emptyset$ được qui định là tập hợp con của tất cả các tập hợp. Mỗi tập hợp bất kỳ cũng là tập hợp con của chính nó. Cho trước tập hợp A, ta ký hiệu tập hợp tất cả các tập hợp con của A là $\mathcal{P}(A)$. Tập hợp các tập hợp con của tập hợp A cho trước luôn chứa tập hợp $\emptyset$ và A làm phần tử.

Ví dụ. Với $A = \{1, 2\}$ thì $\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$.

Tính chất. Quan hệ "chứa nhau" ($\subseteq$") của tập hợp là một quan hệ có tính chất phản xạ và bắc cầu:

1. Với mọi tập hợp A ta có $A \subseteq A$.
2. Nếu $A \subseteq B$ và $B \subseteq C$ thì $A \subseteq C$.

Tập hợp bằng nhau

Trên cở sở khái niệm các tập hợp con, ta định nghĩa khái niệm sự bằng nhau của các tập hợp như sau.

Định nghĩa 2.2. *Cho trước hai tập hợp A và B. Ta nói tập hợp A và B là hai tập hợp bằng nhau khi và chỉ khi A là tập hợp con của tập hợp B và B là tập hợp con của tập hợp A. Khi A và B là hai tập hợp bằng nhau thì ta viết $A = B$.*

Ví dụ. Tập hợp $A = \{1, 2\}$ và tập hợp $B = \{x \mid x^2 - 3x + 2 = 0\}$ là hai tập hợp bằng nhau, vì phương trình $x^2 - 3x + 2 = 0$ có đúng hai nghiệm 1 và 2.

Tính chất. Quan hệ "chứa nhau" của tập hợp có những tính chất sau:
1. Nếu $A \subseteq B$ và $B \subseteq A$ thì $A = B$.

Quan hệ "bằng nhau" của tập hợp là quan hệ tương đương:
1. Với mọi tập hợp ta có $A = A$ (tính phản xạ).
2. Nếu $A = B$ thì $B = A$ (tính đối xứng).
3. Nếu $A = B$ và $B = C$ thì $A = C$ (tính bắc cầu).

Bài tập:

1. Kiểm tra xem các mối quan hệ sau là đúng hay sai:
   a) $\emptyset = \{\emptyset\}$,
   b) $\{1\} \subset \{\{1\}\}$,
   c) $\{1\} = \{\{1\}\}$.

2. Dùng biểu đồ Venn biểu diễn mối quan hệ của các tập hợp $A \subseteq B$ và $B \subseteq C$.

3. Chứng minh tính chất của quan hệ "$\subseteq$" trong các tập hợp:
   1. Với mọi tập hợp A ta có $A \subseteq A$.
   2. Nếu $A \subseteq B$ và $B \subseteq A$thì $A = B$.
   3. Nếu$A \subseteq B$ và $B \subseteq C$ thì $A \subseteq C$.

4. Cho trước tập hợp $A = \{1, 2, 3\}$ và tập hợp $B = \{1, 3, 5, 7\}$. Hãy liệt kê tất cả các tập hợp vừa là tập con của tập hợp A vừa là tập hợp con của tập hợp B.

5. Tìm hai tập hợp A và B sao cho $A \in B$ và $A \subseteq B$.

6. Xác định mối quan hệ giữa các tập hợp và sau:
   a) $A = \{1, 2, 3\}$ và $B = \{1, 3, 5, 7\}$.
   b) $A = \{1, 2, 3\}$ và $B = \{1, 3, 5, 2, 7\}$.

7. Xác định xem các cặp tập hợp sau đây có bằng nhau hay không:
   a) Tập hợp $\{1, 2, 3\}$ và tập hợp $\{3, 2, 1\}$.
   b) Tập hợp tất cả các tập hợp con của $\{a, b\}$ và $\{\emptyset, \{a\}, \{b\}, \{a, b\}$.



c) Tập hợp $\emptyset$ và tập hợp $\{\emptyset\}$.
d) Tập hợp $\mathbb{N}$ và tập hợp $\{\mathbb{N}, 0, 1, 2, \ldots\}$.

8. Xác định tập hợp $\mathcal{P}(\emptyset)$ và tập hợp $\mathcal{P}(\{\emptyset\})$.

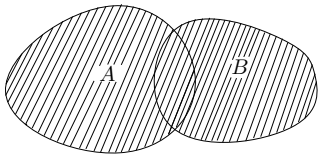
9. Xác định tập hợp $\mathcal{P}\{1, 2, 3\}$.

10. Chứng minh các tính chất của quan hệ tập hợp bằng nhau:
    1. Với mọi tập hợp ta có $A = A$.
    2. Nếu $A = B$ thì $B = A$.
    3. Nếu $A = B$ và $B = C$ thì $A = C$.

11. Chứng minh rằng nếu $A = B$ thì $\mathcal{P}(A) = \mathcal{P}(B)$.

Các phép toán của tập hợp

Phép hợp



Hình 2.3: Phép hợp.

Định nghĩa 2.3. *Cho trước tập hợp A và tập hợp B. Hợp của tập hợp A và tập hợp B là tập hợp chứa tất cả các phần tử thuộc A hoặc thuộc B, và chỉ những phần tử đó mà thôi.*

Hợp của tập hợp A và tập hợp B được ký hiệu bởi $A \cup B$. Một cách hình thức ta viết: $A \cup B = \{x \mid x \in A \text{ hoặc } x \in B\}$.

Hợp của hai tập hợp được biểu diễn theo mô hinh trong biểu đồ Venn (hình 2.3), .

Trong trường hợp tập hợp A và tập hợp B có những phần tử chung thì các phần tử chung này chỉ được phép xuất hiện đúng một lần trong hợp của chúng.

Ví dụ. Hợp của tập hợp $\{1, 2, 3, 4, 5\}$ và $\{2, 4, 6\}$ là tập hợp $\{1, 2, 3, 4, 5, 6\}$.

Tính chất. Bằng biểu đồ Venn ta thấy dễ dàng thấy các tính chất sau của phép hợp hai tập hợp:

1. Luật đồng nhất: $A \cup \emptyset = A$ cho mọi tập hợp A.
2. Luật nuốt : $A \cup U = U$ cho mọi tập hợp $A \subseteq U$.
3. Luật lũy đẳng: $A \cup A = A$ cho mọi tập hợp A.
4. Luật giao hoán: $A \cup B = B \cup A$ cho tập hợp A và B tùy ý.
5. Luật kết hợp: $(A \cup B) \cup C = A \cup (B \cup C)$ cho tập hợp $A,\ B,\ C$ tùy ý.

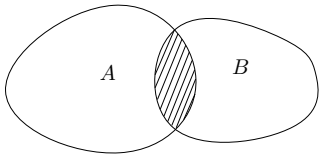
Lưu ý. Nhờ có định luật kết hợp, cho nên ta có thể viết $A \cup B \cup C$ thay cho viết $(A \cup B) \cup C$. Nhờ đó chúng ta có thể nói hợp của nhiều tập hợp $\bigcup_{i=1}^{n} A_i = A_1 \cup A_2 \cup \ldots \cup A_n$ mà không cần phải nói rõ thứ tự thực hiện phép hợp như thế nào.

Các tính chất của phép hợp (sau này là các phép toán khác) có thể còn được chứng minh nhờ sử dụng *bảng thuộc tính*. Để chỉ một phần tử thuộc

một tập hợp, ta dùng số 1, để chỉ một phần tử không thuộc tập hợp đó ta dùng số 0. Ta sẽ xét trong bảng thuộc tính tất cả các trường hợp có thể xảy ra khi xét một phần tử cho trước có thể thuộc hay không thuộc các tập hợp cho trước và chỉ ra rằng phần tử đó thuộc cả hai vế của đẳng thức cần chứng minh. Cụ thể, để chứng minh tính chất giao hoán của phép hợp, ta có bảng thuộc tính như sau. Bảng này có 4 dòng tương ứng với 4 khả năng có thể của một phần tử trong quan hệ thuộc tập hợp. Kết quả của bảng chỉ ra rằng phép hợp của hai tập hợp có tính giao hoán.

Tương tự ta có thể chứng minh tính kết hợp của phép hợp cũng bằng bảng thuộc tính.

Phép giao



5cm

Hình 2.4: Phép giao.

Bảng 2.1: Dùng bảng thuộc tính kiểm tra tính giao hoán và kết hợp.

A	B	$A \cup B$	$B \cup A$
1	1	1	1
1	0	1	1
0	1	1	1
0	0	0	0

A	B	C	$(A \cup B) \cup C$	$A \cup (B \cup C)$
1	1	1	1	1
1	1	0	1	1
1	0	1	1	1
1	0	0	1	1
0	1	1	1	1
0	1	0	1	1
0	0	1	1	1
0	0	0	0	0

Định nghĩa 2.4. *Cho trước tập hợp A và tập hợp B. Giao của tập hợp A và tập hợp B là tập hợp chứa tất cả các phần tử vừa thuộc A vừa thuộc B, và chỉ những phần tử đó mà thôi.*

Giao của hai tập hợp A và B được ký hiệu bởi $A \cap B$. Một cách hình thức ta viết:

$$A \cap B = \{x \mid x \in A \text{ và } \ x \in B\}.$$

Trong trường hợp tập hợp A và tập hợp B không có phần tử chung, giao của chúng là tập hợp $\emptyset$. Trong trường hợp này, chúng ta còn nói rằng hai tập hợp và là hai tập hợp rời nhau.

Ví dụ. Giao của tập hợp $\{1, 2, 3, 4, 5\}$ và $\{2, 4, 6\}$ là tập hợp $\{2, 4\}$.

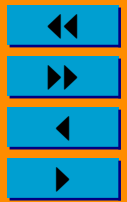
Tính chất. Bằng biểu đồ Venn, hoặc dùng bảng thuộc tính ta thấy dễ dàng các tính chất sau của phép giao hai tập hợp:

1. Luật nuốt : $A \cap \emptyset = \emptyset$ cho mọi tập hợp A.
2. Luật đồng nhất: $A \cap U = A$ cho mọi tập hợp $A \subseteq U$.
3. Luật lũy đẳng: $A \cap A = A$ cho mọi tập hợp A.
4. Luật giao hoán: $A \cap B = B \cap A$ cho tập hợp A và B tùy ý.
5. Luật kết hợp: $(A \cap B) \cap C = A \cap (B \cap C)$ cho tập hợp $A,\ B,\ C$ tùy ý.

Lưu ý. Nhờ có định luật kết hợp, cho nên ta có thể viết $A \cap B \cap C$ thay cho viết $(A \cap B) \cap C$. Nhờ đó, chúng ta có thể nói giao của nhiều tập hợp $\bigcap_{i=1}^{n} A_i = A_1 \cap A_2 \cap \ldots \cap A_n$ mà không cần phải nói rõ thứ tự thực hiện phép giao như thế nào.

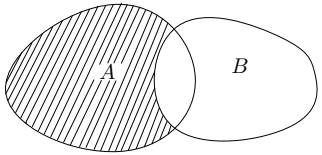
Ngoài ra, phép hợp và phép giao còn có tính chất phân phối, được phát biểu như sau:





6. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$, và
7. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ cho mọi tập hợp A, B và C tùy ý.

Phép trừ



Hình 2.5: Hiệu hai tập hợp.

Định nghĩa 2.5. *Cho trước tập hợp A và tập hợp B. Hiệu của tập hợp A với tập hợp B là tập hợp chứa tất cả các phần tử thuộc A mà không thuộc B, và chỉ những phần tử đó mà thôi. Hiệu của hai tập hợp A và B được ký hiệu là $A \setminus B$ hoặc $A - B$.*

$$A - B = \{x \mid x \in A \text{ và } x \notin B\}.$$

Trong biểu đồ Venn (hình 2.5), chúng ta biểu diễn $A - B$ bởi phần mặt phẳng biểu diễn tập hợp A bỏ đi phần chung với phần mặt phẳng biểu diễn tập hợp B.

Ví dụ. Hiệu của tập hợp $\{1, 2, 3\}$ với tập hợp $\{2, 3, 4\}$ là tập hợp $\{1\}$.

Trong trường hợp tập hợp A và tập hợp B không có phần tử chung thì hiệu của tập hợp A và tập hợp B chính là tập hợp A.

Ví dụ. Hiệu của tập hợp $\{1, 2, 3\}$ với tập hợp $\{4, 5, 6\}$ là tập hợp $\{1, 2, 3\}$.

Nếu tập hợp A là tập hợp con của tập hợp B thì hiệu của tập hợp A và tập hợp B chính là tập hợp $\emptyset$. Lưu ý rằng phép trừ của tập hợp không có tính đối xứng.

Ví dụ. Hiệu của tập hợp $\{1, 2, 3\}$ với tập hợp $\{4, 5, 6\}$ là tập hợp $\{1, 2, 3\}$, còn hiệu của tập hợp $\{4, 5, 6\}$ với tập hợp $\{1, 2, 3\}$ là tập hợp $\{4, 5, 6\}$.

Phép trừ đặc biệt quan trọng đối với họ các tập hợp con của một tập hợp cho trước.

Định nghĩa 2.6. *Cho trước một tập hợp A và một tập hợp U chứa tập hợp A. Khi đó ta nói hiệu $U - A$ là phần bù của tập hợp A trong tập hợp U và ký hiệu $U - A$ bởi $C_A(U)$ hoặc $\overline{A}_U$ và nếu không xảy ra hiểu lầm (khi ta hiểu đang nói tới các phần bù trong tập hợp cho trước U) thì ta có thể viết ngắn gọn là $\overline{A}$.*

Ví dụ. Cho trước tập hợp $A = \{0, 1, 2, 3\}$ và $U = \mathbb{N}$ thì $\overline{A} = \{4, 5, \ldots\}$.

Tính chất. Bằng biểu đồ Venn, và dùng bảng thuộc tính ta có thể chứng minh các tính chất sau của phần bù của tập hợp:





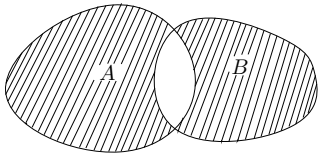



Back
Close

1. Luật bù: $\overline{\overline{A}} = A$.
2. Luật De Morgan cho giao 2 tập hợp: $\overline{A \cap B} = \overline{A} \cup \overline{B}$.
3. Luật De Morgan cho hợp của 2 tập hợp: $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

Trong mối liên quan tới phép trừ của tập hợp, có một phép toán được gọi là hiệu đối xứng của hai tập hợp cho trước.

Định nghĩa 2.7. *Hiệu đối xứng của hai tập hợp cho trước A và B là tập hợp chứa các phần tử chỉ thuộc đúng một trong hai tập hợp cho trước A và B này (hoặc thuộc tập hợp A hoặc thuộc tập hợp B), và chỉ chứa đúng các phần tử này mà thôi. Hiệu đối xứng của hai tập hợp cho trước A và B được ký hiệu là $A \Delta B$ hoặc $A \oplus B$.*

5cm 

Hình 2.6: Hiệu đối xứng.

Ví dụ. Hiệu đối xứng của tập hợp $\{1, 2, 3\}$ và tập hợp $\{2, 3, 4\}$ là tập hợp $\{1, 4\}$.

Trong biểu đồ Venn ở hình 2.6 ta thấy dễ dàng rằng $A \oplus B = (A - B) \cup (B - A)$. Do đó dễ thấy rằng hiệu đối xứng có tính chất giao hoán. Hiệu đối xứng của tập hợp tương ứng với phép tuyển có loại ở chương lôgic mệnh đề.

Hằng đẳng thức đáng nhớ

Với các phép toán hợp, giao và lấy phần bù của tập hợp, ta có 15 tính chất sau, được gọi là hằng đẳng thức đáng nhớ của tập hợp:

1. Luật đồng nhất: $A \cup \emptyset = A$ cho mọi tập hợp A.
2. Luật đồng nhất: $A \cap U = A$ cho mọi tập hợp $A \subseteq U$.
3. Luật nuốt : $A \cup U = U$ cho mọi tập hợp $A \subseteq U$.
4. Luật nuốt : $A \cap \emptyset = \emptyset$ cho mọi tập hợp A.
5. Luật lũy đẳng: $A \cup A = A$ cho mọi tập hợp A.
6. Luật lũy đẳng: $A \cap A = A$ cho mọi tập hợp A.
7. Luật giao hoán: $A \cup B = B \cup A$ cho tập hợp A và B tùy ý.

8. Luật giao hoán: $A \cap B = B \cap A$ cho tập hợp A và B tùy ý.
9. Luật kết hợp: $(A \cup B) \cup C = A \cup (B \cup C)$ cho tập hợp $A,\ B,\ C$ tùy ý.
10. Luật kết hợp: $(A \cap B) \cap C = A \cap (B \cap C)$ cho tập hợp $A,\ B,\ C$ tùy ý.
11. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$, và
12. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ cho mọi tập hợp A, B và C tùy ý.
13. Luật bù: $\overline{\overline{A}} = A$.
14. Luật De Morgan cho giao 2 tập hợp: $\overline{A \cap B} = \overline{A} \cup \overline{B}$.
15. Luật De Morgan cho hợp của 2 tập hợp: $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

Chúng ta có thể sử dụng 15 tính chất này để chứng minh các đẳng thức của tập hợp bằng phương pháp biến đổi đại số.
Ví dụ. $\overline{A \cup B \cup C} = \overline{(A \cup B) \cup C} = \overline{A \cup B} \cap \overline{C} = (\overline{A} \cap \overline{B}) \cap \overline{C} = \overline{A} \cap \overline{B} \cap \overline{C}$.



Ngoài phương pháp biến đổi đại số và phương pháp dùng bảng thuộc tính như trên đã trình bày, ta còn có thể dùng phương pháp dùng các qui tắc để chứng minh đẳng thức trong tập hợp.

Ví dụ. Chứng minh luật De Morgan $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

Chứng minh.

$$\begin{aligned}
\overline{A \cap B} &= \{x \mid x \notin (A \cap B)\} \\
&= \{x \mid \neg(x \in A \cap B)\} \\
&= \{x \mid \neg(x \in A \wedge x \in B\} \\
&= \{x \mid \neg(x \in A) \vee \neg(x \in B)\} \\
&= \{x \mid x \notin A \vee x \notin B\} \\
&= \{x \mid x \in \overline{A} \vee x \in \overline{B}\} \\
&= \{x \mid x \in \overline{A} \cup \overline{B}\} \\
&= \overline{A} \cup \overline{B}.
\end{aligned}$$

Tích Đêcac (Descartes)

Tích Đêcac của các tập hợp được đưa ra đầu tiên bởi nhà toán học Pháp René Descates khi ông biểu diễn mặt phẳng theo hai tọa độ vuông góc với

nhau.

Trên mặt phẳng tọa độ Đêcac, mỗi điểm được biểu diễn bởi một cặp hai số (x, y), với x là hoành độ và y là tung độ của điểm đã cho. Ta còn nói rằng mặt phẳng tọa độ chính là tích Đêcac của tập hợp số của trục tung và trục hoành. Một cách tổng quát, tích Đêcac của hai tập hợp được định nghĩa như sau.

Định nghĩa 2.8. *Cho A và B là hai tập hợp. Tích Đêcac của A và B là tập hợp của tất cả các cặp (a, b) với $a \in A$ và $b \in B$. .*

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

Với $A = \emptyset$ hoặc $B = \emptyset$ ta qui định $A \times B = \emptyset$.

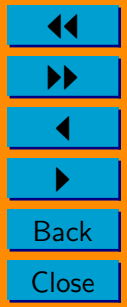
Ví dụ. $\{a, b\} \times \{1, 2\} = \{(a, 1), (a, 2), (b, 1), (b, 2)\}$.

Lưu ý. Tích Đêcac không có tính chất giao hoán như nhiều phép toán khác của tập hợp. Chẳng hạn,

$$\{a, b\} \times \{1, 2\} = \{(a, 1), (a, 2), (b, 1), (b, 2)\}$$
$$\{1, 2\} \times \{a, b\} = \{(1, a), (2, a), (1, b), (2, b)\}.$$

Tích Decac còn được mở rộng cho nhiều tập hợp. Ta có:





Định nghĩa 2.9. *Cho các tập hợp* A_i, $i = 1, 2, \ldots n$. *Khi đó tích Đêcac của* n *tập hợp này là* $A_1 \times A_2 \times \cdots A_n = \{(a_1, a_2, \cdots, n) \mid a_i \in A_i\}$.

Ví dụ. Với $A = \{+, -\}$, $B = \{a, b\}$ và $C = \{1, 2\}$ thì ta có

$$A \times B \times C = \{(+, a, 1), (+, a, 2), (+, b, 1), (+, b, 2), (-, a, 1), (-, a, 2), (-, b, 1), (-$$

Trong trường hợp các tập hợp A_i bằng nhau $A_1 = A_2 = \cdots = A_n = A$, thì thay vì viết $A_1 \times A_2 \times \cdots A_n$, ta có thể viết A^n.

Quan hệ và ánh xạ

Quan hệ

Định nghĩa 2.10. *Cho trước hai tập hợp A và B.*
$\mathcal{Q}$ quan hệ giữa A và B nếu $\mathcal{Q} \subset A \times B$.
Viết $a\mathcal{Q}b$ khi $(a, b) \in \mathcal{Q}$.

Ví dụ. A : tập hợp sinh viên trong một lớp.
$\mathcal{Q}_1$ là quan hệ *bạn bè*
$\mathcal{Q}_2$ là quan hệ *bạn thân*
$\mathcal{Q}_2 \subseteq \mathcal{Q}_1$.








Back
Close

$A = B$ quan hệ A và B là một quan hệ trong/trên A.

Định nghĩa 2.11. *Một quan hệ $\mathcal{Q}$ trong A là quan hệ tương đương khi và chỉ khi*

1. $\forall a \in A \to a\mathcal{Q}a$ *(phản xạ),*
2. $a\mathcal{Q}b \to b\mathcal{Q}a$ *(đối xứng),*
3. $a\mathcal{Q}b,\ b\mathcal{Q}c \to a\mathcal{Q}c$ *(bắc cầu).*

Ví dụ. $a\mathcal{Q}b \Leftrightarrow ab$ là số chính phương

Định nghĩa 2.12. *Họ các tập hợp con (A_i), $i = 1, 2, ..., n$ của A là phân hoạch:* $\Longleftrightarrow$
1) A_i là các tập hợp rời nhau,
2) $\bigcup A_i = A$.

Ví dụ. A tập hợp các số tự nhiên. A_1, A_2 là một phân hoạch của tập hợp A với A_1 là tập hợp các số tự nhiên chẵn, A_2 là tập hợp các số tự nhiên lẻ.
Tính chất. $\mathcal{Q}$ là một quan hệ tương đương trong $A \Longleftrightarrow$ họ các tập hợp $A_i = \{x \mid x\mathcal{Q}a\}$ với $a \in A$ là một phân hoạch của A.



Ánh xạ và hàm số

Định nghĩa 2.13. *Cho trước hai tập hợp A và B. Một quan hệ f của A và B được gọi là ánh xạ nếu:*
a) Với mọi $a \in A$ luôn tồn tại $b \in B$ sao cho $(a, b) \in f$,
b) Với b_1 và b_2 thuộc B sao cho (a, b_1), $(a, b_2) \in f$ ta sẽ có $b_1 = b_2$.

Ký hiệu: $f(a) = b$
b là ảnh của a
a là tạo ảnh của b.
$f : A \to B$. A được gọi là miền xác định
$\{f(a) \mid a \in A\}$ là miền giá trị của ánh xạ f.
Ví dụ. Đặt tên cho con là thiết lập ánh xạ từ tập hợp con người vào tập hợp tên (với giả thiết là mỗi con người chỉ có một tên chính).

Đánh số nhà là thiết lập một ánh xạ từ tập hợp các ngôi nhà vào tập hợp số tự nhiên...

Định nghĩa 2.14. *Cho trước các tập hợp A, B và C và các ánh xạ $f : A \to B$, $g : B \to C$. Khi đó ánh xạ được $h : A \to C$ xây dựng theo quy tắc $h(a) = g(f(a))$ được gọi là tích của hai ánh xạ f và g và được ký hiệu là $h = g \circ f$.*

Ví dụ. Cho trước các tập hợp $A = B = C = \mathbb{R}$ và $f(x) = x^2$ và $g(x) = x^3$ cho mọi $x \in \mathbb{R}$, thì $h(x) = x^6$ là ánh xạ tích $g \circ f$.

Có những loại ánh xạ đặc biệt sau đây.

Định nghĩa 2.15. *Cho trước hai tập hợp A và B. Một ánh xạ f được gọi là toàn ánh từ A vào B nếu nó thỏa mãn điều kiện*
c) với mỗi $b \in B$ tồn tại $a \in A$ sao cho $f(a) = b$.

Ví dụ. Phân phát hết kẹo trong túi cho các em bé sao cho em bé nào cũng có phần là một toàn ánh từ tập hợp kẹo vào tập hợp các em bé.

Định nghĩa 2.16. *Cho trước hai tập hợp A và B. Một ánh xạ f được gọi là đơn ánh từ A vào B, nếu nó thỏa mãn điều kiện*
d) nếu $f(a_1) = f(a_2)$ thì $a_1 = a_2$.

Ví dụ. Việc đánh số trang sách của một cuốn sách là một đơn ánh từ tập hợp các trang sách vào tập hợp các số tự nhiên.

Định nghĩa 2.17. *Cho trước hai tập hợp A và B. Một ánh xạ được gọi là song ánh từ A vào B, nếu nó đồng thời vừa là đơn ánh lại vừa là toàn ánh.*

Tóm lại, một song ánh là một quan hệ của các tập hợp và và thỏa mãn đồng thời 4 tính chất a), b), c) và d).

Ví dụ. Thiết lập mã số các sinh viên trong trường là lập một song ánh từ tập hợp các sinh viên với tập hợp mã số sinh viên. Yêu cầu được đặt ra là mỗi sinh viên có một mã số riêng biệt.

Một ánh xạ được gọi là hàm số nếu B là một tập hợp số.

Ví dụ. Hàm số f với $f(x) = x^2$ là một ánh xạ từ tập hợp số thực $\mathbb{R}$ vào chính nó.

Định nghĩa 2.18. *Nếu f là một song ánh từ A vào B, thì ánh xạ g được xây dựng theo quy tắc $g(b) = a$ cho $b \in B$ với $f(a) = b$, được gọi là ánh xạ ngược của ánh xạ f, và còn được ký hiệu là f^{-1}.*

$id_A : A \to A$ với $f(a) = a$ (ánh xạ đồng nhất trên tập hợp A).

Tính chất. Với f là song ánh từ tập hợp A vào tập hợp B thì ta dễ thấy $f \circ f^{-1} = id_B$ và $f^{-1} \circ f = id_A$.











Lực lượng của tập hợp

Lực lượng, tập hợp hữu hạn, tập hợp vô hạn

Định nghĩa 2.19. *Hai tập hợp cho trước A và B được nói là có cùng lực lượng nếu như tồn tại một song ánh $f : A \to B$.*

Ví dụ. Tập hợp số tự nhiên $\mathbb{N}$ và tập hợp số nguyên $\mathbb{Z}$ có cùng lực lượng.
Chứng minh. Ta xây dựng ánh xạ $f : \mathbb{Z} \to \mathbb{N}$ như sau:

$$f(z) = \begin{cases} 2z & \text{nếu } z \geq 0 \\ -2z - 1 & \text{nếu } z < 0. \end{cases}$$

Dễ thấy f là song ánh từ $\mathbb{Z}$ vào $\mathbb{N}$.

Định nghĩa 2.20. *Tập hợp hữu hạn là tập hợp không có cùng lực lượng với mọi tập hợp con thực sự của nó.*

Số lượng phần tử được ký hiệu bởi $|A|$.
Ví dụ. 1. $\{1, 2, 3\}$.

2. $A = \{n \mid n \in \mathbb{N} \text{ và } n \leq 1000000\}$.
3. $\mathcal{P} = \{p \mid n \in \mathbb{N} \text{ và } p \text{ là số nguyên tố}\}$.

Tính chất. 1. Tập hợp con của một tập hợp hữu hạn cũng là một tập hợp hữu hạn.
2. A là tập hợp hữu hạn thì tập hợp $\mathcal{P}(A)$ các tập con của nó cũng là tập hợp hữu hạn.

Ví dụ. Tập hợp tất cả các tập hợp con của $\{1, 2\}$ là $\{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$ cũng là một tập hợp hữu hạn. Tập hợp này chỉ có 4 phần tử.

Định nghĩa 2.21. *Một tập hợp được gọi là tập hợp vô hạn nếu nó không phải là tập hợp hữu hạn.*

Ví dụ. Các tập hợp $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ và tập hợp $\mathbb{C}$ là các tập hợp vô hạn.

Tính chất. Các tập hợp $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ là các tập hợp vô hạn có cùng lực lượng.

Tập hợp đếm được và tập hợp không đếm được

Định lý 2.1 (Cantor)**.** *Không tồn tại song ánh từ tập hợp số tự nhiên* $\mathbb{N}$ *vào tập hợp số thực trong khoảng* $(0, 1)$.

Định nghĩa 2.22. *Các tập hợp hữu hạn và các tập hợp vô hạn có lực lượng của tập hợp* $\mathbb{N}$ *các số tự nhiên được gọi là tập hợp đếm được. Các*



tập hợp vô hạn không có cùng lực lượng với tập hợp các số tự nhiên $\mathbb{N}$ *được gọi là tập hợp không đếm được.*

Ví dụ. Các tập hợp $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$ là các tập hợp vô hạn đếm được. Các tập hợp $\mathbb{R}$ và $\mathbb{C}$ là tập hợp không đếm được.











Tính chất.

1. Mọi tập hợp con của một tập hợp đếm được là đếm được.
2. Hợp của hai tập hợp đếm được là tập hợp đếm được.
3. Hợp của một họ các tập hợp đếm được là tập hợp đếm được.
4. Một tập hợp là tập hợp không đếm được nếu nó có một tập hợp con là tập hợp không đếm được.



Biểu diễn tập hợp trên máy tính

U tập hợp vũ trụ (chẳng hạn lấy U là hợp của các tập hợp cho trước).
$A \subset U \to a_1 a_2 ... a_n$ $a_i = 0$ nếu $i \notin A$ và $a_i = 1$ nếu $i \in A$.
$\emptyset$ là xâu gồm toàn các bít 0.
U là xâu gồm toàn các bít 1.
Ví dụ. $A = \{1, 2, 4\}$ và $B = \{1, 3, 5\}$
$U = \{1, 2, 3, 4, 5\}$ với $n = 5$.
A tương ứng 11010.
B tương ứng 10101.
Ví dụ. $U = \{1, 2, 3, 4, 5\}$
11011 ứng với $\{1, 2, 4, 5\}$.

Phần bù của một tập hợp ứng với xâu bit $a_1 a_2 ... a_n$, là xâu bit $b_1 b_2 ... b_n$ với b_i là phần bù của a_i (bù 0 là 1 và bù của 1 là 0).
Ví dụ. $A = \{1, 2, 4\}$ và $U = \{1, 2, 3, 4, 5\}$ Xâu biểu diễn A là 11010.
Xâu phủ định này là 00101.
00101 tương ứng tập hợp $\{3, 5\}$ là phần bù của tập hợp A trong U.








Back
Close

Phép hợp tương ứng OR-bit

Phép giao tương ứng với AND-bit

Hiệu đối xứng XOR-bit

Ví dụ. $A = \{1, 2, 4\}$

$B = \{1, 3, 5\}$

$U = \{1, 2, 3, 4, 5\}$.

A tương ứng 11010,

B tương ứng 10101.

$$11010 \vee 10101 = 11111$$
$$11010 \wedge 10101 = 10000$$
$$11010 \oplus 10101 = 01111.$$

$A \cup B = U = \{1, 2, 3, 4, 5\}$

$A \cap B = \{1\}$.

$A \oplus B = \{2, 3, 4, 5\}$.











Back

Close

Bài tập:

1. Hãy dùng biểu đồ Venn kiểm tra công thức $|A \cup B| = |A| + |B| - |A \cap B|$ cho các tập hợp hữu hạn A và B.

2. Cả lớp 12A phải làm một bài kiểm tra toán gồm có 3 bài toán. Giáo viên chủ nhiệm lớp báo cáo với nhà trường rằng cả lớp đều làm được ít nhất một bài, trong lớp có 20 em làm được bài toán thứ nhất, 14 em giải được bài toán thứ hai, 10 em giải được bài toán thứ ba, 6 em giải được cả hai bài thứ nhất và thứ ba, 5 em giải được cả hai bài thứ hai và thứ ba, 2 em giải được cả hai bài thứ nhất và thứ hai, và có mỗi một em được 10 điểm vì đã giải được cả ba bài. Hỏi rằng lớp học có bao nhiêu em tất cả?

3. Tích Đêcac $A \times B$ có bao nhiêu phần tử, nếu A có n và B có m phần tử.

4. Chứng minh rằng các tập hợp sau đây là tập hợp hữu hạn và hãy tính số các phần tử của chúng:
   a) $\mathcal{P}(\{1\})$,
   b) $\mathcal{P}(\{1, 2\})$,



c) $\mathcal{P}(\{1,2,3\})$,
ở đó $\mathcal{P}(A)$ ký hiệu tập hợp các tập hợp con của tập hợp A.

5. Chứng minh rằng mỗi tập con của một tập hợp hữu hạn cũng là một tập hợp hữu hạn.

6. Chứng minh tập hợp A cũng là tập hợp hữu hạn nếu tập hợp $\mathcal{P}(A)$ các tập con của A là tập hợp hữu hạn.

7. Chứng minh rằng nếu A là tập hợp vô hạn thì tập hợp thu được từ bằng cách bỏ đi hữu hạn phần tử bất kỳ của A vẫn là một tập hợp vô hạn.

8. Chứng minh rằng nếu A là một tập hợp hữu hạn, thì tập hợp thu được từ A bằng cách thêm vào nó hữu hạn phần tử vẫn là một tập hợp hữu hạn.

9. Chứng minh nếu tập hợp A là một tập hợp vô hạn thì tập hợp $\mathcal{P}(A)$ các tập con của A cũng là tập hợp vô hạn.

10. Chứng minh rằng nếu A là một bảng chữ cái (một tập hợp hữu hạn khác $\emptyset$) thì tập hợp các xâu được tạo bởi các chữ cái của A là một tập hợp đếm được.



11. Chứng tỏ rằng tập hợp các chương trình máy tính trong một ngôn ngữ lập trình đặc biệt nào đó là đếm được. (Gợi ý: sử dụng bài trên.)

12. Chứng minh rằng tập hợp các hàm số từ tập hợp $\mathbb{N} - \{0\}$ vào tập hợp $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ là một tập hợp không đếm được. (Gợi ý: ta cho ứng với mỗi hàm số $f : (\mathbb{N} - \{0\}) \to \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ một số thực $0, f(1)f(2)\ldots f(n)\ldots$).





Chương 3

Một số công thức tổ hợp

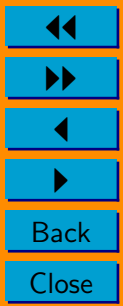
Euler, Gauß, Hamilton...

Nguyên lý Dirichlet, xác suất rời rạc...

Đếm số mật khẩu, số acounts, độ phức tạp thuật toán ...

Một số bài toán sau đây:

a) Bài toán đếm
có bao nhiêu cấu hình thoả mãn điều kiện đã nêu?: tính xác suất của một sự kiện, tính độ phức tạp của một thuật toán.



b) Bài toán liệt kê
Dạng thuật toán *vét cạn*...

c) Bài toán tối ưu
Cấu hình *tốt nhất* theo một nghĩa nào đấy,... xây dựng được những thuật toán hữu hiệu.

d) Bài toán tồn tại
có hay không có cấu hình... thúc đẩy nhiều ngành toán học phát triển.








Một số công thức cơ sở

Cơ sở của phép đếm các phần tử của một tập hợp

Định lý 3.1. *Nếu A_i $(i = 1, 2, \ldots n)$ là một phân hoạch của tập hợp A thì ta có* $\sum_{i=1}^{n} |A_i| = |A|$.

Một cách mở rộng định lý 3.1 là định lý sau:

Định lý 3.2. *Cho trước hai tập hợp hữu hạn A và B. Nếu tồn tại một toàn ánh $f : A \to B$ sao cho mỗi phần tử $b \in B$ có đúng k tạo ảnh trong A thì ta có $|A| = k|B|$.*

Số phần tử của tích Đêcac

Ví dụ. $A = \{a, b\}$.

Tất cả các dãy 3 ký tự của A là

aaa, aab, bbb, bba, abb, baa, aba, bab.

Định lý 3.3. *Cho trước các tập hợp hữu hạn A_i $(i = 1, 2, \ldots k)$, trong đó tập hợp A_i có đúng n_i phần tử. Khi đó tập tích Đê-cac $A_1 \times A_2 \cdots A_k$ có đúng $n_1 n_2 \cdots n_k$ phần tử. Nghĩa là*

$$|A_1 \times A_2 \cdots A_k| = |A_1| \times |A_2| \cdots |A_k|.$$

Và đặc biệt ta có $|A^n| = |A|^n$.

Ví dụ. Có bao nhiêu số tự nhiên có chín chữ số mà trong biểu diễn thập phân của nó không có mặt chữ số nào trong tập hợp $\{0; 1; 2; ; 3; 4; 5\}$?

Lời giải. Mỗi số tự nhiên có chín chữ số và không được viết bởi một trong các chữ số $\{0; 1; 2; 3; 4; 5\}$ là một dãy chín kí tự có lặp của tập hợp $\{6; 7; 8; 9\}$. Theo định lý 3.3 ta có tất cả $4^9 = 262144$ số như vậy.

Bài tập:

1. Số điện thoại của thành phố Hà nội là một số có 7 chữ số không có chữ số 0 đứng đầu. Hỏi rằng Hà nội có thể có nhiều nhất bao nhiêu số điện thoại khác nhau?

2. Một cái khoá số có 3 vòng số, mỗi vòng gồm 10 chữ số khác nhau. Khóa chỉ mở được nếu trên cả 3 vòng là các chữ số đúng của nó. Hỏi rằng trong trường hợp tồi nhất ta phải quay thử bao nhiêu lần?

3. Có bao nhiêu số tự nhiên có chín chữ số mà trong biểu diễn thập phân của nó không có mặt chữ số nào trong các số 0, 1, 2, 3, 4 và 6?

4. Có bao nhiêu số tự nhiên có chín chữ số mà trong biểu diễn thập phân của nó không có mặt chữ số nào trong các số 0, 1, 2, 3, 4 , 5 và 6?

5. Có bao nhiêu số tự nhiên là bội của 5 và có chín chữ số, mà trong biểu diễn thập phân của nó không có mặt chữ số nào trong các số 0, 1, 2, 3, 4 và 6?

6. Có bao nhiêu số tự nhiên có bốn chữ số và là bội của 5?

7. Có bao nhiêu số chẵn có ba chữ số mà trong biểu diễn thập phân của nó không có mặt chữ số nào trong các số 7, 8 và 9?

8. Có bao nhiêu số tự nhiên không vượt quá 2001 là bội của 3 và không có chữ số nào là một trong các số 3, 4, 5, 6, 7, 8, 9?

9. Có bao nhiêu ánh xạ khác nhau từ tập hợp có phần tử vào tập hợp có phần tử?

10. Có bao nhiêu mật khẩu độ dài 3 được viết bởi các chữ cái của bảng chữ cái Latin (26 chữ cái)?





Một số công thức tổ hợp

Chỉnh hợp

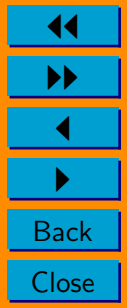
Bài toán: Cho trước một tập hợp A có n phần tử và một số tự nhiên $k \leq n$. Mỗi dãy độ dài k được xếp bởi các phần tử của tập hợp A, trong đó mỗi phần tử chỉ được phép có mặt không quá một lần, được gọi là một dãy k phần tử không lặp của A. Hãy tính số các dãy k phần tử không lặp của tập hợp A.

Chỉnh hợp chập k của n là số các dãy k phần tử không lặp của tập hợp A có n phần tử được kí hiệu A_n^k.

Ví dụ. Cho trước tập hợp $A = \{a, b, c, d\}$. Tất cả các dãy kí tự có hai phần tử không lặp của tập hợp A là :

$$ab, ac, ad, ba, bc, bd, ca, cb, cd, da, db, dc.$$

Tất cả có 12 dãy như vậy.



Định lý 3.4. *Cho trước một tập hợp A gồm n phần tử và một số tự nhiên $k \leq n$. Số các dãy k phần tử không lặp của A là* $A_n^k = \dfrac{n!}{(n-k)!}$.

Ví dụ. Có bao nhiêu số có bốn chữ số mà trong biểu diễn thập phân của nó không có hai chữ số nào giống nhau và không có mặt chữ số chẵn nào cả?

Lời giải. Số các số bốn chữ số mà trong biểu diễn thập phân của nó không có hai chữ số nào giống nhau và không có mặt chữ số chẵn nào cả là số các dãy không lặp của tập hợp $A = \{1, 3, 5, 7, 9\}$. Theo định lí chỉnh hợp không lặp thì ta có tất cả $5!/(5-4)! = 120$ số như vậy.

Bài tập:

1. Trong một giải đấu bóng đá quốc tế có 18 đội bóng đá tham gia. Sau trận đấu, người ta trao giải nhất, nhì và ba cho ba đội bóng đá xuất sắc nhất. Hỏi rằng có bao nhiêu khả năng có thể xảy ra?

2. Cho trước hai số tự nhiên k n. Hãy tính số các đơn ánh từ một tập hợp k phần tử vào một tập hợp n phần tử.

3. Với các chữ số 0, 1, 2, 3, 4 và 5 có thể lập được bao nhiêu số gồm năm chữ số khác nhau?











4. Với các chữ số 0, 1, 2, 3, 4, 5 và 6 có thể lập được bao nhiêu số là bội của 5 gồm năm chữ số khác nhau?

5. Có bao nhiêu số tự nhiên có mười chữ số đôi một khác nhau?

6. Số điện thoại của một thành phố gồm mười chữ số và có thể bắt đầu bằng chữ số 0. Hỏi rằng thành phố đó có thể có tối đa bao nhiêu số điện thoại khác nhau, và thành phố đó có thể có tối đa bao nhiêu số điện thoại nếu mỗi chữ số chỉ được xuất hiện trong nó không quá một lần?

Hoán vị

Cho trước một tập hợp n phần tử. Khi xếp các phần tử của chúng cạnh nhau ta có được một dãy n phần tử của tập hợp đã cho và gọi nó là một hoán vị của n phần tử đã cho.

Bài toán: Hãy tính số các hoán vị khác nhau của một tập hợp n phần tử.

Ví dụ. Cho tập hợp $A = \{a, b, c, d\}$. Tất cả những hoán vị có thể có của

tập hợp A này là:

$$abcd, abdc, acbd, acdb, adbc, adcb$$
$$bacd, badc, bcad, bcda, bdca, bdac$$
$$cabd, cadb, cbad, cbda, cdab, cdba$$
$$dabc, dacb, dbca, dbac, dcab, dcba.$$

Có tất cả 24 hoán vị.

Ký hiệu P_n là số hoán vị các phần tử của một tập hợp A gồm n phần tử cho trước. Hiển nhiên số tất cả các hoán vị của một tập hợp n phần tử chính là số A_n^n các dãy n phần tử không lặp của tập hợp n phần tử cho trước. Do đó ta có theo định lý 3.4:

Định lý 3.5. *Số các hoán vị khác nhau của một tập hợp n phần tử là $P_n = n!$.*

Ví dụ. Hãy tính số các số sau:
a) Có năm chữ số được viết bởi đúng năm chữ số 1, 2, 3, 4 và 5.
b) Có năm chữ số được viết bởi đúng năm chữ số 1, 2, 3, 4 và 5, trong đó ba chữ số đầu là ba số lẻ và hai chữ số sau là hai chữ số chẵn.
Lời giải. Theo định lí 3.5 ta có:

a) Số các số có năm chữ số được viết bởi đúng năm chữ số 1, 2, 3, 4 và 5 là 5!=120.
b) Số các số lẻ được viết bởi đúng ba chữ số 1, 3 và 5 là 3!=6, số các số chẵn được viết bởi đúng hai chữ số chẵn 2 và 4 là 2!=2. Do đó số các số năm chữ số được viết bởi đúng năm chữ số 1, 2, 3, 4 và 5, trong đó ba chữ số đầu là ba số lẻ và hai chữ số sau là hai chữ số chẵn là 3!2!=6.2=12.

Bài tập:

1. Cho năm chữ số 1, 2, 3, 4 và 5. Hãy tính số các số:
   a) Có năm chữ số đôi một khác nhau.
   b) Có năm chữ số đôi một khác nhau và bắt đầu bởi chữ số 1.
   c) Có năm chữ số đôi một khác nhau và không bắt đầu bởi chữ số 1.
   d) Có năm chữ số đôi một khác nhau và bắt đầu bởi 24.
   e) Có năm chữ số đôi một khác nhau và không bắt đầu bởi số 241.

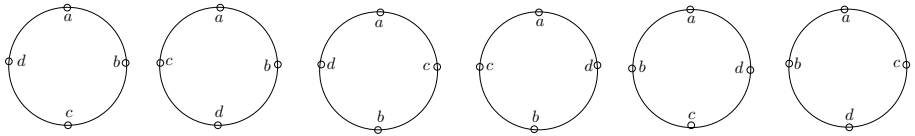
2. Hãy tính tổng của tất cả các số có năm chữ số được viết bởi đúng năm chữ số 1, 2, 3, 4 và 5.

Hoán vị trên đường tròn

Nếu ta xét bài toán xếp n số trên đường tròn thì hai hoán vị khác nhau $(1, 2, ..., n)$ và $(2, 3, ..., n, 1)$ chỉ là một cách xếp mà thôi.

Bài toán: Tính số các hoán vị khác nhau của một tập hợp A gồm n phần tử nằm trên một đường tròn.

Ví dụ. Cho tập hợp $A = \{a, b, c, d\}$. Tất cả các hoán vị khác nhau của các phần tử của A trên đường tròn là $abcd$, $abdc$, $acbd$, $acdb$, $adbc$, $adcb$ (xem hình 3.1). Dễ dàng ta có kết quả sau:



Hình 3.1: Tất cả các cách xếp 4 phần tử trên một đường tròn.

Định lý 3.6. *Số các hoán vị tròn khác nhau của tập hợp A có n phần tử là* $Q_n = (n-1)!$.

Ví dụ. Người ta sản xuất những viên đá hoa hình vuông được chia làm 4 hình vuông nhỏ bằng nhau. Trong mỗi hình vuông nhỏ được in một chữ

G để trang trí. Các chữ G được in trong các hình vuông nhỏ theo 4 vị trí khác nhau: đặt thẳng, nằm ngang vuông $90°$, nằm ngang vuông $-90°$ và đặt lộn ngược. Các chữ G được in sao cho trong mỗi mẫu đá hoa có đủ 4 vị trí khác nhau của chữ G. Hỏi rằng có bao nhiêu mẫu đá hoa khác nhau có thể có?

Lời giải. Ta đặt cho mỗi cách in chữ G một số khác nhau 1, 2, 3 và 4. Mỗi mẫu gạch đá hoa có thể có là một hoán vị trên đường tròn của tập hợp $A = \{1, 2, 3, 4\}$. Như vậy số các mẫu đá hoa khác nhau đúng bằng số hoán vị trên đường tròn của tập hợp $A = \{1, 2, 3, 4\}$, theo công thức của định lý 3.6, là $(4 - 1)! = 6$.

Tổ hợp

Tổ hợp chập k của n phần tử là số các tập hợp con k phần tử của một tập hợp n phần tử cho trước.

Ví dụ. Cho tập hợp $A = \{a, b, c, d\}$. Tất cả các tập hợp hai phần tử của A là

$$\{a; b\}, \{a; c\}, \{a; d\}, \{b; c\}, \{b; d\}, \{c; d\}.$$

Kí hiệu tổ hợp chập k của n là C_n^k, ta có định lý sau đây.

Định lý 3.7. *Số tập hợp con k phần tử của một tập hợp A có n phần tử cho trước là* $C_n^k = \dfrac{n!}{k!(n-k)!}$.

Tính chất. Bạn đọc có thể chứng minh dễ dàng các tính chất sau đây của các hệ số trong khai triển lũy thừa của nhị thức

1. $C_n^k = C_n^{n-k}$,
2. $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$,
3. $C_n^k = C_{n-1}^{k-1} + C_{n-2}^{k-1} + \cdots + C_{k-1}^{k-1}$.

Bài tập:

1. Trong một khoang của tàu hỏa có hai hàng ghế đối mặt nhau. Mỗi hàng ghế có bốn chỗ ngồi. Trong số tám hành khách có hai người muốn ngồi nhìn theo hướng tàu chạy, còn hai người muốn ngồi nhìn ngược hướng tàu chạy. Tính số cách xếp chỗ cho tám người khách đó trên hai hàng ghế này thỏa mãn được yêu cầu của họ?

2. Với các chữ số 0, 1, 3, 6 và 9 có thể lập được bao nhiêu số :
   a) Có bốn chữ số khác nhau?
   b) Có bốn chữ số khác nhau và là số chẵn?
   c) Có bốn chữ số khác nhau và là bội của 3?

3. Cho trước hai số nguyên dương $k \leq n$. Hãy tính số các đơn ánh thỏa mãn điều kiện $f(1) < f(2) < ... < f(k)$ từ tập hợp $\{1, 2; \ldots, k\}$ vào tập hợp $\{1, 2, \ldots ..., n\}$.

4. Trên một đường tròn cho trước n điểm. Hãy tính số các đoạn thẳng nối tất cả các cặp điểm của n điểm này.

5. Trong một n-giác lồi ta kẻ tất cả các đường chéo. Biết rằng không có ba đường chéo nào trong chúng đồng quy. Hãy tính số giao điểm của các đường chéo này.

Chỉnh hợp với tần số lặp cho trước

Một phần tử a_i của một dãy k phần tử cho trước $a_1 a_2 \cdots a_k$ được nói là có tần số lặp k_i, nếu như nó xuất hiện trong dãy đúng k_i lần. Ký hiệu $P(k_1, k_2, \ldots, k_n)$ là số các dãy có lặp $(k_1, k_2, \ldots, k_n)$ (là dãy độ dài $k_1 +$

$k_2 + \cdots + k_n$ các phần tử của một tập hợp A có n phần tử cho trước, sao cho trong mỗi dãy này thì phần tử thứ i của A xuất hiện đúng k_i lần trong dãy).

Ví dụ. Cho tập hợp $A = \{a, b, c, d\}$. Tất cả các dãy có lặp $(3, 2, 0, 0)$ của A là:

$$aaabb, aabba, abbaa, bbaaa, aabab,$$
$$ababa, babaa, abaab, baaba, baaab.$$

Định lý 3.8. *Cho trước một tập hợp A có n phần tử. Số các dãy k phần tử có lặp $(k_1, k_2, ..., k_n)$ $(k = k_1 + k_2 + \ldots + k_n)$ là* $P(k_1, k_2, \ldots, k_n) = \dfrac{k!}{k_1! k_2! \cdots k_n!}$.

Ví dụ. Tính số các số tự nhiên có bảy chữ số, trong đó có ba chữ số 1, hai chữ số 2 và hai chữ số 3.

Lời giải. Số các số tự nhiên cần tìm là $P(3, 2, 2) = 7!/3!2!2! = 210$.

Bài tập:

1. Hãy tính số các mật khẩu khác nhau (có thể vô nghĩa) thu được bằng cách hoán vị các chữ cái của từ TINHOCTUOITRE.

2. Hãy tính số các mật khẩu khác nhau (có thể vô nghĩa) thu được bằng cách hoán vị các chữ cái của từ TINHOCTUOITRE mà trong đó không có ba chữ T đứng cạnh nhau.

3. Hỏi với các chữ số 0, 2, 4, 6, 8 và 9 có thể thiết lập được bao nhiêu mật khẩu có tám chữ số mà trong đó chữ số 9 có mặt đúng ba lần, còn các chữ số khác xuất hiện đúng một lần?

4. Có 12 máy tính. Để sử lý song song một thuật toán, người ta muốn ghép chúng thành 6 cặp, mỗi cặp có 2 máy tính. Hỏi rằng có bao nhiêu cách ghép tất cả?








Hai nguyên lý cơ bản

Nguyên lý nhân

Công đoạn E được thực hiện qua k công đoạn E_1, E_2, ... E_k, trong đó số cách thực hiện E_i là n_i $(i = 1, 2...k)$. Số các cách thực hiện E thông qua việc thực hiện E_1 rồi đến thực hiện E_2 ...và rồi E_k (kế tiếp nhau) là $n_1 n_2 ... n_k$.

Ví dụ. Từ một thành phố A phải đi qua thành phố B rồi mới tới thành phố C. Hãy tính số cách có thể đi từ A tới C biết rằng đi từ A tới B có 7 con đường khác nhau và số con đường đi từ B tới C là 9.

Lời giải. Trong bài toán này ta phải thực hiện nguyên tắc nhân, và do đó số cách đi từ A tới C là $7.9 = 63$.

Nguyên lý cộng

Giả sử chúng ta có một công đoạn E mà mỗi khi thực hiện E là phải thực hiện trên một trong k công đoạn rời nhau E_1, E_2, ... E_k, trong đó số cách

thực hiện E_i là n_i $(i = 1, 2...k)$. Số các cách thực hiện được một trong các chọn lựa E_1, E_2,.. E_k (độc lập với nhau) là $n_1 + n_2 + ... + n_k$.

Ví dụ. Hãy tính số cách có thể đi từ Hà nội tới TP Hồ Chí Minh với ba phương tiện khác nhau: đường không, đường thủy và đường bộ. Biết rằng có 3 hãng hàng không, 2 hãng tàu thủy và 15 hãng giao thông đường bộ có phương tiện đi lại giữa hai thành phố nói trên.

Lời giải. Trong bài toán này rõ ràng chúng ta phải thực hiện nguyên lý cộng, vì mỗi cách đi từ Hà nội tới TP Hà Chí Minh chỉ có thể là một trong ba loại phương tiện: đường thủy, đường không hoặc đường bộ mà thôi. Số cách đi từ Hà nội tới TP Hồ Chí Minh là $3 + 2 + 15 = 20$.

Trong nhiều bài toán chúng ta thường phải áp dụng cả hai nguyên tắc kể trên trong cùng một lúc (có thể cách dùng rất khác nhau nhưng sẽ luôn cùng đi tới một đáp số giống nhau).

Bài tập:

1. Người ta sản xuất những viên đá hoa hình vuông được chia làm 4 hình vuông nhỏ bằng nhau. Trong mỗi hình vuông nhỏ được in một chữ G để trang trí. Các chữ G được in trong các hình vuông nhỏ theo 4 vị trí khác nhau: đặt thẳng, nằm ngang 90, nằm ngang -90 và đặt lộn ngược. Hỏi rằng có bao nhiêu mẫu gạch đá hoa khác nhau có thể có sao cho

có ít nhất hai ô vuông đối xứng qua tâm của viên đá được in giống hệt nhau còn hai ô còn lại khác kiểu nhau và là kiểu của hai trong số ba mẫu chữ G còn lại?

2. Người ta sản xuất những viên đá hoa hình vuông được chia làm 4 hình vuông nhỏ bằng nhau. Trong mỗi hình vuông nhỏ được in một chữ G để trang trí. Các chữ G được in trong các hình vuông nhỏ theo 4 vị trí khác nhau: đặt thẳng, nằm ngang 90, nằm ngang -90 và đặt lộn ngược. Hỏi rằng có bao nhiêu mẫu gạch đá hoa khác nhau có thể có sao cho có ít nhất hai ô vuông đối xứng qua tâm của viên đá được in giống hệt nhau và khác kiểu của hai ô còn lại?

3. Tính số các ước số của số 2310.

4. Có 2 học sinh nữ và 10 học sinh nam phụ trách công tác lễ tân trong một buổi lễ của nhà trường. Hỏi rằng có bao nhiêu cách lập nhóm tiếp khách 8 người mà trong đó phải có ít nhất một học sinh nữ?

Khai triển lũy thừa của đa thức

Nhị thức Newton

Bằng các công thức tổ hợp trên, chúng ta có thể triển khai một cách hình thức lũy thừa của $(x+y)^n$ (còn gọi là công thức nhị thức Newton). Chúng ta đã biết những công thức khai triển nhị thức sau

$$(x+y)^2 = x^2 + 2xy + y^2,$$
$$(x+y)^3 = x^3 + 3x^2y + 3xy^2 + y^3.$$

Nếu nhân cả hai vế của đẳng thức cuối với $x+y$, ta nhận được công thức sau:

$$(x+y)^4 = x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + y^4.$$

Chúng ta có công thức tổng quát tính hệ số của $(x+y)^n$ như sau:

Định lý 3.9. *Cho mỗi số tự nhiên $n \geq 1$ ta có*

$$(x+y)^n = \sum_{k=0}^{n} C_n^k x^{n-k} y^k.$$

Tam giác Pascal, trong đó số bắt đầu và kết thúc của mỗi dòng trong tam giác Pascal là số 1, và mỗi số khác của mỗi dòng bằng tổng của hai số của dòng trên nó, cho ta cách tính nhanh chóng hệ số của nhị thức Newton.

$$\begin{array}{ccccccccc}
 & & & & 1 & & & & \\
 & & & 1 & & 1 & & & \\
 & & 1 & & 2 & & 1 & & \\
 & 1 & & 3 & & 3 & & 1 & \\
1 & & 4 & & 6 & & 4 & & 1
\end{array}$$

.....

Trong thực tế của lập trình, các chương trính sử dụng tam giác Pascan (theo nguyên lý đệ qui) chạy nhanh hơn chương trình dùng nhị thức Newton.



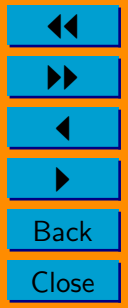


Định lý 3.10. *Cho trước m và n là các số nguyên dương. Khi đó ta có*

$$(x_1 + x_2 + ... + x_m)^n = \sum_{k_1+k_2+\cdots k_m=n} \frac{n!}{k_1!k_2!...k_m!} x_1^{k_1} x_2^{k_2} \cdots x_m^{k_m}.$$

tổng được lấy trên tất cả các bộ m số tự nhiên $(k_1, k_2, ..., k_m)$ sao cho $k_1 + k_2 + ... + k_m = n$.

Ví dụ. $(x_1 + x_2 + x_3)^3 = \sum_{k_1+k_2+k_3=3} \frac{3!}{k_1!k_2!k_3!} x_1^{k_1} x_2^{k_2} x_3^{k_3}$. Tất cả các bộ $k_1 + k_2 + k_3 = 3$ là $(3, 0, 0)$, $(2, 1, 0)$, $(2, 0, 1)$, $(1, 2, 0)$, $(1, 0, 2)$, $(1, 1, 1)$, $(0, 3, 0)$, $(0, 2, 1)$, $(0, 1, 2)$, $(0, 0, 3)$. Khi thay các bộ này vào công thức



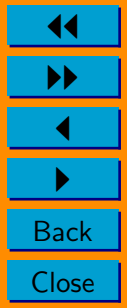
khai triển trên ta thu được:

$$
\begin{aligned}
(x_1+x_2+x_3)^3 &= \sum_{k_1+k_2+k_3=3} \frac{3!}{k_1!k_2!k_3!} x_1^{k_1}x_2^{k_2}x_3^{k_3} \\
&= \frac{3!}{3!0!0!}x_1^3x_2^0x_3^0 + \frac{3!}{2!1!0!}x_1^2x_2^1x_3^0 + \frac{3!}{2!0!1!}x_1^2x_2^0x_3^1 + \\
&\quad + \frac{3!}{1!2!0!}x_1^1x_2^2x_3^0 + \frac{3!}{1!0!2!}x_1^1x_2^0x_3^2 + \frac{3!}{1!1!1!}x_1^1x_2^1x_3^1 + \\
&\quad + \frac{3!}{0!3!0!}x_1^0x_2^3x_3^0 + \frac{3!}{0!2!1!}x_1^0x_2^2x_3^1 + \frac{3!}{0!1!2!}x_1^0x_2^1x_3^2 + \\
&\quad + \frac{3!}{0!0!3!}x_1^0x_2^0x_3^3 \\
&= x_1^3 + 3x_1^2x_2 + 3x_1^2x_3 + 3x_1x_2^2 + 3x_1x_3^2 + 6x_1x_2x_3 + \\
&\quad + x_2^3 + 3x_2^2x_3 + 3x_2x_3^2 + x_3^3 \\
&= x_1^3 + x_2^3 + x_3^3 + 3x_1^2x_2 + 3x_1^2x_3 + 3x_1x_2^2 + 3x_1x_3^2 + \\
&\quad + 3x_2^2x_3 + 3x_2x_3^2 + 6x_1x_2x_3.
\end{aligned}
$$

Bài tập:

1. Hãy tính hệ số của x^{15} trong khai triển của $(1-x^2+x^3)^{20}$.





2. Gọi a_i là hệ số của x_i trong khai triển của đa thức $(1+x+x^2)^n$. Hãy chứng minh rằng:
   a) $a_0a_1 - a_1a_2 + a_2a_3 - ..... - a_{2n-1}a_{2n} = 0$,
   b) $\sum_{i=0}^{n} (-1)^i a_i^2 = an$,
   c) $a_0 + a_2 + a_4 + ... = (3n+1), a_1 + a_3 + a_5 + ... = (3n-1)$,
   d) $a_k - a_{k-1} + a_{k-2} - .... + (-1)^k a_0 = 0, (\mod 3)$.

3. Tính các hệ số c_m của x^m trong khai triển của đa thức

$$P(x) = (1+x)^k + (1+x)^{k+1} + \cdots + (1+x)^n,$$

   với các số tự nhiên $n > k$ cho trước.

4. Chứng minh rằng $C_n^m C_m^k = C_n^k C_{n-k}^{m-k} = C_n^{m-k} C_{n-m+k}^k$.

5. Chứng minh rằng $C_n^0 + C_n^1 + \cdots + C_n^n = 2^n$

6. Chứng minh rằng $C_n^0 - C_n^1 + C_n^2 - \cdots + (-1)^n C_n^n = 0$.

7. Chứng minh rằng $C_{2n}^n = (C_n^0)^2 + (C_n^1)^2 + \cdots + (C_n^n)^2$.



Công thức tổ hợp trong tập hợp

Tính số phần tử của một hợp các tập hợp

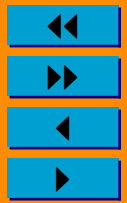
Trong cuộc sống nhiều khi xuất hiện những bài toán phải tính số lượng phần tử của một tập hợp thông qua những tập hợp con của chúng. Chẳng hạn:

Ví dụ. Trong một bài kiểm tra toán có hai câu hỏi. Trong cả lớp có 30 em làm được câu thứ nhất và 20 em làm được câu thứ hai. Chỉ có 10 em làm được cả hai câu hỏi. Hãy tính số học sinh trong lớp.

Lời giải. Gọi A là tập hợp học sinh giải được câu thứ nhất, và B là tập hợp học sinh giải được câu thứ hai, thì $A \cap B$ là tập hợp học sinh giải được cả hai câu. Bài toán được đặt ra là phải tính số phần tử của $A \cup B$.

Sử dụng công thức quen thuộc $|A \cup B| = |A| + |B| - |A \cap B|$, bởi vì trong tổng $|A| + |B|$ thì các phần tử chung trong $A \cap B$ (của cả A và B) được tính lặp đúng hai lần.

Sử dụng công thức này, ta thấy số học sinh của lớp trong ví dụ trên của ta là 30+20-10=40 em.



Back
Close

Không khó khăn, chúng ta có thể thấy công thức sau là đúng:

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

Theo công thức của ta, số học sinh trong lớp sẽ là 20+14+10-6-5-2+1=32. Một cách tổng quát, ta có định lý sau.

Định lý 3.11. *Cho trước các tập hợp A_1, A_2,... A_n. Khi đó ta có*

$$|A_1\cup A_2\cup\cdots A_n| = \sum |Ai|-\sum |A_i\cap A_j|+...+(-1)^{n+1}|A_1\cap A_2\cap\cdots A_n|. \quad (*)$$

Nhiều khi, bài toán ta gặp trở nên phức tạp hơn khi phải tính số phần tử của một phép hợp với nhiều hơn hai tập hợp.

Ví dụ. Học sinh của một lớp 12 phải làm bài thi gồm có ba bài toán. Biết rằng mỗi em trong lớp đều làm được ít nhất một bài, trong lớp có 20 em làm được bài toán thứ nhất, 14 em giải được bài toán thứ hai, 10 em giải được bài toán thứ ba, 6 em giải được cả hai bài thứ nhất và thứ ba, 5 em giải được cả hai bài thứ hai và thứ ba, 2 em giải được cả hai bài thứ nhất và thứ hai, và có mỗi một em được 10 điểm vì đã giải được cả ba bài. Hỏi rằng lớp học có bao nhiêu em tất cả?

Lời giải. Gọi A là tập hợp các em học sinh giải được bài toán thứ nhất,



B là tập hợp các em học sinh giải được bài toán thứ hai và C là tập hợp các em học sinh giải được bài toán thứ ba, ta phải tính số phần tử của tập hợp $A \cup B \cup C$.

Tính số các tập hợp thỏa mãn tính chất cho trước

Cho trước một tập hợp A gồm n phần tử. Ta đã biết trong các phần trước là số các tập hợp con k phần tử của A là . Trong nhiều bài toán chúng ta phải tính số tập hợp con của A thỏa mãn tính chất cho trước nào đó. Trước hết ta có định lý sau đây về số các tập hợp con của một tập hợp A cho trước:

Định lý 3.12. *Số tập hợp con của một tập hợp A có n phần tử là $2n$. Số các tập hợp con có lẻ phần tử bằng số các tập hợp con có chẵn phần tử của A và là 2^{n-1}.*

Bài tập:

1. Trong một kì thi học sinh giỏi Toán, Lí và Hóa có một số em tham gia. Biết rằng có 20 em tham gia thi Toán, 14 em tham gia thi Lí, 10 em thi Hóa, 6 em vừa thi Toán vừa thi Lí, 5 em vừa thi Lí vừa thi Hóa, 2



em thi Toán và thi Hóa và có một em tham gia tất cả 3 môn Toán, Lí và Hóa. Hỏi rằng có bao nhiêu em tham gia kì thi học sinh giỏi này?

2. Tính số các mật khẩu tạo được bằng cách hoán vị các chữ cái của từ "TINHOCTRE" sao cho không có hai chữ cái nào giống nhau đứng cạnh nhau.

3. Tính số các mật khẩu được viết độ dài $2k$ ký tự của một tập hợp gồm k chữ cái khác nhau sao cho mỗi chữ cái có mặt đúng 2 lần và không có hai chữ cái nào đứng cạnh nhau giống nhau.
   a) Tính số các số này khi $k = 3$,
   b) Tính số các số này khi $k = 4$.








Back
Close











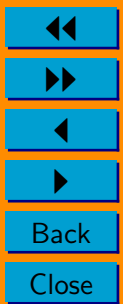
Back

Close



Chương 5

Đại số Boole và cấu trúc mạch lôgic



Biểu thức Boole và hàm Boole

Định nghĩa đại số Boole

Định nghĩa 5.1. *Đại số Boole là một tập hợp B với ba phép toán: phép lấy phần bù (được kí hiệu với $-$), phép lấy tổng Boole (được kí hiệu $+$) và phép nhân Boole (được kí hiệu "."). Tập hợp B có hai phần tử đặc biệt là 0 và 1 sao cho các đẳng thức sau được thỏa mãn:*

1. $b.1 = b + 0 = b$ *cho mọi* $b \in B$, *(luật đồng nhất)*
2. $b + \overline{b} = 1, b.\overline{b} = 0, \forall b \in B$ *(luật bù)*
3. $(x + y) + z = x + (y + z)$ *và* $(x.y).z. = x.(y.z)$ *(luật kết hợp)*
4. $x + y = y + x$ *và* $x.y = y.x$ *(luật giao hoán)*
5. $x(y+z) = x.y+x.z$*và* $(x.y)+z = (x+z).(y+z)$ *(luật phân phối).*

Thứ tự ưu tiên: lấy phần bù, tích Boole, tổng Boole.
Ngoặc trong cùng được thực hiện trước.

Ví dụ.

1. $B = \{0, 1\}$ với
   1. $0.0 = 0.1 = 1.0 = 0 + 0 = 0$, $1.1 = 1 + 0 = 0 + 1 = 1 + 1 = 1$,
   2. $\overline{0} = 1, \overline{1} = 0$.
2. B là tập hợp các mệnh đề lôgic với các phép toán phủ định lôgic, $\vee$ và $\wedge$ cũng là một đại số Boole với phần tử 0 là $\mathbb{F}$ và phần tử 1 là $\mathbb{T}$.
3. $B = \mathcal{P}(A)$ với A là một tập hợp cho trước với phép tính lấy phần bù, hợp và giao của tập hợp.



Ví dụ. Tính $x + \overline{x}(1 + y)$.

Lời giải. Áp dụng luật tính toán của các phép toán đại số Boole ta có:

$$\begin{aligned} x + \overline{x}(1 + y) &= x + (\overline{x}.1 + \overline{x}.y) && \text{(Luật phân phối)} \\ &= x + (\overline{x} + \overline{x}.y) && \text{(Luật đồng nhất)} \\ &= (x + \overline{x}) + \overline{x}.y && \text{(Luật kết hợp)} \\ &= 1 + \overline{x}.y && \text{(Luật nuốt)} \\ &= 1. && \text{(Luật nuốt)} \end{aligned}$$











Các hằng đẳng thức của đại số Boole

1. $p.1 = p$ (luật đồng nhất),
2. $p + 0 = p$ (luật đồng nhất),
3. $p.0 = 0$ (luật nuốt),
4. $p + 1 = 1$ (luật nuốt),
5. $p + p = p$ (luật lũy đẳng),
6. $p.p = p$ (luật lũy đẳng),
7. $\overline{\overline{p}} = p$ (luật phủ định kép),
8. $p + q = q + p$ (luật giao hoán),
9. $p.q = q.p$ (luật giao hoán),
10. $(p + q) + r = p + (q + r)$ (luật kết hợp),
11. $(p.q).r = p.(q.r)$ (luật kết hợp),
12. $(p + q).r = p.r + q.r$ (luật phân phối của phép nhân với phép cộng),

13. $(p.q) + r = (p + r).(q + r)$ (luật phân phối của phép cộng với phép nhân),

14. $\overline{p + q} = \overline{p}.\overline{q}$ (luật De Morgan đối với phép cộng),

15. $\overline{p.q} = \overline{p} + \overline{q}$ (luật De Morgan đối với phép nhân).








Back
Close

Biểu thức Boole và hàm Boole

Biến Boole: biến có giá trị 0 hoặc 1.

Định nghĩa 5.2. *Một hàm số Boole F với n biến $x_1, x_2, ..., x_n$, ký hiệu $F(x_1, x_2, ..., x_n)$ (gọi là hàm Boole bậc n) là một ánh xạ $F : \{0,1\}^n \to \{0,1\}$.*

Bảng 5.1: Hàm số Boole được biểu diễn qua bảng

x	y	$F(x,y)$
1	1	1
1	0	0
0	1	0
0	0	0



Định nghĩa 5.3. *Biểu thức Boole n biến x_1, x_2, ..., x_n:*
1. 0, 1, x_1, x_2, ..., x_n là các biểu thức Boole,
2. Nếu F, G là các biểu thức Boole thì $\overline{F}$, $F.G$, $F+G$ cũng là các biểu thức Boole.

Định nghĩa 5.4. *Hai hàm $F(x_1, x_2, ..., x_n)$ và $G(x_1, x_2, ..., x_n)$ là hai hàm Boole bằng nhau nếu $F(x_1, x_2, ..., x_n) = G(x_1, x_2, ..., x_n)$ cho mọi giá trị của các biến.*








Luật đối ngẫu

Định nghĩa 5.5. *Phép nhân và phép cộng là hai phép đối ngẫu với nhau. Số 0 và số 1 cũng là đối ngẫu với nhau. Biểu thức F gọi là biểu thức đối ngẫu với biểu thức G nếu F thu được từ G bằng cách thay các phép toán và các hằng bởi đối ngẫu của nó.*

Đối ngẫu của F ký hiệu là F_d.

Ví dụ.

$F = x.0 + y(y + \overline{z} + 1)$ thì $F_d = (x + 1).(y + y.\overline{z}.0)$.

Với $F = x + xy$, thì $F_d = x.(x + y)$.

Lưu ý: Người ta dễ nhầm đối ngẫu F_d của F với phần bù $\overline{F}$ của F.



Nguyên lý đối ngẫu. Nếu ta có đẳng thức $F = G$, thì bằng đối ngẫu ta cũng có đẳng thức $F_d = G_d$.

Ví dụ. Từ đẳng thức $x+xy = x$, ta thu được đẳng thức mới $x(x+y) = x$ qua phép lấy đối ngẫu.











Xác định biểu thức Boole của hàm Boole

Phương pháp dùng tiểu hạng

Định nghĩa 5.6. *Tiểu hạng là một tích các biến $x_1, x_2, ..., x_n$ hoặc phần bù của nó (tục biến).*

$F(x_1, x_2, ..., x_n) = 1$ tại duy nhất $(a_1, a_2, ..., a_n)$ là tiểu hạng

$$F(x_1, x_2, ..., x_n) = y_1 y_2 ... y_n,$$

với $y_i = x_i$ khi $a_i = 1$ và $y_i = \bar{x}_i$ nếu $a_i = 0$.

Ví dụ.

Hàm Boole nhận giá trị 1 duy nhất tại $(1, 1)$ là xy.

Hàm Boole nhận giá trị 1 duy nhất tại $(0, 0)$ là $\bar{x}\bar{y}$.

Hàm không đồng nhất với 0 là tổng các tiểu hạng ứng với các giá trị 1 của hàm Boole.

Bảng 5.2: Hàm số $F(x, y) = xy + \bar{x}\bar{y}$

x	y	$F_1(x, y)$
1	1	1
1	0	0
0	1	0
0	0	1








Phương pháp dùng đại hạng

Định nghĩa 5.7. *Đại hạng là một tổng các biến $x_1, x_2, ..., x_n$ hoặc phần bù của nó (tục biến).*

$F(x_1, x_2, ..., x_n) = 0$ tại duy nhất $(a_1, a_2, ..., a_n)$ là đại hạng

$$F(x_1, x_2, ..., x_n) = y_1 + y_2 + ... + y_n,$$

với $y_i = x_i$ khi $a_i = 0$ và $y_i = \bar{x}_i$ nếu $a_i = 1$.
Ví dụ. $F(x_1, x_2) = 0$ chỉ tại $(1, 0)$ là biểu thức $F(x_1, x_2) = \bar{x}_1 + x_2$.

Hàm không đồng nhất với 1 là tích các đại hạng ứng với các giá trị 0 của hàm Boole.

Bảng 5.3: Hàm số $F(x, y) = (\bar{x} + y).(x + \bar{y})$

x	y	$F_1(x, y)$
1	1	1
1	0	0
0	1	0
0	0	1



Tính đầy đủ của các phép toán Boole

Hàm Boole bất kỳ bao giờ cũng biểu diễn được thành biểu thức Boole.

Do:

$x + y = \overline{\overline{x}.\overline{y}}$,

$x.y = \overline{\overline{x} + \overline{y}}$.

Nên có thể bỏ bớt phép $+$ hoặc phép $.$ thì ham Boole tùy ý vẫn biểu diễn thành biểu thức với 2 phép toán còn lại.

Lưu ý. Không thể loại bỏ phép lấy phần bù.











NAND là $|$ và được định nghĩa như sau:

$1|1 = 0, 1|0 = 0|1 = 0|0 = 1.$

NOR là $\downarrow 1$ và được định nghĩa :

$1 \downarrow 1 = 1 \downarrow 0 = 0 \downarrow 1 = 0, 0 \downarrow 0 = 1.$

$\{|\}$ và $\{\downarrow\}$ đều đầy đủ.

Bài tập: Hãy chứng tỏ NAND và NOR là đầy đủ.

Gợi ý: $\overline{x} = x|x,\ x + y = (x|x)|(y|y),\ x.y = (x|y)|(x|y)$

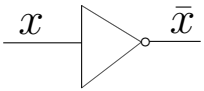
$\overline{x} = x \downarrow x,\ x.y = (x \downarrow x) \downarrow (y \downarrow y),\ x + y = (x \downarrow y) \downarrow (x \downarrow y)$

Sơ đồ mạch

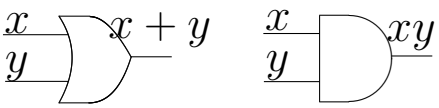
Các cổng lôgic cơ bản

Các mạch không có khả năng nhớ là các mạch tổ hợp.

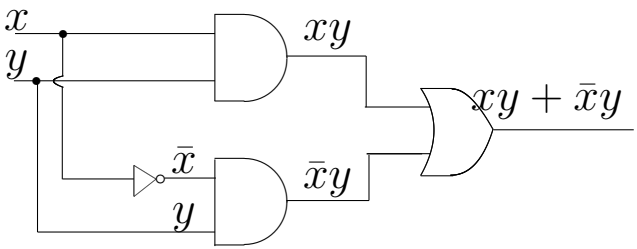
Các cổng cơ bản:



Hình 5.1: Cổng NOT = cổng đảo.



Hình 5.2: Cổng OR bên trái và cổng AND bên phải.



Hình 5.3: Mạch biểu diễn $xy + \bar{x}y$.

Ví dụ. Một máy dập lỗ kim loại cần phải được chế tạo hai nút bấm liền để có thể bảo đảm an toàn cho người sử dụng máy. Người sử dụng sau khi đặt miếng kim loại vào máy phải bấm cả hai nút hai tay, để tránh không thò tay vào máy dập. Hãy thiết kế mạch theo yêu cầu đó.











Lời giải. Hai nút bấm điện được ta cho ứng với biến x và y. Khi một công tắc được đóng mạch, thì biến tương ứng có giá trị chân lý bằng 1. Trạng thái máy dập sẽ là một hàm mệnh đề hai biến $F(x, y)$ chỉ có giá trị 1 tại $(x, y) = (1, 1)$ (là khi cả hai công tắc được đóng mạch). Như vậy, ta có bảng giá trị 5.4.

Bảng 5.4: Hàm Boole biểu diễn máy dập kim loại.

x	y	$F(x, y)$
1	1	1
1	0	0
0	1	0
0	0	0

Bài tập

Thiết kế sơ đồ mạch cho công tắc bóng đèn cầu thang.



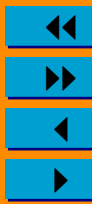
Chương 6

Thuật toán

Có nhiều bài toán tổng quát xuất hiện trong toán học rời rạc. Chẳng hạn tìm số lớn nhất trong một dãy số nguyên cho trước, liệt kê hết tất cả các tập hợp con của một tập hợp cho trước, xếp lại một dãy số nguyên theo thứ tự tăng dần hoặc giảm dần...

Một dãy các bước thực hiện để tới được nghiệm bài toán được gọi là thuật toán.

Một vấn đề quan trọng của thuật toán là ta phải đánh giá và so sánh được các thuật toán với nhau.



Khái niệm thuật toán

Định nghĩa thuật toán

Abu Jafar Mohammed ibn Musa al Khowarizmi (khoảng năm 825): algorithm.

Ví dụ: Thuật toán Euclid:
Input: *m, n nguyên dương.*
Output: *g (ước chung lớn nhất của m và n).*
Phương pháp:
Bước 1: *Tìm r, phần dư của m cho n.*
Bước 2: *Nếu $r = 0$, thì $g := n$ (gán giá trị của n cho g), và dừng lại. Trong trường hợp ngược lại ($r \neq 0$), thì $m := n$; $n := r$ và quay lại bước 1.*

Định nghĩa 6.1. *Thuật toán là một dãy hữu hạn các bước, mỗi bước mô tả chính xác các phép toán, hoặc hành động cần thực hiện ... kết thúc sau hữu hạn bước thực hiện cho ta lời giải của bài toán.*

Các yêu cầu về thuật toán

Định nghĩa trên về thuật toán tất nhiên còn chứa nhiều điều chưa rõ ràng. Để hiểu đầy đủ ý nghĩa của khái niệm thuật toán, chúng ta đưa ra 5 đặc trưng sau đây của thuật toán.

1. **Tính đứng đắn.** *Mọi kết quả của sự thực hiện thuật toán đều phải là kết quả đúng.*

2. **Tính xác định.** *Ở mỗi bước, các bước thao tác phải hết sức rõ ràng, không gây nên sự nhập nhằng và bước tiếp theo phải rõ ràng.*

3. **Tính khả thi.** *Tất cả các phép toán có mặt trong thuật toán phải đủ đơn giản và thực hiện được.*

4. **Tính đa năng.** *Thuật toán phải thực hiện được cho tất cả các tập hợp dữ liệu có thể của đầu vào.*

5. **Tính dừng.** *Với mọi bộ dữ liệu vào thoả mãn các điều kiện của dữ liệu vào (tức là được lấy ra từ các tập của dữ liệu vào), thuật toán phải dừng lại sau một số hữu hạn bước thực hiện.*

Bài tập:



1. *Mô tả thuật toán tìm phần tử lớn nhất của một dãy hữu hạn số thực.*

2. *Mô tả thuật toán xếp lại một dãy theo thứ tự tăng dần.*

3. *Mô tả thuật toán tìm một dãy các số liên tiếp nhau có tổng dương trong một dãy số thực cho trước.*





Các vấn đề liên quan tới thuật toán

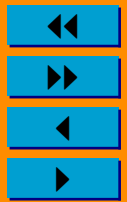
Thiết kế thuật toán

Một số kỹ thuật thiết kế thuật toán chung như:
- Chia để trị (divide-and-conque),
- phương pháp tham ăn (greedy method),
- qui hoạch động (dynamic programming)...

Tính đúng đắn của thuật toán

Ta cần phải chứng minh rằng, thuật toán khi được thực hiện sẽ cho ta kết quả đúng với mọi dữ liệu vào hợp lệ.

Ví dụ. thuật toán Euclid:
Bước 1, ta có $m = qn + r$ *,trong đó* q *là số nguyên nào đó. Nếu* $r = 0$ *thì* n *là ước của* m *và hiển nhiên* n *(do đó* g*) là ước chung lớn nhất của* m *và* n*.*



Nếu $r \neq 0$ thì một ước chung bất kỳ của m và n cũng là ước chung của n và r (vì $r = m - qn$).

Ngược lại một ước chung bất kỳ của n và r cũng là ước chung của m và n (vì $m = qn + r$).

Phân tích thuật toán

Chúng ta cần chọn thuật toán nào trong số các thuật toán đó để áp dụng.

Đánh giá hiệu quả của thuật toán.

hai tiêu chuẩn đánh giá thuật toán:

1. *Thuật toán đơn giản, dễ hiểu, dễ cài đặt (dễ viết chương trình)*
2. *Thuật toán sử dụng tiết kiệm nhất các nguồn tài nguyên của máy tính, và đặc biệt chạy nhanh nhất có thể được.*

Ta sẽ cài đặt thuật toán có thể sẽ rất phức tạp, miễn là chương trình nhận được chạy nhanh hơn so với các chương trình khác.





Tiêu chuẩn (2) được xem là tính hiệu quả của thuật toán. Tính hiệu quả của thuật toán bao gồm hai nhân tố cơ bản:
+ Dung lượng không gian nhớ cần thiết để lưu giữ các giữ liệu vào, các kết quả tính toán trung gian và các kết quả của thuật toán...
+ Thời gian cần thiết để thực hiện thuật toán (ta gọi là thời gian chạy).

Bài tập:

1. *Xác định số phép tính so sánh nhiều nhất có thể trong thuật toán của bạn tìm phần tử lớn nhất của một dãy n số thực cho trước.*

2. *Xác định số phép tính so sánh nhiều nhất có thể trong thuật toán của bạn để xếp lại một dãy có n số thực theo thứ tự tăng dần.*

3. *Xác định số bước thực hiện nhiều nhất có thể trong thuật toán của bạn nhằm xác định được nhiều nhất có thể các số liên tiếp nhau có tổng dương trong một dãy n số hạng cho trước.*








Back
Close

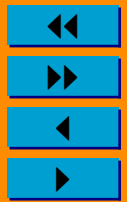
Các phương pháp biểu diễn thuật toán



Ngôn ngữ thuật toán là ngôn ngữ dùng để miêu tả thuật toán. Thông thường ngôn ngữ thuật toán bao gồm ba loại:
+ Mô tả từng bước;
+ Sơ đồ khối;
+ Ngôn ngữ lập trình.

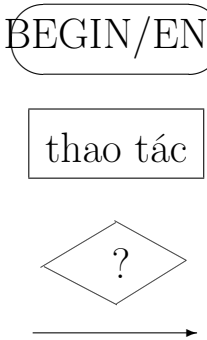
Mô tả từng bước

Thuật toán: *Tên thuật toán và chức năng.*
Input: *Dữ liệu vào với tên kiểu.*
Output: *Các dữ liệu ra với tên kiểu.*
Biến phụ (nếu có) gồm tên kiểu.
Hành động là các thao tác với các lệnh có nhãn là các số tự nhiên.
Ví dụ. Để giải phương trình bậc hai $ax^2 + bx + c = 0$, *ta có thể mô tả thuật toán bằng ngôn ngữ mô tả thuật toán như sau:*



Bước 1: *Nhập các hệ số a, b, c.*
Bước 2: *Kiểm tra xem các hệ số a có khác 0 hay không? Nếu $a = 0$ quay lại thực hiện bước 1.*
Bước 3: *Tính biểu thức $\Delta = b^2 - 4ac$.*
Bước 4: *Nếu $\Delta < 0$ thông báo phương trình vô nghiệm và chuyển sang bước 7.*
Bước 5: *Nếu $\Delta = 0$, tính $x_1 = x_2 = \dfrac{-b}{2a}$ và chuyển sang bước 7.*
Bước 6: *Tính $x_1 = \dfrac{-b - \sqrt{\Delta}}{2a}, x_2 = \dfrac{-b + \sqrt{\Delta}}{2a}$ và chuyển sang bước 7.*
Bước 7: *Kết thúc thuật toán.*





Hình 6.1: Các khối cơ bản.

Phương pháp sơ đồ khối

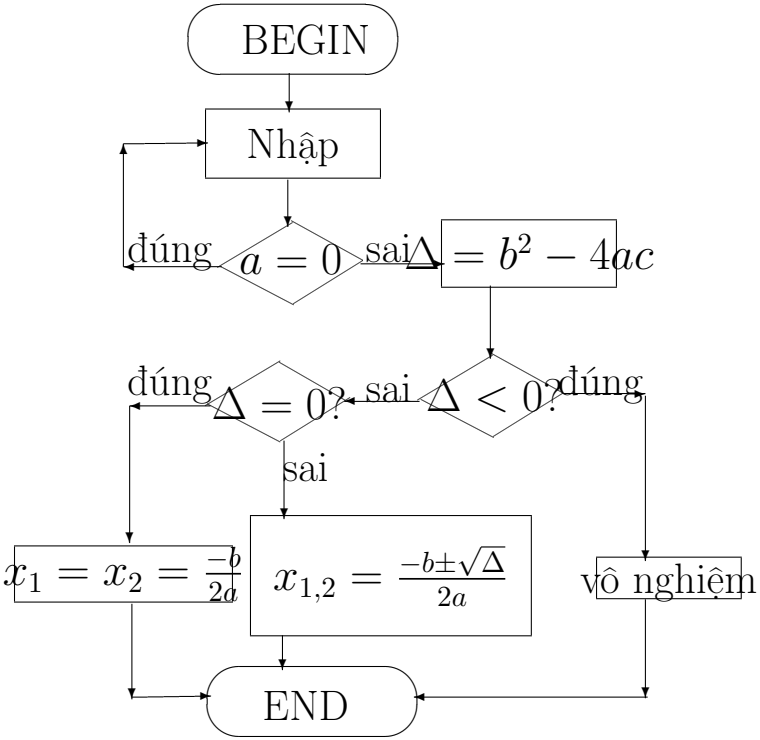
Phương pháp dùng sơ đồ khối mô tả thuật toán là dùng mô tả theo sơ đồ trên mặt phẳng các bước của thuật toán. Sơ đồ khối có ưu điểm là rất trực giác dễ bao quát. Để mô tả thuật toán bằng sơ đồ khối ta cần dựa vào các nút sau đây:

Nút khởi đầu, kết thúc: *Thường được biểu diễn bằng hình tròn thể hiện sự bắt đầu hay kết thúc quá trình.*

Nút thao tác: *Biểu diễn bằng hình chữ nhật,*

Nút điều khiển: *Được biểu diễn bằng hình thoi.*

Cung : *Đoạn nối từ nút này đến nút khác và có mũi tên chỉ hướng.*



Hình 6.2: Sơ đồ khối của thuật toán giải phương trình bậc hai.

Ví dụ. Trong hình 6.2 chúng ta có sơ đồ khối mô tả thuật toán trên.

Bài tập:

Mô tả các thuật toán của bạn bằng sơ đồ khối:

1. *Thuật toán tìm phần tử lớn nhất của một dãy hữu hạn số thực.*
2. *Thuật toán tìm phần tử bé nhất của một tập con của tập hợp.*
3. *Thuật toán xếp lại một dãy theo thứ tự tăng dần.*
4. *Thuật toán tìm một dãy các số liên tiếp nhau (dài nhất có thể) có tổng dương trong một dãy số thực cho trước.*








Ngôn ngữ tựa Pascal

Bảng chữ cái và ký tự chủ yếu

Ký tự và biểu thức:
Các ký tự la tinh: A, a, ..., Z, z. Chữ số: 0...9.
*Các phép toán số học: + , − , * , /. Các phép toán quan hệ: < , >, =. Giá trị lôgic: T (true), F (false)*
Phép toán logic: and, or, not
Hằng là các giá trị cụ thể nào đó
Tên biến: Là một dãy kí tự mà kí tự đầu phải là chữ cái.
Có hai loại biến chính:

Loại integer (biến nguyên). Ví dụ: **Var** *bien : integer;*

Loại Real (biến thực). Ví dụ: **Var** *bien : real;*
Ngoài ra còn biến chỉ số. Ví dụ: A[i], ở đây i là các biến nguyên.

Biểu thức là kết hợp các hằng, biến và các phép toán.

Một số câu lệnh chính

a) **Đầu chương trình.**
program *(tên chương trình). Tên chương trình có cấu trúc giống tên biến.*

Ví dụ. **Program** *giaipt;*
.....

b) **Các thủ tục và hàm.**
Câu lệnh Procedure (function). Mô tả thuật toán trong ngôn ngữ tựa Pascal, được bắt đầu bằng câu lệnh procedure (function), trong đó ta đặt tên cho thuật toán và mô tả danh sách biến của thuật toán. Chẳng hạn, câu lệnh
Function $\max(a, b, c)$*;*
cho biết tên của thuật toán là $\max$ *và các biến là* a, b, c
Procedure *move(n,A,B,C);*
Cho biết tên thuật toán được mô tả là move với các biến là n, A, B, C*;*
Các bước của thuật toán được mô tả trong thân thủ tục (hàm) được bắt đầu bởi Begin và kết thúc bởi end.

Ví dụ.

Function $\max(a, b, c)$*;*

Begin

..... (thân hàm)

End *;*

Procedure *move*(n, A, B, C)*;*

Begin

..... (thân thủ tục)

End *;*

c) **Câu lệnh gán.**

Câu lệnh gán được dùng để gán giá trị cho các biến Vế trái của câu lệnh gán là tên của biến, còn vế phải là biểu thức của các hằng, biến đã gán giá trị hoặc các hàm đã được định nghĩa. Ký hiệu $:=$ *được sử dụng để biểu diễn phép gán.*











Ví dụ. $Variable := exp;$
$Max := a;$
$x :=$ *số lớn nhất trong các số* $a, b, c \ldots$

d) **Câu lệnh khối.**
Các câu lệnh có thể nhóm lại thành một khối. Để mô tả khối lệnh ta sử dụng Begin và end
Ví dụ.

Begin

Câu lệnh 1;

Câu lệnh 2;

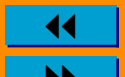
.....

Câu lệnh n;

End *;*

Các câu lệnh trong khối được thực hiện tuần tự. Dưới đây thuật ngữ câu lệnh được dùng để chỉ chung một câu lệnh cũng như một khối câu lệnh.








Chú giải. *Để diễn giải thêm nội dung của các đoạn lệnh có thể dùng câu diễn giải.Các dòng diễn giải được đặt trong dấu* $\{\ldots\}$ *hoặc* $(*\ldots*)$

*Ví dụ. (*a là phần tử lớn nhất trong danh sách L*)*

(k là số phần tử của danh sách L*)*

e) **Câu lệnh điều kiện.**
Câu lệnh đơn giản là

If *điều kiện* **then** *câu lệnh;*

Khi thực hiện câu lệnh, điều kiện sẽ được kiểm tra, nếu nó được thoả mãn thì câu lệnh sẽ được thực hiện. Nhiều khi ta cần thực hiện một thao tác nào đó khi điều kiện được thực hiện còn nếu ngược lại ta phải thực hiện một thao tác khác. Khi đó ta có thể thực hiện câu lệnh phức tạp hơn sau đây:

If *điều kiện* **then** *câu lệnh 1*
else câu lệnh 2;

f) **Các câu lệnh lặp.**
Các câu lệnh sau đây sẽ được sử dụng








Back
Close

For *biến:=gía trị đầu* **to** *giá trị cuối* **do** *câu lệnh;*

Tại đầu vòng lặp, biến sẽ được gán cho giá trị đầu, nếu giá trị đầu nhỏ hơn hơn hoặc bằng giá trị cuối và câu lệnh được thực hiện với giá trị này của biến. Tiếp đến giá trị của biến sẽ tăng lên 1và câu lệnh sẽ được thực hiện với giá trị mới của biến. Quá trình sẽ được tiếp tục cho đến khi biến bằng gía trị cuối. Sau khi thực hiện câu lệnh với biến bằng giá trị cuối sẽ chuyển sang thực hiện câu lệnh tiếp theo. Nếu giá trị đầu lớn hơn giá trị cuối thì không có câu lệnh nào được thực hiện.

Câu lệnh lặp thứ hai được sử dụng là câu lệnh "while"

While *điều kiện* **do** *câu lệnh;*

Khi câu lệnh này được sử dụng, điều kiện sẽ được kiểm tra, nếu nó là đúng thì câu lệnh được thực hiện. Điều đó sẽ tiếp tục cho đến khi điều kiện sai. Câu lệnh tiếp theo được sử dụng là câu lệnh "repeat".

Repeat *câu lệnh*

.....

until *điều kiện;*











Khi câu lệnh này được sử dụng thì câu lệnh trong vòng lặp được thực hiện, điều đó có thể sẽ dẫn đến sự thay đổi giá trị của các biến trong điều kiện. Nếu điều kiện vẫn là đúng, thì câu lệnh lại được thực hiện. Điều đó sẽ tiếp diễn cho đến khi điều kiện là đúng.

Ví dụ. Để giải phương trình bậc hai $ax^2 + bx + c = 0$ *ta mô tả thuật toán bằng một chương trình viết trên Pascal như sau :*

Program *giaiPT;*
Var $a, b, c, denta, x1, x2 : real;$
Begin
Clrscr;
Write(*Nhập hệ số:*);
Repeat
Write(*a:=*); **readln** (*a*);
Write (*b:=*); **readln** (*b*);
Write (*c:=*); **readln** (*c*);
Until *a<>0;*
*Denta:=sqr(b)-4*a*c;*

```
If  denta<0 then
 begin
   Write( phương trình vô nghiệm);
   Halt ;
 end
Else
   begin
     If denta=0 then
        begin
          Write (phương trình có nghiệm kép x=,(b/(2*a)));
          exit ;
        end
     else
        begin
          X1:=(-b-sqrt(denta))/(2*a);
          X2:=(-b+sqrt(denta))/(2*a);
```











```
            Write (phương trình có hai nghiệm phân biệt:);
            Write (x1 = , x1 , x2 = ,x2);
            exit ;
        end ;
    end;
  readln;
End.
```

Bài tập:

1. *Viết thuật toán của bạn bằng ngôn ngữ tựa Pascal mô tả thuật toán tìm phần tử lớn nhất của một dãy hữu hạn số thực.*
2. *Viết thuật toán của bạn bằng ngôn ngữ tựa Pascal tìm phần tử bé nhất của một tập hợp con của tập hợp.*
3. *Viết thuật toán của bạn bằng ngôn ngữ tựa Pascal xếp lại một dãy theo thứ tự tăng dần.*
4. *Viết thuật toán của bạn bằng ngôn ngữ tựa Pascal tìm một dãy các số liên tiếp nhau có tổng dương trong một dãy số thực cho trước.*



Một vài thuật toán đặc biệt

Thuật toán đệ qui

Cơ sở xây dựng thuật toán đệ qui là ta dùng phương pháp đệ qui để định nghĩa đối tượng và chuyển đổi định nghĩa đề qui sang thuật toán đệ qui.

Ví dụ. Viết thuật toán đệ qui tính giá trị a_n với a là số thực và n là số tự nhiên.

Lời giải. Trước hết ta xây dựng a_n bằng định nghĩa đề qui.

Bước cơ sở: Với giá trị nhỏ nhất của n là 0, ta định nghĩa $a_0 = 1$.

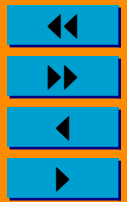
Bước đệ qui: Ta định nghĩa giá trị a_{n+1} thông qua đẳng thức $a_{n+1} = a_n.a$.

Thuật toán được viết cụ thể như sau:

Procedure *luythua (a: số thực; n : số tự nhiên);*

if $n = 0$ **then** $luythua(a, n) := 1$

else $luythua(a, n) := luythua(a, n - 1).a$



Back
Close

Cũng tương tự như tính lũy thừa một số tự nhiên, ta có thể tính $n!$ bằng phương pháp đệ qui.

Ví dụ. Viết thuật toán đệ qui tính giá trị $n!$ với n là số tự nhiên.

Lời giải. Hàm $f(n) = n!$ được định nghĩa đệ qui như sau. Nó là hàm số thỏa mãn:

1. $f(0) = 1$,

2. $f(n+1) = (n+1).f(n)$.

Thuật toán được viết như sau:

Procedure *giaithua*$(n \in \mathbb{N})$

if $n = 0$ **then** $giaithuagt(n) := 1$

else $giaithuagt(n) := n * giaithuagt(n \breve{} 1)$

Ví dụ. Viết thuật toán đệ qui tính giá trị số hạng thứ n của dãy số Fibonacci với n là số tự nhiên.

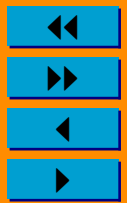
Lời giải. Trước hết ta nhớ là dãy Fibonacci được định nghĩa đệ qui là dãy số (f_n) thỏa mãn $f_0 = f_1 = 1$ và $f_{n+1} = f_n + f_{n-1}$ cho $n \geq 2$.

Procedure $Fib(n \in \mathbb{N})$
  if $n = 0$ **then** $Fib(n) := 0$
  else if $n = 1$ **then** $Fib(n) := 1$
  else $Fib(n) := Fib(n \breve{} 1) + Fib(n \breve{} 2)$



Thuật toán sinh các hoán vị và tổ hợp

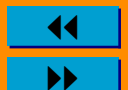
Phần sau đây giải quyết vấn đề lập trình sinh ra các hoán vị và các tổ hợp. Nhiều bài toán cần giải phải có được kỹ năng này.
Ví dụ. Nếu phải tìm cho một doanh nhân một hành trình bán hàng qua một số các thành phố cho trước và trở về thành phố ban đầu sao cho phí tổn đi lại là nhỏ nhất (lưu ý hành trình đi và hành trình ngược lại giữa hai thành phố có thể có phí tổn khác nhau) thì ta có thể phải xem xét tất cả các hoán vị có thể của các thành phố. Còn với bài toán khác ta lại phải biết cách lập ra các tổ hợp. Chẳng hạn từ 10 số cho trước cần chọn ra 6 số có tổng bằng 20 chẳng hạn. Khi đó, khó có thuật toán tốt để chọn được 6 số như vậy ngoài cách xét tất cả các tổ hợp 6 số của 10 số đã cho.



Cho trước tập hợp n phần tử, chẳng hạn tập hợp $A = \{1, 2, ..., n\}$. *Có nhiều thuật toán sinh ra n! hoán vị của A.*

Sau đây chúng ta làm quen với thuật toán sinh tất cả các hoán vị bằng phương pháp thứ tự từ điển. Với phương pháp này, các hoán vị được so sánh với nhau theo phương pháp sắp xếp thứ tự các chữ cái của từ điển. Ta nói hoán vị $(x_1, x_2, ..., x_n)$ *đứng trước hoán vị* $(y_1, y_2, ..., y_n)$ *nếu như tồn tại* $k < n$ *sao cho* $x_1 = y_1, ..., x_k = y_k, x_{k+1} < y_{k+1}$.
Ví dụ. Với $n = 6$ *hoán vị* $(6, 5, 2, 1, 3, 4)$ *đứng trước hoán vị* $(6, 5, 3, 1, 4, 2)$ *vì chúng trùng nhau ở hai vị trí đầu tiên, nhưng vị trí thứ ba của hoán vị đầu tiên có giá trị nhỏ hơn vị trí thứ ba của hoán vị thứ hai. Ngoài ra có thể thấy là không có hoán vị nào có thể được xếp giữa hai hoán vị này, cho nên trong dãy được sắp các hoán vị, chúng là hai hoán vị liên tiếp nhau. Phương pháp dùng trong thuật toán này là xuất phát từ hoán vị đầu tiên là hoán vị* $(1, 2, .., n)$ *ta xây dựng hoán vị tiếp ngay theo nó ... và cứ thế cho tới hoán vị cuối cùng là hoán vị* $(n, n-1, ..., 2, 1)$. *Sau đây là thuật toán sinh các hoán vị bằng cách xây dựng các hoán vị kế tiếp nhau.*









Back
Close

Procedure *next;* {*hoán vị* $(a1, a2, ..., an)(n, n-1, ..., 2, 1)$}

var j, k, r, s, z :byte;

Begin

j:= n-1;

while *a[j] > a[j+1]* **do** *j:=j-1; so lon nhat a[j]<a[j+1]*

k:=n;

while *(a[j]>a[k])* **do** *k:=k-1; so nho nhat sao cho a[k]*

begin

z:=a[j];

a[j]:=a[k];

a[k]:=z;

end;

r:=n;









```
s:=j+1;
while (r>s) do
begin
    z:=a[r];
    a[r]:=a[s];
    a[s]:=z;
    r:=r-1;
    s:=s+1;
end;
end;
```

Bài toán tương tự với bài toán sinh tất cả các hoán vị là bài toán sinh ra tất cả các tổ hợp của một tập hợp $A = \{1, 2, ..., n\}$. *Vì mỗi tổ hợp chập* r *của* n *là một tập hợp con* r *phần tử của tập hợp* n *phần tử. Như vậy tương ứng với việc tạo ra tất cả các tổ hợp thì ta thiết kế tất cả các*

tập hợp con r phần tử của A. Lưu ý rằng ta biết có phương pháp biểu diễn một tập hợp con thông qua một xâu bit độ dài n, trong đó bit thứ i là 1 nếu phần tử i thuộc tập con và là 0 nếu ngược lại.

Vì mỗi xâu bit độ dài n tương ứng với một số tự nhiên không vượt quá $2^{n+1} - 1$*, cho nên ta có thể dùng thứ tự số tự nhiên để sắp xếp xâu bit tương ứng với nó.*

Ví dụ. Xâu bit 0001100111 tương ứng với số trong hệ đếm cơ số 2. Xâu bit tiếp theo là số tiếp theo, có giá trị tương ứng với xâu 0001101000.

Dễ thấy qui luật tạo xâu kế tiếp khá đơn giản. Ta chỉ việc bắt đầu từ phải qua trái thay các bit 1 bởi các bit 0 cho tới khi gặp bit 0 đầu tiên thì thay bit này bởi 1 và kết thúc. Thủ tục tạo xâu bit tiếp theo xâu bit cho trước được thể hiện trong thuật toán sau.

Procedure *xauketiep; (cho trước xâu* $b_{n-1}b_{n-2}...b_1b_0 \neq 111...1$*)*

$i := 0$

while $b_i = 1$*;*

begin

$b_i = 0;$

$i := i + 1;$

end;

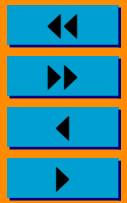
$b_i = 1;$

Phương pháp xây dựng tổ hợp chập k của n cũng theo một nguyên tắc tương tự. Mỗi tổ hợp chập k của n có thể biểu diễn bằng một xâu tăng và chúng được liệt kê theo thứ tự từ điển. Thuật toán sau đây cho tổ hợp liền sau tổ hợp $a_1a_2...a_k$ bằng cách sau. Trước hết tìm phần tử đầu tiên a_1 trong dãy đó cho kể từ phải qua trái sao cho $a_i \neq n-k+i$. Tiếp đó thay a_i bằng $a_i + 1$ và a_j bằng $a_i + j - i + 1$ với $j = i + 1, i + 2, ..., k$. Dễ chứng minh thuật toán này cho tổ hợp liền sau tổ hợp $a_1a_2...a_k$.

Ví dụ. Tìm tổ hợp chập 4 của 6 liền sau tổ hợp $\{1, 2, 5, 6\}$.

Lời giải. Với $n = 6, k = 4$, ta có $n - k + i \neq a_i$ thỏa mãn cho $i = 2$. Như vậy ta thay a_2 bởi $a_2 + 1 = 3$, và với $j > 2$ ta thay $a + j$ bởi $2 + j - 2 + 1 = j + 1$. Và ta thu được tổ hợp mới là $\{1, 3, 4, 5\}$ là tổ hợp liền sau tổ hợp đó cho $\{1, 2, 5, 6\}$.





Thuật toán này được viết thành chương trình như sau:

Procedure *tohopketiep; cho trước tổ hợp* $\{a_1, a_2, ..., a_k\} \neq \{n-k, ..., n\}$

while $a_i \neq n - k + i$ **do** $i := i - 1;$

$a_i := a_i + 1;$

for $j := i + 1$ **to** k

$a_j := a_i + j - i;$

Bài tập:

1. *Viết chương trình sinh tất cả các hoán vị của một tập hợp n phần tử.*

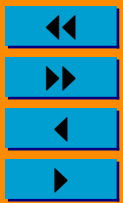
2. *Viết chương trình sinh tất cả các tổ hợp chập k của n.*





Chương 7

Lý thuyết đồ thị



Back
Close

Khái niệm đồ thị

Định nghĩa đồ thị

Cuốn sách giáo khoa đầu tiên: König, 1933, Leipzig Đức
Cuốn sách giáo khoa thứ hai: Berge, 1959, Paris.

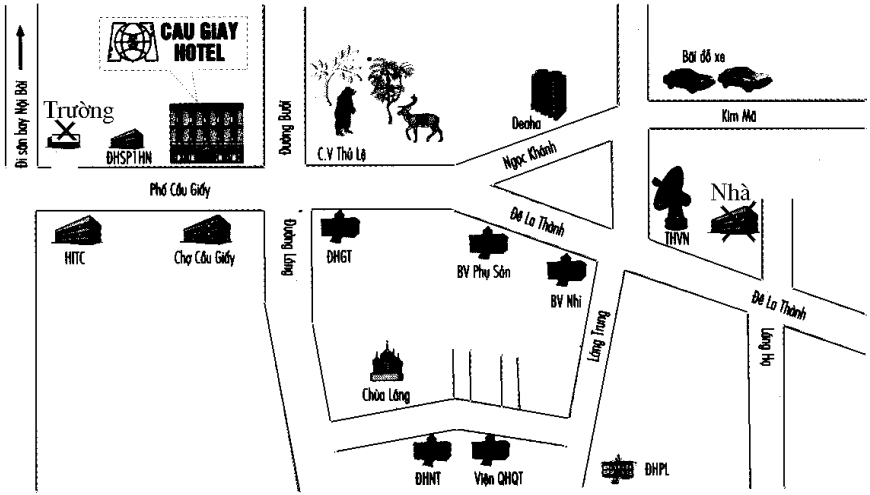






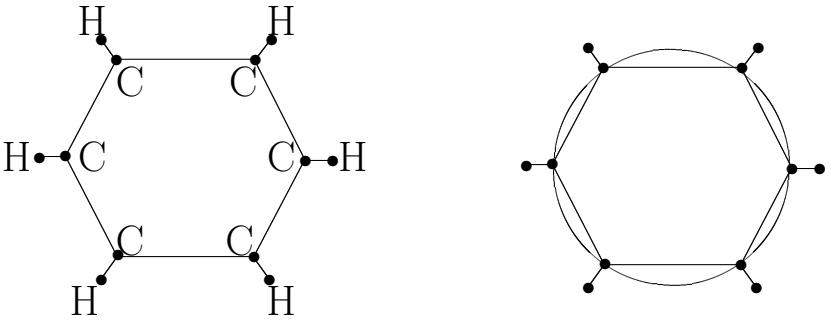

Back
Close

Ví dụ. Một bản đồ giao thông là một đồ thị (hình 7.1).



Hình 7.1: Bản đồ có thể mô hình thành một đồ thị.

Ví dụ. Biểu diễn công thức hóa học (hình 7.2) dưới dạng đồ thị.



Hình 7.2: Biểu diễn công thức hóa học.

Định nghĩa 7.1. *Một đồ thị $G = (V, E)$ được hiểu là một bộ hai tập hợp hữu hạn: Tập hợp đỉnh và tập hợp cạnh nối các đỉnh này với nhau.*

Phân loại đồ thị

Đồ thị được phân loại theo tính chất cạnh của chúng.

Một đồ thị được gọi là đồ thị vô hướng nếu tất cả các cạnh của chúng đều là cạnh vô hướng. Đỉnh a và b thì (a, b) là cạnh.

Đồ thị được gọi là đồ thị có hướng nếu tất cả các cạnh của nó là đều là cạnh có hướng. Đỉnh a và b thì $[a, b]$ là cạnh. Đỉnh a là đỉnh xuất phát, b là đỉnh kết thúc.

Đồ thị hỗn hợp là đồ thị có cả cạnh có hướng và cạnh không có hướng.

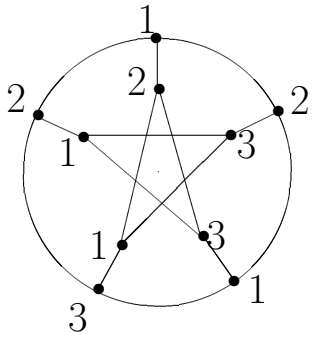
Đồ thị đơn không có khuyên và không có cạnh kép.

Đồ thị điểm là đồ thị có đúng một đỉnh và không có cạnh nào.

Đồ thị rỗng dùng để gọi một đồ thị không có đỉnh và cạnh nào cả.

Biểu diễn trên mặt phẳng của đồ thị và đẳng cấu

Hình 7.12: hai biểu diễn của đồ thị Petersen.

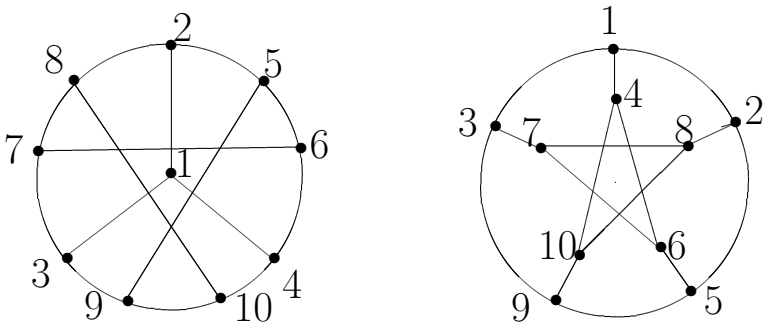


Hình 7.3: Hai biểu diễn phẳng của đồ thị Petersen.

Bài toán nhận biết hai biểu diễn phẳng khác nhau thực chất là hai biểu diễn phẳng khác nhau của cùng một đồ thị là bài toán khó. Nếu quả là hai biểu diễn phẳng khác nhau thực chất là biểu diễn phẳng của cùng một đồ thị, thì chúng ta phải cho các đỉnh của chúng tương ứng được với nhau (tức là thiết lập song ánh giữa hai tập đỉnh của hai biểu diễn phẳng này) sao cho sự kề nhau của hai đỉnh tương ứng trong hai biểu diễn phẳng là hoàn toàn như nhau. Chính xác về mặt toán học người ta định nghĩa như sau:

Định nghĩa 7.2. *Hai đồ thị $G_1 = (V_1, E_1)$ và $G_2 = (V_2, E_2)$ được coi là đẳng cấu với nhau (được coi chỉ là một đồ thị) nếu như tồn tại một*

song ánh $f : V_1 \to V_2$ *sao cho hai đỉnh* a *và* b *kề nhau trong đồ thị* G_1 *(tức là* $(a, b) \in E_1$*) khi và chỉ khi* $f(a)$ *và* $f(b)$ *kề nhau trong đồ thị* G_2 *(tức là khi* $(f(a), f(b)) \in E_2$*).*



Hình 7.4: Song ánh xác định sự đẳng cấu của hai đồ thị.

Chẳng hạn trong hình 7.12 ở trên, người ta có thể đánh số các đỉnh của hai đồ thị như trong hình 7.4 để thấy chúng đẳng cấu với nhau.

Dùng ma trận để biểu diễn đồ thị

Để đưa thông tin vào máy tính, người ta dùng ma trận để biểu diễn các thông tin của đồ thị. Bằng cách này có thể bảo đảm xây dựng lại được

đồ thị (đẳng cấu với đồ thị ban đầu) bằng ma trận đã cho. Có hai cách dùng ma trận biểu diễn đồ thị. Cho trước một đồ thị $G = (V, E)$.

Ta xây dựng một ma trận (a_{ij}) *có kích thước* $n \times n$ *trong đó* n *là số đỉnh của đồ thị và* ai_{ij} *là số cạnh nối đỉnh* i *với đỉnh* j. *Ma trận này được gọi là ma trận liền kề của đồ thị cho trước. Ma trận liền kề của đồ thị vô hướng luôn đối xứng. Trong trường hợp đồ thị là đồ thị đơn thì ma trận liền kề là ma trận 0-1. Chẳng hạn đồ thị Petersen có ma trận liền kề* A *là*

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



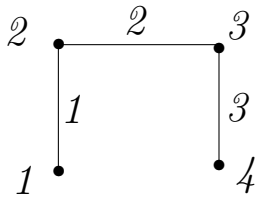





Back
Close

Còn một cách dùng ma trận biểu diễn đồ thị cho trước là dùng ma trận liên thuộc (a_{ij}) có kích thước $n \times m$ với n là số đỉnh và m là số cạnh của đồ thị.

Trước hết ta đánh số các đỉnh của đồ thị từ 1 tới n, còn các cạnh được đánh số từ 1 tới m. Ta gán cho a_{ij} giá trị bằng 0 nếu cạnh thứ j không chứa đỉnh i hoặc 1 nếu ngược lại. Như vậy đồ thị trong hình 7.5 sẽ có ma trận liên thuộc B là:



Hình 7.5: Ma trận liên thuộc.

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Bài tập:

1. *Có một trận đấu bóng bàn có 20 đấu thủ tham gia. Hai đấu thủ bất kì phải đấu với nhau đúng một hiệp. Điểm được cho mỗi hiệp là thắng được 1 điểm, thua 0 điểm (không có hòa). Hỏi có khi nào trận đấu kết thúc với kết quả là tất cả các đấu thủ đều bằng điểm nhau hay không?*

2. *Hãy biểu diễn bằng đồ thị quan hệ có ước chung lớn nhất là 2 của các cặp hai số trong tập hợp* $\{1, 2, 3, 4, 5, 6, 7\}$.

3. *Hãy biểu diễn bằng đồ thị quan hệ nguyên tố cùng nhau của các cặp 2 số tự nhiên trong tập hợp* $\{1, 2, 3, 4, 5, 6, 7\}$.







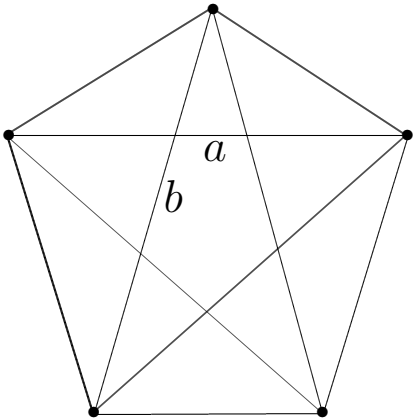

Các yếu tố cơ bản của đồ thị

Đồ thị con, đồ thị thành phần

Cho trước một đồ thị G với tập đỉnh X và tập cạnh E.

Định nghĩa 7.3. *Một đồ thị G' với tập đỉnh X' và tập cạnh E' được gọi là đồ thị con của đồ thị G nếu như X' là tập hợp con của X và E' là tập hợp con của E. Trong trường hợp X' là tập hợp con của X và E' là tập hợp tất các cạnh của G nối hai đỉnh của X', thì G' được gọi là đồ thị thành phần của G và còn được gọi là đồ thị sinh bởi tập đỉnh X'.*

Ví dụ. Đồ thị trong hình 7.6 có các cạnh được tô đậm là đồ thị con của đồ thị được biểu diễn trong hình. Nếu thêm vào nó hai cạnh a và b thì ta được một đồ thị thành phần của đồ thị đã cho. Đương nhiên đồ thị rỗng là đồ thị thành phần của mọi đồ thị cho trước.

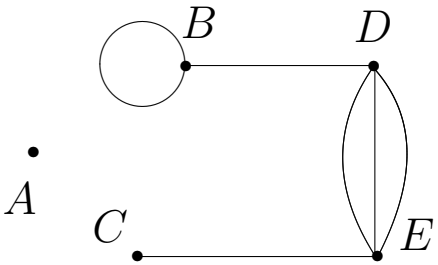


Hình 7.6: Đồ thị con không phải đồ thị thành phần của đồ thị đầy đủ K_5.

Bậc của đỉnh

Ta gọi bậc của một đỉnh là số cạnh xuất phát từ đỉnh đó (các khuyên được tính gấp đôi). Đương nhiên bậc của một đỉnh là một số nguyên không âm. Một đỉnh được gọi là đỉnh cô lập nếu nó không có cạnh nào cả, tức là khi đỉnh đó có bậc là 0. Đỉnh có bậc bằng 1 được gọi là đỉnh treo. Bậc của đỉnh v trong đồ thị G được ký hiệu là $d_G(v)$ hoặc $d(v)$.

Ví dụ. Đồ thị trong hình 7.7 có A là đỉnh cô lập, C là đỉnh treo. Đỉnh B có bậc là 3 và đỉnh D và đỉnh E có bậc bằng 4.



Hình 7.7: Bậc của đỉnh đồ thị.

Định lý sau tuy đơn giản, nhưng được sử dụng khá nhiều trong lý thuyết đồ thị.

Định lý 7.1. *Trong một đồ thị vô hướng G tùy ý tổng bậc của tất cả các đỉnh gấp đôi số cạnh của đồ thị.*

Chứng minh. Thật vậy, trong tổng bậc tất cả các đỉnh của một đồ thị thì mỗi cạnh được tính đúng hai lần bởi hai đỉnh của nó. Do đó tổng này là gấp đôi số cạnh của đồ thị. □

Hệ quả 7.2. *Số đỉnh bậc lẻ của một đồ thị vô hướng luôn là số chẵn.*

Hệ quả 7.3. *Đồ thị lẻ đỉnh luôn có một đỉnh với bậc là số chẵn.*

Định lý đã chứng minh trên là cơ sở quan trọng để giải nhiều bài toán chứng minh tồn tại những đỉnh có tính chất đặc biệt trong một đồ thị cho trước. Chẳng hạn, với những bài toán như sau ta có thể áp dụng định lý đã được chứng minh một cách hiển nhiên và trực tiếp:

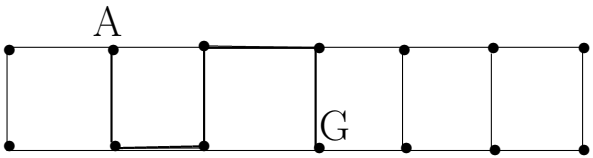
Ví dụ. Trong một mạng máy tính có một số đường kết nối mỗi máy tính với một số máy tính khác trong mạng. Một máy tính được gọi là máy tính lẻ nếu nó có một số lẻ đường kết nôi tới nó. Chứng minh rằng số các máy tính lẻ là một số chẵn.



Đường đi

Định nghĩa 7.4. *Cho trước một đồ thị G với tập đỉnh V và tập cạnh E. Một dãy cạnh dạng $e_i = (A_i, A_{i+1})$ với $i = 1, 2, ..., m$ được gọi là một đường đi nếu các đỉnh A_1, A_2, ..., A_m đôi một khác nhau. Đặc biệt một đường đi qua tất cả các đỉnh của đồ thị được gọi là đương Hamilton. Một đường đi thường được kí hiệu là*

$$H = (A_1, e_1, A_2, e_2, ..., e_m, A_{m+1}).$$



Hình 7.8: Đường đi trong đồ thị.

Trong trường hợp G là một đồ thị đơn thì ta có thể biểu diễn một đường đi bởi các đỉnh của chúng, chẳng hạn đường đi $H = (A_1, e_1, A_2, e_2, ..., e_m, A_{m+1})$ của ta ở trên được kí hiệu đơn giản là: $H = (A_1, A_2, ..., A_{m+1})$.

Số các cạnh của đường đi được gọi là độ dài của đường đi đã cho.

Ví dụ. Trong hình 7.8 có một con đường đi độ dài 5 (được tô đậm) đi từ đỉnh A tới đỉnh G.

- **Thuật toán tìm đường đi ngắn nhất**

  Thông thường các cạnh đồ thị không có độ dài. Để giải quyết những vấn đề của cuộc sống, đôi khi người ta gán cho cạnh e của đồ thị một độ dài (được ký hiệu $\omega(e)$) và thường gọi đó là trọng số của cạnh e. Thông thường trọng số của các cạnh đồ thị là những số không âm. Đồ thị mà các cạnh được gán trọng số được gọi là đồ thị trọng số. Ta có thể biểu diễn đồ thị có trọng số bằng ma trận (a_{ij}) kích thước $n \times n$ trong đó n là số đỉnh của đồ thị và a_{ij} là trọng số cạnh nối đỉnh i với đỉnh j. Ma trận này được gọi là ma trận trọng số của đồ thị cho trước.

  Bài toán tìm đường đi ngắn nhất từ đỉnh a tới đỉnh b của một đồ thị trọng số cho trước là bài toán tìm một đường đi nối a với b trong đồ thị đã cho, sao cho tổng trọng số các cạnh của con đường này nhỏ nhất có thể.

  Có nhiều thuật toán xác định đường đi ngắn nhất, trong đó quen biết nhất là thuật toán Dijkstra do nhà toán học người Hà lan tên là E. Dijkstra đề xuất năm 1959. Thuật toán đó được viết bằng ngôn








ngữ tựa Pascal như sau:

Procedure *Dijkstra; (G: đồ thị đơn liên thông có trọng số dương)*
{*đánh số các đỉnh của G:* $a = v_0$, v_1,..., $v_n = b$, *nối và gán thêm trọng số các cạnh không có trong đồ thị bởi* ∞}

For $i := 1$ *to* n

$L(v_i) := \infty$;

$L(a) := 0$; {*gán nhãn (đánh dấu) các đỉnh, bắt đầu với nhãn của* a *là* $L(a) = 0$, *còn các đỉnh khác có nhãn bằng* ∞, *tập* S *là tập hợp rỗng*}

$S := \emptyset$

While $b \notin S$

Begin

u:= *đỉnh không thuộc* S *có nhãn* $L(u)$ *nhỏ nhất;*

$S := S \cup \{u\}$;

For tất cả các đỉnh $v \notin S$

if $L(u) + (u, v) < L(v)$ *then* $L(v) := L(u) + (u, v)$ {*thêm vào* S *đỉnh có nhãn nhỏ nhất và sửa đổi nhãn của các đỉnh không thuộc* S}







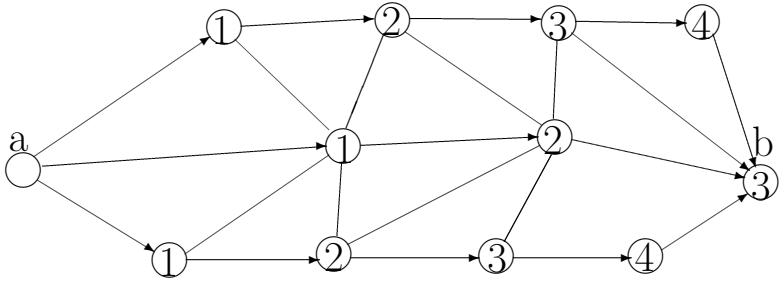




End; {$L(b)$ *là độ dài đường đi ngắn nhất từ đỉnh* a *đến đỉnh* b}

Thuật toán Dijkstra có độ phức tạp $O(n^2)$.

Ví dụ. Trong đồ thị ở hình 7.9, các cạnh đều có trọng số 1. Ta phải tìm đường đi ngắn nhất từ đỉnh a *tới đỉnh* b. *Mũi tên trên các cạnh chỉ để ký hiệu quá trình ta gán nhãn cho các đỉnh (là các số ghi trong các vòng tròn nhỏ tại các đỉnh). Con đường ngắn nhất nối* a *và* b *có tổng trọng số là 3.*



Hình 7.9: Đường đi trong đồ thị.

Liên thông

Trong phần này chúng ta làm quen với một khái niệm khá quan trọng của lý thuyết đồ thị là khái niệm liên thông.

Khi biểu diễn một đồ thị trên mặt phẳng, chúng ta đã thấy rằng có nhiều khi hình biểu diễn của chúng là những cụm tách rời nhau không được nối với nhau. Tương ứng với mỗi hình rời nhau như vậy là một đồ thị thành phần của đồ thị đã cho mà ta sẽ gọi là một thành phần liên thông của đồ thị cho trước.

Để chính xác hoá khái niệm liên thông, trước hết chúng ta nói hai đỉnh của một đồ thị cho trước là liên thông với nhau nếu có một dãy cạnh kế tiếp nối chúng với nhau trong đồ thị đã cho. Chẳng hạn, đồ thị được biểu diễn trong hình 7.8 của mục trước có đỉnh a và g liên thông với nhau. Tất nhiên là một đỉnh cho trước luôn được coi là liên thông với chính nó (được nối với chính nó bởi một dãy cạnh kế tiếp có độ dài 0).

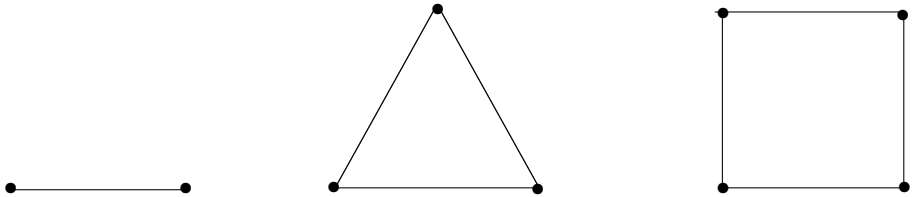
Định nghĩa 7.5. *Một đồ thị được gọi là liên thông nếu hai đỉnh bất kì của nó liên thông với nhau.*

Quan hệ liên thông của các đỉnh có những tính chất cơ bản sau:







Hình 7.10: Đồ thị có 3 thành phần liên thông.

a) Mỗi đỉnh a của đồ thị liên thông với chính nó.
b) Nếu a liên thông với b thì b liên thông với a.
c) Nếu a liên thông với b và b liên thông với c, thì a liên thông với c.

Thực chất quan hệ liên thông giữa các đỉnh là một quan hệ tương đương trong tập hợp các đỉnh của đồ thị. Quan hệ tương đương này chia tập đỉnh của đồ thị thành các lớp có hai tính chất sau:
1) Các đỉnh thuộc cùng một lớp thì liên thông với nhau.
2) Các đỉnh không cùng thuộc một lớp không liên thông với nhau.

Các lớp đỉnh này là đỉnh của các đồ thị thành phần liên thông trong đồ thị cho trước, được gọi là thành phần liên thông của đồ thị đã cho.

Ví dụ. Trong hình 7.10 chúng ta có một đồ thị với 3 thành phần liên thông.

Chỉ số liên thông

Ở phần trên chúng ta đã làm quen với khái niệm liên thông. Thế nhưng ngay trong các đồ thị liên thông cũng có sự phân biệt. Khái niệm sau đây cho ta một thước đo về sự liên thông của đồ thị.

Định nghĩa 7.6. *Cho trước một đồ thị G và một số tự nhiên $k \geq 2$, ta nói G là một đồ thị k-liên thông (đỉnh), nếu như G là một đồ thị liên thông và nếu như bỏ đi một số $t < k$ đỉnh tùy ý, đồ thị thu được vẫn là một đồ thị liên thông.*

Ví dụ. Đồ thị Petersen trong hình 7.12 là một đồ thị 3-liên thông, vì nó là một đồ thị liên thông và chỉ khi bỏ đi tới 3 đỉnh ta mới có thể thu được một đồ thị không liên thông. Còn khi bỏ đi 2 đỉnh tùy ý, đồ thị Petersen vẫn còn liên thông.

Tương tự khái niệm liên thông đỉnh, ta gọi một đồ thị liên thông là đồ thị k-liên thông cạnh nếu như bỏ đi ít hơn k cạnh, từ đồ thị ban đầu ta vẫn thu được một đồ thị liên thông. Khái niệm liên thông cạnh ít được nghiên cứu tới hơn khái niệm liên thông đỉnh.

Thông thường ta biểu kí hiệu số thành phần liên thông của một đồ thị G cho trước bởi $\omega(G)$. Nếu G là một đồ thị có n đỉnh rời nhau, thì

$\omega(G) = n$.

Định nghĩa 7.7. *Cho trước một đồ thị G. Số tự nhiên bé nhất k thỏa mãn điều kiện: G là đồ thị k-liên thông (đỉnh) nhưng không $k+1$ liên thông (đỉnh), được gọi là chỉ số liên thông (đỉnh) của đồ thị G.*

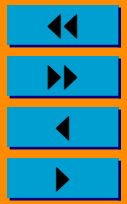
Chỉ số liên thông của đồ thị G thường được ký hiệu bởi $\kappa(G)$.
Ví dụ. Đồ thị Petersen trong hình 7.12 có chỉ số liên thông là 3.

Tương tự như vậy, ta có thể định nghĩa chỉ số liên thông cạnh của một đồ thị G cho trước, là số nhỏ nhất k sao cho G là một đồ thị liên thông cạnh mà không $k+1$ liên thông cạnh. Đồ thị Petersen trong hình 7.12 có chỉ số liên thông cạnh là 3.

Chu trình của đồ thị

Khi định nghĩa đường đi nối hai đỉnh a và b của một đồ thị, ta luôn giả thiết rằng các đỉnh a và b này phải khác nhau. Trong trường hợp a và b được nối với nhau bởi một con đường, thì khi thêm cạnh mới nối a với b vào đồ thị đã cho, ta thu được một chu trình. Như vậy chu trình là một dãy cạnh kế tiếp khép kín sao cho mỗi đỉnh của đồ thị được đi qua không quá một lần.





Chu trình được kí hiệu bởi việc đưa ra các cạnh và các đỉnh liên tiếp nhau trên chu trình. Chẳng hạn, nếu chu trình C đi qua các đỉnh p_1, p_2, .. ., p_k và các cạnh e_1, e_2, .. . , e_k thì ta viết

$$C = (p_1, e_1, p_2, e_2, ..., p_k, e_k, p_1).$$

Trong trường hợp đồ thị là một đồ thị đơn, thì thay vì viết rõ các cạnh và các đỉnh, chu trình được xác định duy nhất qua việc gọi tên các đỉnh nó đi qua. Chẳng hạn, chu trình C đề cập ở trên có thể viết thành:

$$C = (p_1, p_2, \ldots, p_k, p_1).$$

Số cạnh của chu trình được gọi là độ dài của chu trình và thông thường hay được kí hiệu bởi $\ell(C)$. Một khuyên lập thành một chu trình có độ dài 1. Dễ thấy rằng một đồ thị cho trước chỉ có chu trình có độ dài 2 nếu như nó có cạnh kép. Trong một đồ thị đơn mỗi chu trình có độ dài ít nhất là 3. Một đồ thị không đơn hiển nhiên luôn có ít nhất một chu trình có độ dài 1 hoặc 2. Trong đồ thị đơn không phải lúc nào ta cũng có thể tìm thấy một chu trình. Chẳng hạn trong các cây là các đồ thị ta sẽ nói tới ở chương sau.

Bài tập:



1. *Trong một kì nghỉ hè, có 7 người bạn đi nghỉ mát ở xa. Họ hứa với nhau rằng trong suốt kì nghỉ mỗi người phải viết thư cho đúng ba người trong số họ. Chứng minh rằng có một người không viết thư trả lời cho người viết cho mình.*

2. *Tồn tại hay không một đồ thị đơn vô hướng với 6 đỉnh và bậc của các đỉnh lần lượt là*
   a) 2, 3, 3, 3, 4, 4.
   b) 2, 3, 3, 4, 4, 4.

3. *Lập một đồ thị có mỗi đỉnh tương ứng với một tập con của tập con của tập hợp* $\{1, 2, 3, ..., 2001\}$ *và hai đỉnh được nối với nhau bởi một cạnh nếu hai tập con tương ứng với chúng không có điểm chung. Hãy tính số cạnh và số đỉnh của đồ thị thu được.*

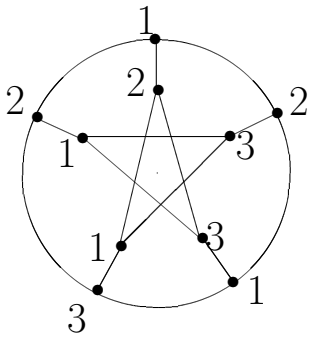
4. *Chứng minh rằng một đồ thị* G *không liên thông khi và chỉ khi nó có ít nhất 2 thành phần liên thông.*

5. *Hãy kiểm tra trong đồ thị Petersen (hình 7.12) có những chu trình loại độ dài bao nhiêu!*

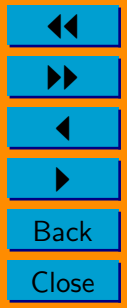


Định nghĩa 7.8. *Sắc số là số nhỏ nhất các mầu có thể dùng để tô màu các đỉnh của đồ thị G sao cho hai đỉnh kề nhau bất kỳ không cùng màu, gọi là tô màu ổn định, ký hiệu là $\chi(G)$.*

Ví dụ. Sắc số của đồ thị Petersen là 3. Ta có thể tô màu các đỉnh đồ thị 3 màu như hình dưới và có thể thấy không có cách tô 2 màu đồ thị một cách ổn đinh.



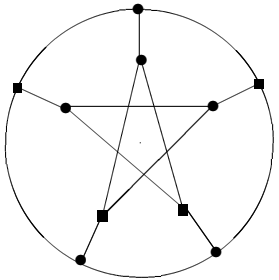
Hình 7.11: Sắc số của đồ thị Petersen.



Chỉ số ổn định trong

Định nghĩa 7.9. *Chỉ số ổn định trong là số lớn nhất các đỉnh đôi một không kề nhau, ký hiệu là* $\alpha(G)$.

Ví dụ. Chỉ số của đồ thị Petersen là 4, là 4 đỉnh hình vuông trong hình sau. Có thể thấy dễ dàng không tồn tại 5 đỉnh đôi một không kề nhau như vậy.

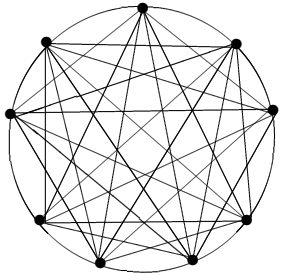


Hình 7.12: Chỉ số của đồ thị Petersen là 4.



Một số loại đồ thị đơn vô hướng

Đồ thị đầy đủ



Hình 7.13: Đồ thị đầy đủ 9 đỉnh.

Một đồ thị đơn vô hướng với n đỉnh được gọi là đồ thị đầy đủ nếu giữa hai đỉnh bất kì của nó có đúng một cạnh nối. Trong hình 7.13 ta có một đồ thị đầy đủ có 9 đỉnh. Nếu số đỉnh n của đồ thị cho trước thì chỉ có đúng một đồ thị đầy đủ với n đỉnh mà thôi. Do đó, người ta vẫn kí hiệu đồ thị đầy đủ n đỉnh là K_n. Với kí hiệu này, thì đồ thị đầy đủ trong hình 7.13 là đồ thị K_9.

Trong đời sống chúng ta hay gặp các đồ thị đầy đủ. Chẳng hạn, đồ thị biểu diễn quan hệ quen biết của các học sinh trong một lớp học là một đồ thị đầy đủ, vì đương nhiên học sinh của một lớp học bao giờ cũng quen biết nhau hết. Nhiều khi mô hình đồ thị đầy đủ khó nhận biết hơn, chẳng hạn khi phải xét một xếp đặt tối ưu mà một cặp đôi bất

kì đều phải có mặt.

Đồ thị đầy đủ K_n có C_n^2 cạnh.

Đồ thị đều

Một đồ thị đơn vô hướng G được gọi là đồ thị đều bậc t nếu như mỗi đỉnh của đồ thị G có bậc là t. Đồ thị Petersen là đồ thị đều bậc 3. Đặc biệt một đồ thị đầy đủ với n đỉnh là một đồ thị đều bậc $n-1$. Một đồ thị chỉ có các đỉnh rời được gọi là đồ thị đều bậc 0. Đồ thị đầy đủ K_2 là đồ thị liên thông đều bậc 1 duy nhất. Mỗi đồ thị đều liên thông bậc 2 là một chu trình. Việc xây dựng đồ thị đều bậc 3 thì khó khăn hơn. Rõ ràng một đồ thị với 2 đỉnh và 3 cạnh kép nối các đỉnh này với nhau là một đồ thị đều bậc 3. Cho trước một số tự nhiên $m \geq 1$, thì hợp của m đồ thị đều bậc 3 như vậy là một đồ thị đều bậc 3 với $2m$ đỉnh. Rõ ràng không tồn tại đơn đồ thị đều bậc 3 với 2 đỉnh. Đồ thị Petersen (hình 7.12, trang 166) là đồ thị đều bậc 3. Đồ thị đầy đủ K_4 là đồ thị đơn và đều bậc 3 duy nhất với đúng 4 đỉnh.

Những tính chất sau đây là những tính chất cơ bản của các đồ thị đều:








Back
Close

Định lý 7.5. *Số đỉnh của đồ thị đều bậc lẻ G luôn là một số chẵn.*

Định lý này được suy ra một cách hiển nhiên từ bổ đề 7.2 là số đỉnh bậc lẻ của một đồ thị luôn là số chẵn.

Ngoài ra chúng ta cũng dễ chứng minh định lý sau:

Định lý 7.6. *Cho trước G là một đồ thị đều bậc g với n đỉnh và m cạnh. Khi đó ta có $n.g = 2m$.*

Định lý được chứng minh dễ dàng nhờ định lý 7.1 là tổng các bậc của các đỉnh của một đồ thị bằng đúng hai lần số cạnh.

Đồ thị lưỡng phân

Đồ thị lưỡng phân là đồ thị $G = (V; E)$ mà tập đỉnh V có thể phân hoạch thành hai tập hợp X, Y sao cho tập cạnh E chỉ gồm các cạnh nối hai đỉnh không cùng một tập hợp.

Ta có thể thấy dễ dàng hai tính chất sau đây của đồ thị lưỡng phân:
1) Mỗi đồ thị con của đồ thị lưỡng phân là một đồ thị lưỡng phân.
2) Đồ thị lưỡng phân không có khuyên.

Ta có thể tìm thấy trong cuộc sống có rất nhiều ví dụ về đồ thị lưỡng phân. Chẳng hạn đồ thị biểu diễn quan hệ yêu đương khác giới là đồ

thị lưỡng phân. Một tính chất cơ bản để nhận biết đồ thị lưỡng phân là định lý sau đây:

Định lý 7.7. *Một đồ thị G là đồ thị lưỡng phân khi và chỉ khi mọi chu trình của nó có độ dài chẵn.*

Chứng minh. Giả sử $G = (X, Y; E)$ là một đồ thị lưỡng phân. Khi đó dọc theo chu trình bất kì của G các đỉnh thuộc tập X và tập hợp Y lần lượt kế tiếp nhau. Do đó, khi trở về đỉnh xuất phát đầu tiên, ta phải đi qua một số chẵn các đỉnh, và do đó số cạnh (bằng số đỉnh) của chu trình là một số chẵn.

Đảo lại, giả sử rằng G là một đồ thị có tính chất là tất cả các chu trình của G đều có độ dài chẵn. Ta sẽ chứng minh rằng tất cả các thành phần liên thông của G đều là các đồ thị lưỡng phân, và do đó G cũng là đồ thị lưỡng phân.

Thật vậy, giả sử rằng G_1 là một thành phần liên thông của G và P_0 là một đỉnh của đồ thị G_1. Với mỗi đỉnh P của đồ thị G_1 ta chọn một đường đi W nối đỉnh P_0với đỉnh P. Nếu đường đi W có độ dài (số cạnh của W) chẵn thì đỉnh P thuộc tập X, còn nếu đường đi Wcó độ dài lẻ thì đỉnh P được lấy vào tập hợp Y. Sự phân loại các đỉnh của đồ thị G_1 không phụ thuộc vào cách chọn đường đi W. Thật vậy, nếu có hai con đường W với





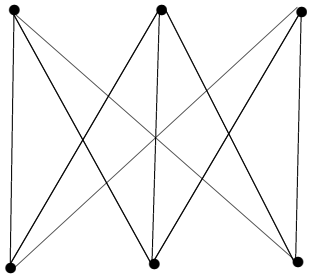



Back
Close

độ dài chẵn và con đường đi W' với độ dài lẻ nối đỉnh P_0 với đỉnh P thì đồ thị G_1 sẽ có chu trình với độ dài lẻ, mâu thuẫn với giả thiết ban đầu là đồ thị G chỉ có chu trình với độ dài chẵn.

Với cách thiết lập tập hợp X và Y này, các đỉnh của đồ thị G_1 hoặc thuộc tập hợp X hoặc thuộc tập hợp Y. Bây giờ ta chứng minh rằng chỉ có các cạnh nối các đỉnh không cùng một tập hợp với nhau mà thôi. Thật vậy, giả sử rằng có hai đỉnh P và Q kề nhau trong đồ thị G_1, thì chúng không thể cùng thuộc một tập hợp X hoặc Y, nếu không từ P_0 ta có thể đi tới đỉnh P rồi tới đỉnh Q bởi cạnh (P, Q) và trở về đỉnh P_0 với một đường đi lẻ cạnh, là điều không thể xảy ra trong đồ thị G, do đồ thị G chỉ có chu trình với số chẵn cạnh mà thôi. Như vậy đồ thị G là đồ thị lưỡng phân với hai tập đỉnh X và Y, và với tập cạnh E mà ta có thể viết như đã qui định $G = (X, Y; E)$. □

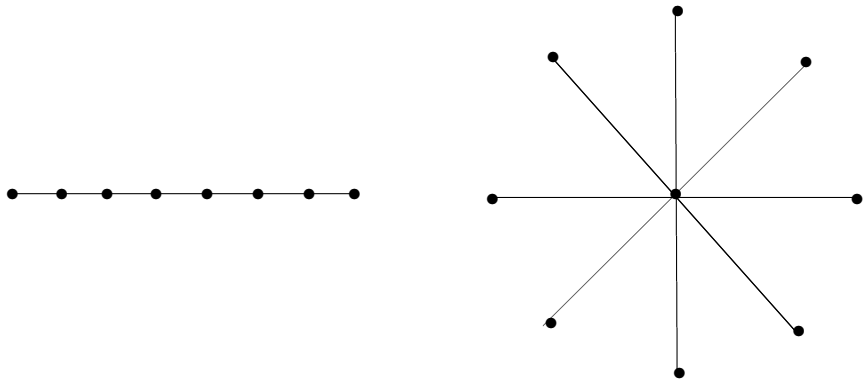
Nếu trong đồ thị lưỡng phân $G = (X, Y; E)$ giữa hai đỉnh bất kì không cùng trong một lớp đỉnh luôn có đúng một cạnh nối, thì ta gọi $G = (X, Y; E)$ là đồ thị lưỡng phân đầy đủ và ký hiệu nó bởi $K_{m,n}$, với n là số đỉnh của X và m là số đỉnh của Y. Trên hình 7.14 ta có đồ thị đầy đủ $K_{3,3}$.



Hình 7.14: Đồ thị đầy đủ $K_{3,3}$.

Cây

Một cây là một đồ thị đơn vô hướng liên thông không có chu trình với ít nhất một đỉnh. Các đồ thị loại này đã được nhiều nhà khoa học và nhiều nhà toán học nghiên cứu từ rất sớm do những ứng dụng của nó trong lý thuyết mạng điện. Hình 7.15 biểu diễn 2 cây khác nhau. Nếu quan sát kỹ các biểu diễn của cây, ta có thể nhận thấy một cây bất kì với ít nhất 2 đỉnh luôn có những đỉnh có bậc đúng bằng 1. Những đỉnh này được gọi là đỉnh treo và chúng đóng một vai trò hết sức quan trọng trong lý thuyết về cây. Trước hết, ta chứng minh sự tồn tại của các đỉnh treo trong cây có ít nhất 2 đỉnh.



Hình 7.15: Hai cây.

Định lý 7.8. *Một cây bất kì với ít nhất hai đỉnh có ít nhất hai đỉnh treo.*

Chứng minh. Trong cây chỉ có hữu hạn con đường. Ta xét $W = (P_1, P_2, ..., P_k)$ là một con đường có nhiều cạnh nhất. Khi đó ta có $k > 1$, do cây phải có ít nhất hai đỉnh.

Hai đỉnh cuối của W, P_1 và P_k, rõ ràng là hai đỉnh treo. Giả sử ngược lại là P_1 không phải là đỉnh treo thì đỉnh P_1 được nối với đỉnh $Q \neq P_2$ nào đó.

Khi đó đỉnh $Q = P_i$ với $i > 2$ nào đó, do W có nhiều cạnh nhất có thể.

Nhưng khi đó ta sẽ thu được một chu trình $K = (P_1, P_2, ..., P_i, P_1)$, trái với giả thiết ban đầu là G không có chu trình. □

Dựa trên sự tồn tại của các đỉnh treo, ta có thể xác định được số cạnh của cây thông qua định lý sau đây.

Định lý 7.9. *Một cây có n đỉnh có đúng $n - 1$ cạnh.*

Chứng minh. Chứng minh bằng quy nạp theo số đỉnh n của cây.

Với $n = 1$ rõ ràng cây với 1 đỉnh không có cạnh nào cả.

Giả sử một cây tùy ý với n đỉnh có đúng $n - 1$ cạnh.

Xét G là một cây có $n + 1$ đỉnh tùy ý. Theo định lý 7.9, G có ít nhất một đỉnh treo P nào đó. Xét đồ thị $G - \{P\}$.

Vì đỉnh P là đỉnh treo, nên đồ thị $G - \{P\}$ là một đồ thị liên thông. Giả sử ngược lại là đồ thị $G - \{P\}$ có ít nhất hai thành phần liên thông G_1 và G_2 nào đó. Do G là đồ thị liên thông, cho nên có một con đường G_1 với G_2 trong G. Rõ ràng con đường này phải đi qua đỉnh P và nhận đỉnh P làm đỉnh trong của nó. Vậy P có bậc ít nhất là 2, mâu thuẫn với giả thiết là đỉnh P là đỉnh treo trong đồ thị G.

Do G không có chu trình, cho nên đồ thị $G - \{P\}$ không có chu trình. Tóm lại là đồ thị $G - \{P\}$ là một cây có n đỉnh. Theo giả thiết quy nạp



thì đồ thị $G - \{P\}$ có đúng $n-1$ cạnh. Suy ra đồ thị G có đúng n cạnh, vì bậc của P trong đồ thị G bằng 1. □

Định lý 7.10. *Cho trước một đồ thị $G = (V, E)$, khi đó sáu tính chất sau là tương đương*

1) G là một cây (G là đồ thị liên thông và không có chu trình).

2) G không có chu trình và có $n-1$ cạnh.

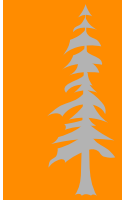
3) G liên thông và có $n-1$ cạnh.

4) G không có chu trình và nếu thêm vào một cạnh nối 2 đỉnh không kề nhau thì G xuất hiện duy nhất một chu trình.

5) G liên thông và nếu bỏ đi một cạnh tùy ý thì đồ thị nhận được sẽ không liên thông.

6) Mỗi cặp đỉnh trong G được nối với nhau bằng một đường duy nhất.

Chứng minh. Ta chứng minh lần lượt 1) $\Rightarrow$ 2) $\Rightarrow$ 3) $\Rightarrow$ 4) $\Rightarrow$ 5) $\Rightarrow$ 6) $\Rightarrow$ 1) 1) $\Rightarrow$ 2) Cho G là đồ thị liên thông và không có chu trình. Ta phải chứng minh 2). Điều này hiển nhiên đúng theo định lý đã được chứng minh về số cạnh của cây.








2) $\Rightarrow$ 3) Cho trước G không có chu trình và có $n-1$ cạnh, ta phải chứng minh 3), tức là chỉ còn cần phải chứng minh G liên thông. Thật vậy nếu G có p thành phần liên thông, thì mỗi thành phần liên thông là một cây và số cạnh của G theo định lý về số cạnh của cây sẽ là $|V|-p$ (lấy tổng các cạnh của các cây thành phần lại). Do số cạnh của G được cho biết là $n-1$, nên ta có $|V|-p=n-1$, vậy $p=1$ hay nói cách khác là G là đồ thị liên thông.

3) $\Rightarrow$ 4) Cho trước G liên thông và có $n-1$ cạnh, ta phải chứng tỏ rằng nếu thêm một cạnh nối 2 đỉnh không kề nhau bất kỳ trong G thì đồ thị thu được sẽ có chu trình. Trước hết ta thấy G là cây, vì nếu không có thể lần lượt bỏ bớt các cạnh khỏi G sao cho đồ thị thu được là đồ thị liên thông, và đến lúc nào đó không còn có thể bỏ được cạnh nào nữa, lúc đó G không còn chu trình (nếu không có thể bỏ bất kỳ cạnh nào khác trên chu trình vẫn không ảnh hưởng tính liên thông của đồ thị thu được). Đồ thị thu được sau cùng là cây, và sẽ có $n-1$ cạnh theo định lý về số cạnh của đồ thị. Suy ra số cạnh bỏ đi là 0, tức là G là cây và không có chu trình. Sử dụng định lý về đường đi giữa 2 đỉnh tùy ý x và y của G, ta có chu trình khi thêm cạnh nối 2 đỉnh không kề nhau này.

4) $\Rightarrow$ 5) Cho trước G không có chu trình và biết rằng nếu thêm vào một











cạnh nối 2 đỉnh không kề nhau thì G xuất hiện duy nhất một chu trình, ta phải chứng minh G liên thông và bỏ đi cạnh tùy ý thì G không còn liên thông.

Giả sử ngược lại G không liên thông, tức là tồn tại cặp đỉnh x, y trong G mà không có đường nào nối x với y. Khi đó nối x và y bởi 1 cạnh, đồ thị nhận được vẫn không có chu trình điều này mâu thuẫn với 4). Vậy G là liên thông. Nếu bỏ đi 1 cạnh trong G mà đồ thị vẫn liên thông thì nếu khôi phục lại cạnh này đồ thị sẽ có chu trình. Điều này mâu thuẫn với 4). Vậy ta có 5).

5) $\Rightarrow$ 6) Cho trước đồ thị G liên thông và bỏ đi cạnh tùy ý thì G không còn liên thông, ta phải chứng minh rằng mỗi cặp đỉnh trong G được nối với nhau bằng một đường duy nhất. Thật vậy, giả sử ngược lại, nếu trong G có tồn tại cặp đỉnh x, y không nối với nhau bằng đường nào cả, chứng tỏ G không liên thông mâu thuẫn với 5). Vậy mỗi cặp đỉnh đều có đường đi nối với nhau. Đường nối đó là duy nhất vì nếu có nhiều hơn, thì sau khi bỏ đi 1 cạnh trên chu trình xuất hiện bởi 2 đường nối 2 đỉnh này thì đồ thị vẫn liên thông, trái với 5).

6) $\Rightarrow$ 1) Cho trước một đồ thị G với tính chất là mỗi cặp đỉnh trong G được nối với nhau bằng một đường duy nhất, ta phải chứng minh G là cây.



Do mỗi cặp đỉnh nối với nhau bởi một đường nên G là liên thông. Giả sử G có chu trình thì xét cặp đỉnh x, y trên chu trình đó. Khi đó x, y có 2 cặp đường nối với nhau, mâu thuẫn với 6). □

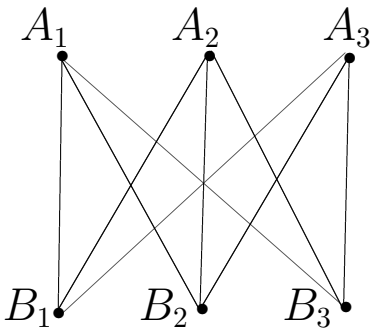
Cây có rất nhiều ứng dụng trong đời sống và khoa học. Những ứng dụng về cây sẽ được giới thiệu riêng trong một mục sau.

Đồ thị phẳng

Đôi khi người ta còn muốn biểu diễn đồ thị trên mặt phẳng sao cho không có 2 cạnh được biểu diễn nào cắt nhau. Một trong những vấn đề quan trọng của lý thuyết đồ thị là vấn đề nghiên cứu câu hỏi làm sao có thể biểu diễn đồ thị trên mặt phẳng sao cho không có hai đường biểu diễn cạnh nào của chúng cắt nhau. Không phải đồ thị nào cũng có thể biểu diễn như vậy, và những đồ thị có thể biểu diễn được thỏa mãn điều kiện đó được gọi là đồ thị phẳng. Ngay từ những năm 1930 nhà toán học Kuratowski, người Ba lan, đã phân lớp được những lớp đồ thị như vậy.

Ví dụ. Có ba cái nhà và ba cái giếng. Mỗi nhà có ba con đường đi từ nó tới ba cái giếng. Hỏi có thể làm những con đường đi như vậy sao cho





Hình 7.16: Ba nhà và ba giếng nước.

không có hai con đường nào cắt nhau hay không?

Để giải bài toán này, chúng ta phải giả sử là các nhà là các điểm A_1, A_2, A_3 trên mặt phẳng và các giếng là các điểm B_1, B_2, B_3 nào đó. Các con đường đi là các đường (liên tục) nối các đỉnh A_i với các đỉnh B_i như trong hình 7.16. Bằng cách thiết lập mô hình này, chúng ta đã thiết lập một đồ thị có 6 đỉnh và 9 cạnh. Đây chính là đồ thị $K_{3,3}$ trong hình 7.14. Khi đó, câu hỏi của bài toán là liệu có những đường nối các điểm A_i tới các điểm B_j trên mặt phẳng sao cho không có hai con đường nào cắt nhau hay không?

Thuộc lớp đồ thị phẳng có cây, bụi (hợp của những cây không có đỉnh

chung) và đồ thị biểu diễn của các khối đa diện lồi. Cho lớp đồ thị biểu diễn của các khối đa diện lồi, nhà toán học Steinitz đã chứng minh:

Định lý 7.11 (Steinitz)**.** *Lớp các đồ thị biểu diễn các khối đa diện lồi chính là lớp các đồ thị phẳng và 3-liên thông.*

Các đồ thị quen biết nhất không thể biểu diễn phẳng được chính là đồ thị đầy đủ K_5 và đồ thị lưỡng phân đầy đủ $K_{3,3}$.

Nếu một đồ thị được biểu diễn phẳng sao cho không có hai cạnh nào của chúng cắt nhau, thì những phần mặt phẳng được chia ra bởi những cạnh của đồ thị được gọi là miền. Trong trường hợp đồ thị là đồ thị biểu diễn của một khối đa diện lồi thì miền của nó cũng tương ứng với các mặt của khối đa diện lồi đã cho. Một định lý cơ sở cho việc nghiên cứu các đồ thị phẳng là định lý của Euler:

Định lý 7.12. *Cho đồ thị biểu diễn khối đa diện lồi có n đỉnh, c cạnh và m miền, ta có đẳng thức $n + m = c + 2$.*

Ví dụ. Đồ thị K_5 không phải là đồ thị phẳng.
Chứng minh. Ta chứng minh định lý bằng phản chứng. Giả sử đồ thị K_5 là đồ thị phẳng. Khi đó gọi n là số đỉnh, c là số cạnh và m là số

miền của đồ thị phẳng K_5 thì ta có $n = 5$, $c = 10$ và do đó $m = 7$ theo đẳng thức $n + m = c + 2$.

Gọi S là tổng số các cạnh của các mặt. Do mỗi cạnh được tính lặp đúng 2 lần thì tổng S này bằng 2 lần số cạnh của đồ thị nên $S = 2c$. Vì $c = 10$, ta có $s = 20$. Mặt khác mỗi mặt có ít nhất 3 cạnh, và do $m = 7$, nên $S \geq 3.7 = 21$ là điều vô lý. Điều vô lý này chứng tỏ K_5 không phải là đồ thị phẳng.

Để chứng minh định lý tổng quát hơn định lý này, ta gọi một cạnh của đồ thị phẳng cho trước G là cầu nếu bỏ nó đi thì số thành phần liên thông của đồ thị tăng lên ít nhất 1 và là cạnh biên, nếu cạnh này không phải là cầu của đồ thị G. Ta thấy hiển nhiên là một cạnh e của một đồ thị phẳng G cho trước là một cạnh biên khi và chỉ khi nó là cạnh chung của hai miền khác nhau của đồ thị G.

Dễ thấy một cạnh bất kì của đồ thị phẳng chỉ có thể hoặc là cầu hoặc là cạnh biên mà thôi. Bây giờ ta chứng minh định lý tổng quát hơn định lý của Euler về quan hệ giữa số đỉnh, số cạnh và số miền của một đồ thị phẳng G với $\omega(G)$ thành phần liên thông :

Định lý 7.13. *Cho trước đồ thị G là một đồ thị phẳng với n đỉnh, c cạnh và m miền. Khi đó ta có $n + m = c + \omega(G) + 1$.*








Chứng minh. Ta chứng minh bằng quy nạp theo số cạnh của đồ thị. Khẳng định của định lý hiển nhiên đúng cho trường hợp $c = 0$, vì trong trường hợp này ta có $n = \omega(G)$ và $m = 1$.

Bây giờ ta giả sử rằng đẳng thức $n + m = c + \omega(G) + 1$ đúng cho mọi đồ thị phẳng G với c cạnh. Xét đồ thị phẳng G với $c + 1$ cạnh tùy ý. Chúng ta chọn một cạnh e trong đồ thị G và khảo sát đồ thị $G - e$, sinh ra khi bỏ cạnh e ra khỏi đồ thị G. Chỉ có hai trường hợp sau đây có thể xảy ra:

1) Cạnh e là một cầu của đồ thị G.

Trong trường hợp này đồ thị $G - e$ có đúng m miền, do cạnh e không là cạnh chung của hai miền nào cả, và rõ ràng chỉ có đúng c cạnh. Theo giả thiết quy nạp, ta có $n + m = c + \omega(G - e) + 1$.

Do $\omega(G - e) = \omega(G) + 1$, vì e là cầu của đồ thị G, ta có được:

$$n + m = (c + 1) + \omega(G) + 1,$$

là điều cần phải chứng minh.

2) Cạnh e là cạnh biên của đồ thị G.

Khi đó đồ thị $G - e$ có đúng $m - 1$ miền, do hai miền có cạnh chung e sẽ trở thành một miền duy nhất trong đồ thị $G - e$ với đúng c cạnh. Theo











giả thiết quy nạp, ta có

$$n + m - 1 = c + \omega(G - e) + 1.$$

Do $\omega(G - e) = \omega(G)$, vì e không phải là cầu của đồ thị G, ta có:

$$n + m - 1 = c + \omega(G) + 1,$$

và do đó $n + m = (c + 1) + \omega(G) + 1$, là điều cần phải chứng minh.

Trong cả hai trường hợp ta có $\emptyset + m = (c + 1) + \omega(G) + 1$. Định lý được chứng minh. □

Ta có thể thấy dễ dàng là định lý 7.12 của Euler là một trường hợp riêng của định lý 7.13 khi $\omega(G) = 1$.

Bài tập:

1. *Hãy tìm tất cả các khối đa diện đều mà tập đỉnh và tập cạnh lập thành một đồ thị lưỡng phân.*
2. *Một đồ thị đơn có n đỉnh và p thành phần liên thông. Chứng minh rằng đồ thị này có sẽ có* $n - p$ *cạnh nếu như nó không có chu trình.*
3. *Sử dụng định lý 7.12 để chứng minh đồ thị lưỡng phân đầy đủ* $K_{3,3}$ *không phải là đồ thị phẳng.*

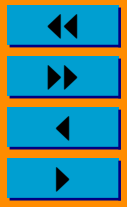




Đường một nét Euler và chu trình Hamilton

Mở đầu

Lý thuyết về đường một nét Euler khép kín, đường một nét Euler mở, đường và chu trình Hamilton đã có từ lâu và được nghiên cứu nhiều. Ta có thể bắt gặp nhiều bài toán trong thực tiễn mà có thể sử dụng các lý thuyết về chu trình, đường đi Euler và Hamilton để giải quyết. Ví dụ, sử dụng lý thuyết đường đi, chu trình Euler ta có thể tìm hành trình đường đi cho người phát thư, cho xe rửa đường... sao cho hành trình là tối ưu nhất. Hoặc là trong một hệ thống mạng, một máy đơn cần gửi 1 thông điệp đến tất cả các máy còn lại, thì đường truyền tin sẽ đi như thế nào để cho hiệu quả nhất? Một số bài toán lịch biểu ... có thể được giải quyết bằng cách vận dụng các lý thuyết chu trình và đường đi Hamilton.



Đường một nét Euler

Trong lý thuyết đồ thị có khái niệm đường một nét Euler. Khái niệm này hoàn toàn không phải là khái niệm đường trong đồ thị mà ta nói ở các phần trên. Đường một nét của Euler trong một đồ thị vô hướng (không nhất thiết phải là đồ thị đơn) $G = (V, E)$ *có* n *đỉnh và* m *cạnh được hiểu là một dãy chứa tất cả* m *cạnh của đồ thị và có dạng* $P_1, e_1, P_2, e_2, ..., P_m, e_m, P_{m+1}$ *sao cho cạnh* e_i *là cạnh nối 2 đỉnh* P_i *và* P_{i+1} *(thiếu điều kiện dãy cạnh này chứa tất cả* m *cạnh của đồ thị, thì nó được gọi là đường một nét mà thôi). Đường một nét Euler được gọi là đường một nét Euler khép kín nếu* $P_1 = P_{m+1}$ *và là đường một nét Euler mở nếu như* $P_1 \neq P_{m+1}$*. Bài toán đường một nét này khá quen thuộc với chúng ta với những đánh đố vẽ hình cái phong bì thư bằng một nét chẳng hạn. Bài toán đường một nét Euler bắt nguồn từ bài toán 7 cây cầu thành phố Königsberg (xem hình dưới) với thách đố làm sao có thể đi qua tất cả 7 cây cầu, mỗi cầu chỉ được đi qua đúng một lần. Euler đã giải quyết bài toán này bằng cách chứng minh được là không có cách nào có thể đi qua tất cả 7 cây cầu của thành phố Königsberg sao cho mỗi cầu chỉ được đi qua đúng một lần.*

Trước khi chứng minh định lý Euler, ta chứng minh bổ đề sau:

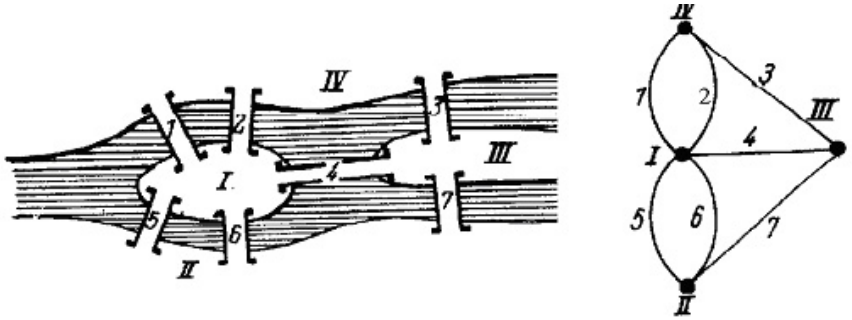












Hình 7.17: Bảy cây cầu thành Königsberg.

Nếu bậc của mỗi đỉnh x của một đồ thị G là số chẵn, thì từ mỗi đỉnh x của một đồ thị G đều tồn tại một đường một nét khép kín (không nhất thiết là đường một nét Euler) đi qua nó.

Chứng minh. Thật vậy, ta chỉ cần xuất phát từ x đi theo các cạnh của G sao cho mỗi cạnh chỉ đi qua đúng một lần. Lưu ý rằng vì mỗi đỉnh có số chẵn cạnh, và mỗi khi đi vào một đỉnh y không phải là x thì số cạnh xuất phát từ y đã được đi qua luôn là một số lẻ. Do bậc của các đỉnh là số chẵn, nên mỗi lần khi đi vào $y \neq x$ ta luôn tìm được một cạnh xuất phát từ y mà chưa được đi qua và có thể ra khỏi y qua cạnh đó. Từ đó suy ra đỉnh kết thúc quá trình đi này chỉ có thể là ở x mà thôi. Tóm lại từ mỗi đỉnh x

của đồ thị đã cho G luôn có một đường một nét khép kín P_x đi qua nó. □

Định lý 7.15. *Đồ thị vô hướng và liên thông $G = (V, E)$ có đường một nét Euler khép kín khi và chỉ khi bậc của tất cả các đỉnh trong đồ thị G là số chẵn.*

Chứng minh. Chứng minh được thực hiện theo hai chiều:
Điều kiện cần: Giả sử đồ thị G có đường một nét khép kín H. Ta cần chứng minh G là bậc của mỗi đỉnh $x \in V$ là một số chẵn.

Điều này là khá hiển nhiên, vì nếu đi trên H theo các cạnh của G, bắt đầu từ một cạnh nào đó rồi trở về cạnh đầu tiên, ta thấy dễ dàng rằng bậc của mỗi đỉnh đúng bằng $2k$, với k là số lần ta đi qua nó trên đường đi (do mỗi lần đi vào nó ta lại phải đi ra khỏi nó).
Điều kiện đủ: Giả sử G là đồ thị liên thông và mỗi đỉnh $x \in V$ đều có bậc là số chẵn, ta sẽ chứng minh trong đồ thị G tồn tại một đường một nét Euler. Từ bổ đề 7.14 ta suy ra là G có ít nhất một đường một nét khép kín.

Bây giờ gọi P là một đường một nét khép kín trong G và có nhiều cạnh nhất. Bây giờ ta sẽ chứng minh rằng P chính là đường một nét Euler trong đồ thị G.



Thật vậy, giả sử rằng P không phải là đường một nét Euler trong G. Khi đó bỏ tất cả các cạnh của P ra khỏi G, ta thu được một đồ thị G_1 có ít nhất một cạnh. Có thể thấy rằng tại mỗi đỉnh x của G đều có một số chẵn cạnh thuộc P (là $2k$ với k là số lần x xuất hiện trên P). Do bản thân x có bậc trong G là số chẵn, cho nên bậc của mỗi đỉnh x trong đồ thị thu được trong G_1 cũng là số chẵn. Do G là đồ thị liên thông, cho nên có ít nhất có một cạnh nào đó của G_1 nối một đỉnh x trên P với một đỉnh y của G_1. Theo nhận xét đã chứng minh ở trên, tồn tại một đường một nét khép kín P_x nào đó chứa x trong G_1. Lúc này, hợp của hai đường một nét P và P_x là một đường một nét trong G và có nhiều cạnh hơn G, là điều mâu thuẫn với giả thiết P là đường một nét có nhiều cạnh nhất trong G. Mâu thuẫn đó chứng tỏ P chứa hết các cạnh của G, tức là P là đường một nét Euler trong G. Định lý được chứng minh. □

Định lý 7.16. *Điều kiện cần và đủ để đồ thị một đồ thị liên thông G có đường một nét Euler mở là số đỉnh bậc lẻ trong đồ thị là 2.*

Chứng minh. Ta chứng minh điều kiện cần trước:
Điều kiện cần. Nếu G là một đồ thị có đường một nét Euler mở, thì khi đi từ đỉnh đầu tới đỉnh cuối và dọc theo các cạnh của đường một nét Euler mở này, thì chỉ có đỉnh đầu và đỉnh cuối có bậc là số lẻ, còn tại các đỉnh

còn lại, số cạnh của đồ thị luôn là $2k$ với k là số lần xuất hiện của nó trên đường một nét Euler đã cho.
Điều kiện đủ. Cho trước một đồ thị vô hướng G với đúng 2 đỉnh bậc lẻ x và y. Khi đó ta thêm vào G một cạnh mới nối x và y (là cạnh kép nếu x và y đã được nối bởi một cạnh nào đó rồi). Khi đó theo định lý trên, ta có trong đồ thị thu được một đường một nét Euler khép kín. Bỏ đi cạnh thêm vào, ta có trong G đường một nét Euler mở. □

Qua cách chứng minh trên, ta thấy việc xác định đường một nét Euler mở được đưa về việc xác định đường một nét Euler khép kín. Thuật toán sau đây cho ta cách xác định đường một nét Euler khép kín.

THUẬT TOÁN: *đường một nét Euler khép kín.*

Procedure *Euler; (Đồ thị G liên thông, bậc của các đỉnh là chẵn)*

K := *đường một nét bắt đầu từ một đỉnh nào đó và quay trở lại đỉnh đó*

H := *đồ thị thu được từ G sau khi bỏ các cạnh của K*

while H *còn có cạnh*

BEGIN

$W :=$ *đường một nét khép kín trong* H *từ một đỉnh nằm trên* K

$K := K \cup W$

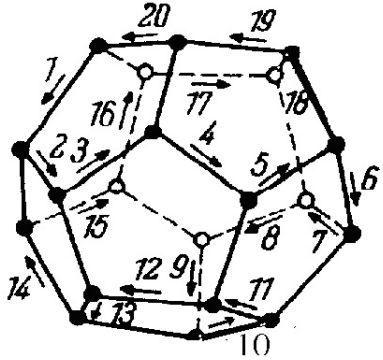
$H :=$ *thu được từ* H *sau khi loại bỏ các cạnh của* W

END. $\{K$ *là đường một nét Euler*$\}$

Chu trình Hamilton

Một chu trình được gọi là chu trình Hamilton nếu như nó đi qua tất cả các đỉnh của đồ thị, mỗi đỉnh đúng một lần. Tên gọi của nó xuất phát từ một trò chơi của nhà toán học nổi tiếng người Anh là R. Hamilton đòi hỏi phải chỉ ra một chu trình đi qua tất cả các đỉnh của đồ thị biểu diễn khối thập nhị diện đều (hình 7.18 với một chu trình Hamilton). Bài toán chu trình Hamilton cho đến nay vẫn chưa được giải quyết một cách thỏa đáng và trở thành một trong những động lực chính để phát triển lý thuyết đồ thị. Từ năm 1858, khi Hamilton nêu vấn đề tìm chu trình Hamilton trong đồ thị cho tới nay có rất nhiều bài toán về chu trình Hamilton được đưa ra mà đến nay vẫn chưa giải được. Trong số những giả thuyết sớm nhất về chu trình Hamilton được Tait đưa ra





Hình 7.18: Trò chơi đi vòng quanh thế giới của Hamilton.

năm 1884 là "Mọi đồ thị biểu diễn các khối đa diện lồi đều có chu trình Hamilton" bị bác bỏ bởi Tutte năm 1946. Trong năm 1931, Whiney đã chứng minh được kết quả đầu tiên về sự tồn tại chu trình Hamilton trong đồ thị phẳng có mỗi miền là một tam giác với điều kiện là 3 cạnh tùy ý bất kỳ của nó lập thành một tam giác chỉ khi tam giác đó là một miền của đồ thị. Tutte đã tổng quát kết quả này khi chứng minh mọi đồ thị phẳng 4-liên thông đỉnh đều có một chu trình Hamilton. Năm 1952, Dirac chứng minh kết quả đầu tiên một cách hình thức về sự tồn tại chu trình Hamilton với sự ràng buộc về bậc nhỏ nhất của các đỉnh.

Định lý 7.17. *Nếu đồ thị đơn vô hướng G có n 3 đỉnh và bậc của mỗi đỉnh $\geq \frac{n}{2}$ thì G có chu trình Hamilton.*

Chứng minh. Trong một hoán vị tròn các đỉnh của đồ thị

$$H = (P_1, P_2, ..., P_n, P_{n+1} = P_1),$$

mỗi cặp hai đỉnh liên tiếp nhau trong hoán vị được gọi là một lỗ hổng nếu như chúng là hai đỉnh không kề nhau trong đồ thị. Trong $(n-1)!$ hoán vị tròn có thể có, giả sử không mất tổng quát là H là hoán vị có ít lỗ hổng nhất.

Ta khẳng định H chính là một chu trình Hamilton của đồ thị đã cho. Thật vậy, giả sử H có ít nhất một lỗ hổng mà không mất tổng quát đó là lỗ hổng tạo nên bởi cặp (P_n, P_1). Như vậy, nếu P_1 kề với đỉnh P_i nào đó thì đỉnh P_n không kề với P_{i-1}, nếu không thì $H = (P_{i-1}, ...P_2, P_1, P_i, ..., P_n, P_{n+1} = P_{i-1})$ là một hoán vị có ít lỗ hổng hơn của H, trái với giả thiết là H có ít lỗ hổng nhất. Do P_1 kề với không ít hơn $\frac{n}{2}$ đỉnh nên P_n không kề với ít nhất $\frac{n}{2}$ đỉnh, tức là bậc của nó nhỏ $\frac{n}{2}$, trái với giả thiết của đầu bài. Do đó H phải là chu trình Hamilton trong G. □



Thuật toán liệt kê tất cả các chu trình Hamilton. *Thuật toán được xây dựng dựa trên cơ sở thuật toán quay lui cho phép liệt kê tất cả các chu trình Hamilton của đồ thị.*

Procedure *Hamilton(k);* $\{$*Liệt kê các chu trình Hamilton thu được bằng việc phát triển dãy đỉnh* $x[1], ..., x[k-1]$ *của đồ thị* $G = (V, E)$ *cho bởi danh sách kề* $Ke(v)$, $v \in V\}$

Begin

For *y Ke(x[k-1])* **do**

if *(k = n +1) and (y = v0)* **then** *GhiNhan(x[1],....,x[n],v0)*

Else if *ChuaXet[y]* **then**

Begin

x[k] := y;

ChuaXet[y] := False;

Hamilton(k+1);

ChuaXet[y] := True;











End;

End;

Begin

For $v \in V$ **do** *ChuaXet[v] := True;*

x[1] := v0; v0 là một đỉnh nào đó của đồ thị

ChuaXet[v0] := False;

Hamilton(2);

End.

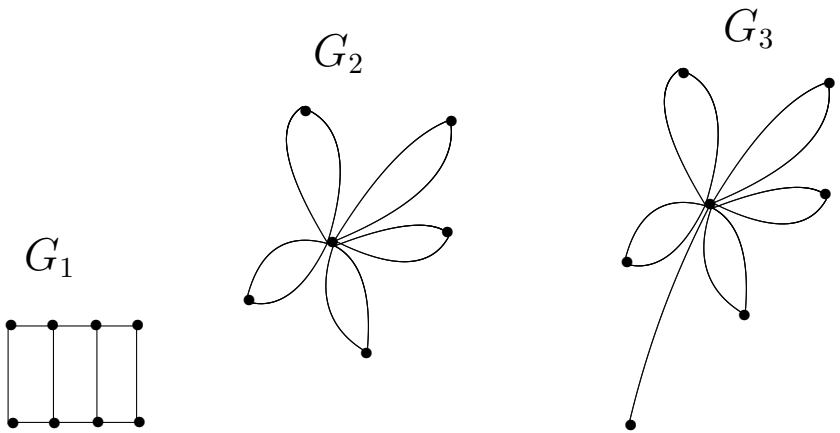
Bài tập:

1. *Nếu đồ thị lưỡng phân* $G = (X, Y; E)$ *có một chu trình hamilton thì các tập hợp* X *và* Y *có cùng một số đỉnh. Hãy chứng minh điều đó.*

2. *Trong các đồ thị trong hình 7.19 sau đây, đồ thị nào có đường một nét Euler. Giải thích và nói rõ loại đường một nét Euler (mở hay khép kín) nếu nó tồn tại.*





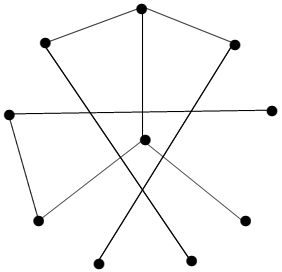


Hình 7.19: Tìm đường một nét Euler.

Các thuật toán về cây và ứng dụng

Cây khung và các thuật toán tìm kiếm

Định nghĩa 7.10. *Cho trước một đồ thị liên thông G. Một đồ thị con T của được gọi là cây khung (còn gọi là cây bao trùm) của G nếu:*
1. T là một cây,
2. T chứa tất cả các đỉnh của G.



Hình 7.20: Một cây khung của đồ thị Petersen (hình 7.12, trang 166).

Ví dụ. Cây khung hay thường được gặp khi chúng ta phải mắc một hệ thống các tuyến cung cấp nào đó cho n địa điểm cho trước. Thông

thường người ta phải tiến hành thăm dò để thiết lập một bản đồ các đường dẫn có thể. Hệ thống cung cấp sẽ là cây khung của đồ thị được thiết lập về các đường dẫn có thể.

Sau đây là hai thuật toán xác định cây khung trong một đồ thị cho trước.



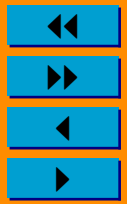
Thuật toán tìm kiếm theo chiều sâu

Bước 1: Lấy một đỉnh a bất kỳ làm gốc của cây khung

Bước 2: Xây dựng đường đi từ đỉnh này bằng cách lần lượt ghép thêm các cạnh vào, sao cho mỗi cạnh ghép mới sẽ nối đỉnh cuối cùng trên đường đi với đỉnh chưa thuộc đường đi. Bước này thực hiện cho tới khi nào không thể ghép thêm cạnh mới được nữa.

Bước 3: Nếu đường đi chứa tất cả các đỉnh của đồ thị G, thì đó chính là cây bao trùm cần tìm. Ngược lại thì thực hiện tiếp bước sau đây.

Bước 4: Quay lui lại đỉnh trước đỉnh cuối cùng của đường đi và xây dựng đường đi mới xuất phát từ đỉnh này. Nếu không xây dựng được đường đi mới từ đỉnh này thì lùi tiếp lại một đỉnh nữa trên đường đi và lặp lại quá trình xây dựng đường mới càng dài càng tốt.



Bước 5: Vì đồ thị là hữu hạn nên quá trình trên sau một số hữu hạn bước sẽ dừng lại và cho ta cây bao khung của đồ thị G đã cho.

Thuật toán được thể hiện với ngôn ngữ tựa Pascal như sau:

Procedure *Dfs(v: Integer);{Thu tuc tim cay bao trum theo chieu sau}*

Var u: Integer;

Begin

ChuaXet[v]:=False; Dinh v da duoc tham

For $u := 1$ **to** n **do**

If *(mG[v,u]<>0)* **and** *(ChuaXet[u])* **then**

Begin *Duyet nhung canh ke voi v*

$mT[v, u] := mG[v, u]$; *Bo sung vao tap canh cua cay*

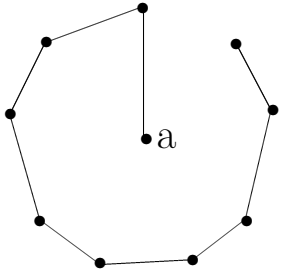
$mT[u, v] := mT[v, u]$;

Dfs(u);

End;

End;

Độ phức tạp của thuật toán tìm kiếm ưu tiên chiều sâu là $O(m)$, với m là số cạnh của đồ thị.



Hình 7.21: Cây khung của đồ thị Petersen theo tìm kiếm chiều sâu từ a.

Thuật toán tìm kiếm theo chiều rộng:

Bước 1: Chọn một đỉnh a làm gốc của cây

Bước 2: Ghép các cạnh liên thuộc với đỉnh này. Các đỉnh mới kề với gốc trong bước này đều nằm trên mức 1 của cây (với thứ tự tuỳ ý).

Bước 3: Tiếp tục với mỗi đỉnh ở mức 1, ta ghép các cạnh liên thuộc nó sao cho chúng không tạo nên chu trình. Các đỉnh kề được tạo ra ở bước này nằm trên mức 2 của cây.

Bước 4: Quá trình này sẽ dừng lại sau một số bước làm việc (do tập cạnh hữu hạn) và tất cả các đỉnh của đồ thị đều được ghép vào cây.

Thuật toán được thể hiện với ngôn ngữ tựa Pascal như sau:

Procedure *Bfs(v: Integer);{Tim cay bao trum cua do thi theo chieu rong}*

Var i, u, First, Last: Integer;

Begin

First := 1; *Last* := 1;

Queue[Last]:=v;

ChuaXet[v]:=False; Dinh v da duoc tham

While *First<=Last* **do**

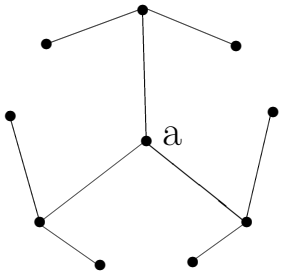
Begin {*Trong khi hang doi chua ron*}

u:=Queue[First]; Inc(First);

For *i* := 1 **to** *n* **do If** *(mG[u,i]<>0)* **and** *(ChuaXet[i])* **then**

```
        Begin Duyet nhung canh ke voi v
          mT[u,i]:=mG[u,i]; Bo sung vao tap canh cua cay
          mT[u,i]:=mT[i,u];
          ChuaXet[i]:=False;
          Inc(Last); Queue[Last]:=i;
        End;
    End;
End;
```

Độ phức tạp của thuật toán tìm kiếm ưu tiên chiều rộng là $O(m)$, với m là số cạnh của đồ thị.



Hình 7.22: Cây khung của đồ thị Petersen theo tìm kiếm chiều rộng từ a.

Cây khung nhỏ nhất

Cho trước một đồ thị, nhiều khi các cạnh của nó được gán trọng số (thường là các số không âm) thể hiện chiều dài của nó trong thực tế hoặc thời gian vận chuyển trên tuyến.... Một cây khung của đồ thị đã cho được gọi là cây khung nhỏ nhất nếu tổng trọng số các cạnh của nó là nhỏ nhất. Sau đây là hai thuật toán phổ biến nhất để tìm cây khung nhỏ nhất.

THUẬT TOÁN PRIM

Procedure *Prim; (G: đồ thị có n đỉnh và các cạnh có trọng số)*

T:= *đồ thị có duy nhất một cạnh là cạnh có trọng số nhỏ nhất*

for $i := 1$ **to** $n - 2$

begin

e:= *cạnh có trọng sô nhỏ nhất trong các cạnh không thuộc* T, *có đúng một đỉnh thuộc* T

$T := \{e\} \cup T$

end. {*T là cây khung nhỏ nhất*}

Độ phức tạp thuật toán Prim là $O(m.\log n)$ *với* n *là số đỉnh và* m *là số cạnh của đồ thị.*

THUẬT TOÁN KRUSKAL

procedure *Kruskal; (G: đồ thị có n đỉnh và các cạnh có trọng số)*

T: =*đồ thị rỗng*

for $i := 1$ **to** $n - 1$

begin

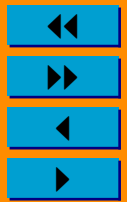
e:= *cạnh có trọng sô nhỏ nhất trong các cạnh không thuộc* T *và* $\{e\} \cup T$ *là đồ thị không có chu trình*

$T := e \cup T$

end. {*T là cây khung nhỏ nhất*}

Độ phức tạp của thuật toán Kruskal là $O(m.\log m)$ *với* m *là số cạnh của đồ thị. Như vậy khi đồ thị có số cạnh nhỏ thì nên sử dụng thuật toán Kruska.*





Lưu ý. *Trong trường hợp các cạnh có trọng số đôi một khác nhau thì cây khung nhỏ nhất được xác định duy nhất. Còn trong trường hợp có nhiều cạnh có trọng số bằng nhau thì có thể có nhiều cây khung tối ưu khác nhau.*

Cây nhị phân và ứng dụng

Trước hết ta nêu vài khái niệm mới về cây chưa được nhắc tới ở phần trên.

- **Gốc**: *Đối với một cây T bất kỳ có thể chọn 1 đỉnh nào đó làm gốc, một cây đã được chọn 1 đỉnh làm gốc thì được gọi là cây có gốc. Vậy một cây có thể tạo thành nhiều cây có gốc khác nhau.*

- *Quan hệ* **cha con**: *Giả sử a là gốc nếu có b, c kề với a thì b, c được gọi là con của a(hoặc gọi a là cha của b, c) tương tự nếu có d, e kề với b thì b là cha của chúng còn d, e là con của b (xem hình).*

- *Trong một cây tất cả các đỉnh là cha được gọi là các* **đỉnh trong**. *Các đỉnh không phải là đỉnh trong được gọi là* **lá** *(hay lá là đỉnh con không có con), trong cây chỉ có gốc là đỉnh duy nhất không phải là con.*



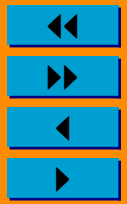
- **Bậc của đỉnh** *là số các con của nó,* **bậc của cây** *là bậc lớn nhất của đỉnh.*

- **Mức của cây:** *Mỗi đỉnh đều được gán bằng một mức, mức của gốc là 0, con của gốc có mức là 1. Nếu mức của cha là i thì mức của con sẽ là $i+1$. Một đỉnh x nào đó có mức bằng độ dài đường đi từ gốc đến x, mức cao nhất trong số các đỉnh được gọi là chiều cao của cây.*

Khi cây chưa có gốc, chưa phân chia thành các các mỗi quan hệ cha con, bậc, mức... thì cây là cây tự do còn khi được phân chia gọi là cây phân cấp.
Ví dụ. Cây T trong hình 7.23 vốn là một cây tự do, do chọn a làm gốc thì nó trở thành cây phân cấp T' có gốc. Với cây T' gốc a có bậc 2 và mức 0, đỉnh c có bậc là 3 và mức 1 ... Mức cao nhất là 3 ở các đỉnh là lá như i, j, k nên chiều cao của cây là $h(T') = 3$.

Định nghĩa 7.11. *Cây phân cấp T nếu mỗi đỉnh trong của nó có không quá m con thì T được gọi là cây m-phân. Đặc biệt nếu $m = 2$ thì cây được gọi là cây nhị phân, cây nhị phân rất quan trọng và có nhiều ứng dụng rộng rãi.*



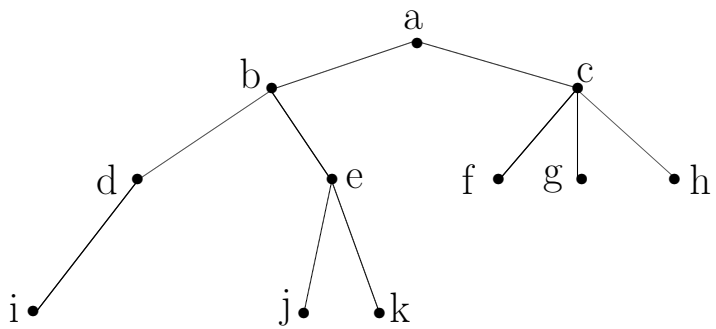


- *Cây đầy đủ: Cây m- phân T được gọi là cây m- phân đầy đủ nếu mỗi đỉnh trong đều có đúng m con.*

- *Cây cân đối: Xét cây m- phân có chiều cao h. Nếu các lá của cây có mức $h-1$ hoặc h thì cây là cây cân đối.*

Ví dụ. Cây T' trong hình 7.23 là cây 3 phân, không phải cây đầy đủ nhưng là cây cân đối.

Khi $m = 2$ thì cây của ta được gọi là cây nhị phân (hình 7.24). Sau đây ta nêu ra một ứng dụng của cây nhị phân.



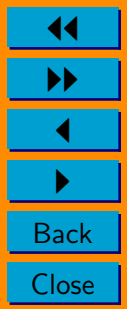
Hình 7.23: Cây tam phân.

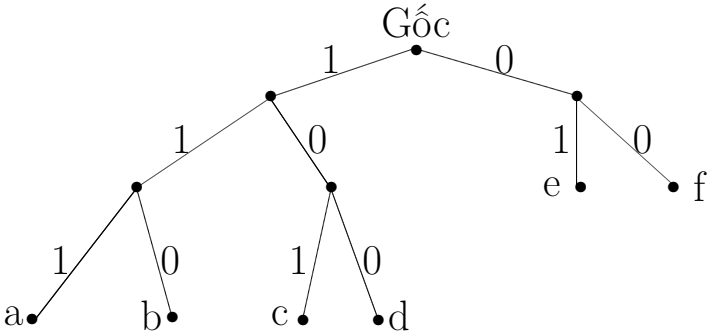
Mã tiền tố *là phương pháp mã hóa các chữ cái bảng chữ Latin bằng ký các xâu nhị phân (trong đó không phân biệt chữ cái thường và chữ cái viết hoa) sao cho xâu nhị phân ứng với một từ không bao giờ có thể là đoạn đầu của xâu nhị phân ứng với một từ khác. Kỹ thuật nén và mã hoá mã tiền tố là một trong những lĩnh vực thường hay được sử dụng trong Tin học. Cây nhị phân có nhiều ứng dụng trong việc nghiên cứu áp dụng các giải thuật trong lĩnh vực này. Một ví dụ mã tiền tố là dùng các xâu nhị phân có độ dài khác nhau để mã các ký tự để không có xâu nhị phân nào ứng với hơn một chữ cái. Trên cây nhị phân mã hoá, các lá là các ký tự cần mã hoá và đường đi từ cha đến con trái là 1 (hoặc 0), còn đi tới con phải là 0 (hoặc 1). Quá trình mã hoá sẽ duyệt cây đó từ gốc tới lá, khi tới nút con sẽ tạo ra một bít 0 hoặc 1 và tới nút lá sẽ tạo ra một xâu bít. Do vậy, mã sinh ra cho một ký tự sẽ không là phần đầu của ký tự khác.*

Ví dụ. Trong hình 7.24, cây nhị phân biểu diễn mã tiền tố của các ký tự a, b, c, d, e, f trong đó:

$a : 111 \quad b : 110 \quad c : 101$
$d : 000 \quad e : 01 \quad f: 00$







Hình 7.24: Cây nhị phân.

Mã Huffman *Một trong những thuật toán về mã tiền tố là Huffman xuất hiện từ năm 1952 tạo ra mã có độ dài thay đổi khác nhau khi đã có bảng tần số xuất hiện của các ký tự. Quá trình mã hoá và giải mã phụ thuộc vào việc xây dựng cây nhị phân mã hoá. Thuật toán Huffman tạo cây nhị phân từ nút lá đến nút gốc, ký tự nào có tần số càng cao thì nút lá tương ứng càng gần gốc hơn.*

Thuật toán mã hóa Huffman

Vào: Bảng tần số xuất hiện các ký tự sắp xếp giảm dần

Ra : Cây nhị phân biểu diễn mã, nhánh phải là 1, trái là 0.

Bước 1: Lấy hai phần tử cuối bảng tần số xuất hiện ra khỏi bảng

Bước 2: Nếu phần tử nào chưa nằm trong cây nhị phân thì tạo ra một nút lá chứa phần tử đó, phần tử này chính là ký tự. Nối hai nút tương ứng với hai phần tử này với nhau thông qua việc tạo nút cha của chúng. Phần tử có tần số xuất hiện lớn hơn là nút trái, nhỏ hơn là nút phải.

Bước 3: Tính tổng tần số xuất hiện của 2 phần tử này chèn vào bảng sao cho phù hợp với nguyên tắc giảm dần của bảng. Phần tử mới của bảng sẽ tương ứng với nút vừa được tạo ra ở bước 2.

Bước 4: Quay trở lại bước 1 đến khi bảng chỉ còn lại 1 phần tử. Phần tử cuối cùng tương ứng với nút gốc của cây nhị phân.

Ví dụ. Ta có kết quả mã Huffman cho các ký tự ở bảng sau:





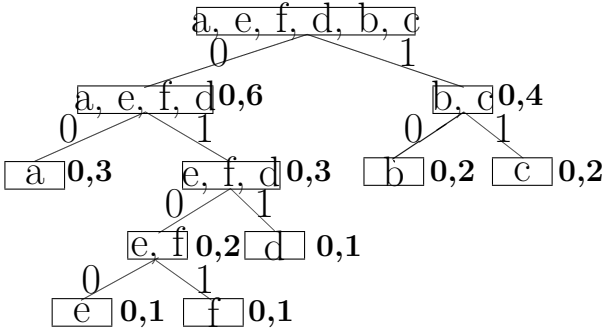






Ký tự	*Tần suất*	*Mã nhị phân*	*Chiều dài mã*
A	*0.3*	*00*	*2*
B	*0.2*	*10*	*2*
C	*0.2*	*11*	*2*
D	*0.1*	*011*	*3*
E	*0.1*	*0100*	*4*
F	*0.1*	*0101*	*4*

Cây nhị phân tương ứng được biểu diễn trong hình 7.25. Với thuật



Hình 7.25: Cây nhị phân biểu diễn mã Huffman.

toán Huffman trường hợp xấu nhất là thời gian hình thành cây nhị phân là $O(n)$ với n là số ký tự cần mã hoá.

Chương trình sau viết bằng ngôn ngữ tựa Pascal minh họa thuật toán tạo mã Huffman:

```
Program MahoaHuffman;
 Const n = 6; {Số ký tự cần mã hoá a, b, c, .....}
 Type Nod = record
 S:integer; {tần suất}
 Code:String; {mã nhị phân}
 Name:char; {tên ký tự}
 end;
Var a:array[1..n] of Nod;
 i,m:integer;
Procedure InputData; Khởi tạo bảng tần suất các ký tự
 Var i:integer;
 begin
 for i := 1 to n do with A[i] do
 begin S:=Round(exp(n/5)/exp(i/5))+1;Name:=Char(64+i);Code:='';
    end;
```

```
end;
Procedure FindCode(m:integer); {Sinh mã huffman}
 Var y,z:nod;
 k:integer;
 Begin
 if m=1 then exit;
 y:=a[m-1];
 a[m-1].s:=a[m-1].s+a[m].s;
 k:=m-1;
 While (k>1) and (a[k].s>a[k-1].s) do Begin
 z:=a[k]; a[k]:=a[k-1]; a[k-1]:=z; k:=k-1;
 End;
 FindCode(m-1);
 z:=a[k];
 for i:=k to m-2 do a[i]:=a[i+1];
 a[m-1]:=y;
```








```
    a[m-1].code:=z.code+'1'; a[m].code:=z.code+'0';
    End;
  BEGIN
    InputData; FindCode(n);
    For i:=1 to n do writeln(a[i].code);
  END.
```

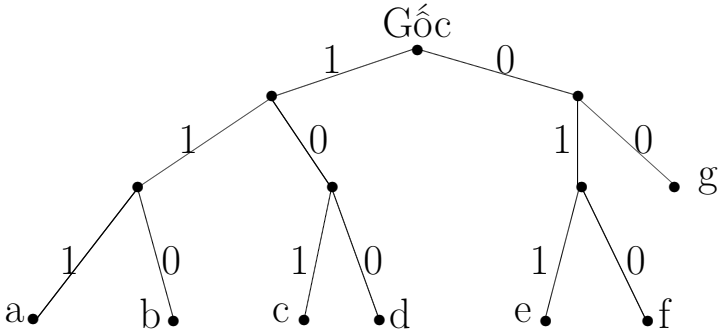
Bài tập:

1. *Hãy tìm cây khung tối ưu của đồ thị Petersen (hình 7.12) với trọng số các cạnh đều bằng 1.*

2. *Xác định mã tiền tố trong các mã sau đây:*
   a) a: 11, e: 00, t :10, s : 01.
   b) a: 0, e: 1, t: 01, s: 001,

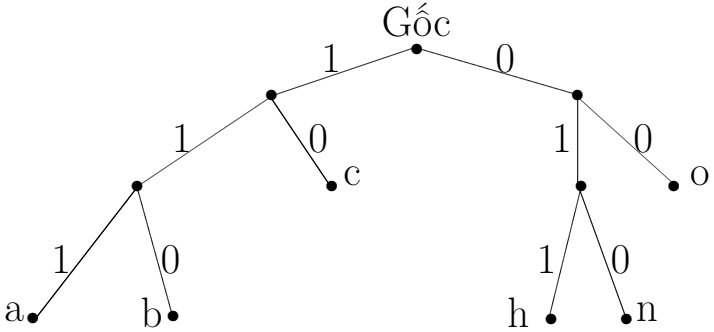
3. *Dựng cây nhị phân với mã tiền tố biểu diễn sơ đồ mã hóa*
   a: 11, e: 0, t :101, s : 100.

4. *Xác định mã của a, b, c, d, e, f và g nếu sơ đồ mã là cây nhị phân trong hình 7.26.*



Hình 7.26: Mã tiền tố.

5. *Hãy dùng cây nhị phân trong hình 7.27 để mã hóa câu "Chao cac ban". Và hãy dùng mã Huffman để mã hóa câu này và so sánh số bit sử dụng để mã hóa.*

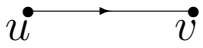


Hình 7.27: Cây mã hóa câu "*Chao cac ban*".

Một số loại đồ thị có hướng

Khái niệm cơ sở

Một đồ thị được gọi là đồ thị (hữu hạn) có hướng khi nó chỉ có các cạnh có hướng mà ta còn gọi là các cung. Đồ thị có hướng cũng được biểu diễn trên mặt phẳng như đồ thị vô hướng.



Hình 7.28: Cung nối $[u, v]$.

Một cung u được xác định nhờ điểm đầu x và điểm cuối y, mà ta thường kí hiệu $u = [x, y]$, trong đó x được gọi là đỉnh xuất phát và y được gọi là đỉnh đích của u. Chúng ta cũng nói rằng cung u liên hợp với đỉnh x và đỉnh y. Đồ thị có hướng G với tập đỉnh X và tập cạnh E thường được kí hiệu bởi $G = [X, E]$. Một đồ thị có hướng cũng được biểu diễn trên mặt phẳng bởi một hình, trong đó các đỉnh được biểu diễn thành các điểm được tô đậm và các cạnh có hướng và các cung là các đường liên tục và được bổ sung bởi mũi tên thể hiện chiều đi từ đỉnh xuất phát tới đỉnh đích. Các cung có cùng đỉnh xuất phát và đỉnh đích được gọi là các cung song song. Nếu

đỉnh xuất phát và đỉnh đích của một cung u trùng nhau thì cung u này được gọi là khuyên có hướng. Một đồ thị không có cung song song và khuyên có hướng nào cả được gọi là đồ thị có hướng đơn. Ta nói rằng cung u liên hợp với đỉnh x, nếu như x là đỉnh xuất phát hoặc là đỉnh đích của cung u. Nếu cung u liên hợp với đỉnh x, thì cung u được gọi là liên hợp hướng ra ngoài (liên hợp hướng vào trong) đối với đỉnh x, nếu như đỉnh x là đỉnh xuất phát (đỉnh đích) của cung u.

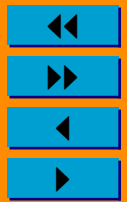
Ví dụ. Trong hình 7.28, đỉnh u liên hợp hướng vào trong đỉnh v.

Nếu x là một đỉnh của một đồ thị có hướng G, thì với $d_G^-(x)$, hoặc $d^-(x)$, ta kí hiệu số các cung đi tới x và gọi nó là bậc đi tới của đỉnh x. Cũng tương tự như vậy, với kí hiệu $d_G^+(x)$ hoặc $d^+(x)$ ta biểu thị số các cung xuất phát từ đỉnh x và gọi nó là bậc đi ra của đỉnh x. Các con số này là số các cung hướng vào trong và hướng ra ngoài liên hợp với đỉnh x. Bậc của đỉnh x chính là tổng $d_G(x) = d_G^-(x) + d_G^+(x)$. Tương tự như với đồ thị đơn vô hướng, ta gọi một đỉnh bậc 0 là đỉnh cô lập.

Ví dụ. Trong hình 7.29 ta có $d^+(a) = 4$, $d^-(a) = 1$, $d^(b) = 1$, $d^-(b) = 2$, $d^+(c) = 0$ và $d^-(c) = 2$.

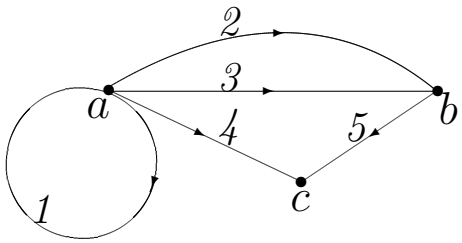
Ngoài ra, người ta cũng sử dụng ma trận liền kề và ma trận liên thuộc để biểu diễn đồ thị có hướng.





Cho trước một đồ thị có hướng $G = [V, E]$. Ta xây dựng một ma trận (a_{ij})có kích thước $n \times n$ trong đó n là số đỉnh của đồ thị và a_{ij} là số cung nối đỉnh i với đỉnh j. Ma trận này được gọi là ma trận liền kề của đồ thị cho trước. Ma trận liền kề của đồ thị có hướng không phải lúc nào cũng đối xứng. Chẳng hạn, đồ thị trong hình 7.29 có ma trận liền kề A:

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



Hình 7.29: Cung nối u với v.

Đường đi là một dãy các cung u_1, u_2, ..., u_k , sao cho với mọi $i = 1, 2, ..., k-1$ ta có đỉnh đích của cung u_i cũng chính là đỉnh xuất phát của cung u_{i+1} và mỗi đỉnh của đồ thị là không có mặt quá 1 lần trên đường đi. Gọi đỉnh xuất phát của u_1 là x và đỉnh đích của $u_k$là y, thì con đường còn được gọi là đường $x - y$. Số các cạnh k được gọi là độ dài của đường đi. Đường đi qua tất cả các đỉnh của đồ thị được gọi là đường Hamilton. Đường

đi với đỉnh xuất phát

Chu trình là một dãy các cung u_1, u_2, ..., u_k sao cho đỉnh xuất phát của u_1 là đỉnh tới của u_k và u_1, u_2, ..., u_{k-1} là một đường đi. Độ dài của chu trình là số cạnh của nó. Chu trình chứa tất cả các đỉnh của đồ thị được gọi là chu trình Hamilton.

Ví dụ. Đồ thị trong hình có một chu trình do cung được đánh số 1 tạo thành. Nếu đổi chiều cạnh số 4 thì ta sẽ có một chu trình độ Hamilton.

Một đồ thị có hướng G được gọi là liên thông, nếu bỏ hướng các cung thì G trở thành một đồ thị vô hướng liên thông. G được gọi là liên thông mạnh, nếu với hai đỉnh x, y tùy ý của G luôn tồn tại một đường đi nối x với y. Rõ ràng một đồ thị có hướng là đồ thị liên thông nếu như nó là đồ thị liên thông mạnh. Ta có thể dễ thấy điều ngược lại thì không đúng.

Ví dụ. Đồ thị trong hình 7.29 là đồ thị liên thông nhưng không liên thông mạnh, do từ đỉnh c không có đường đi tới các đỉnh còn lại của đồ thị.

Định lý sau đây cho ta một điều kiện cần và đủ cho tính liên thông mạnh của đồ thị.





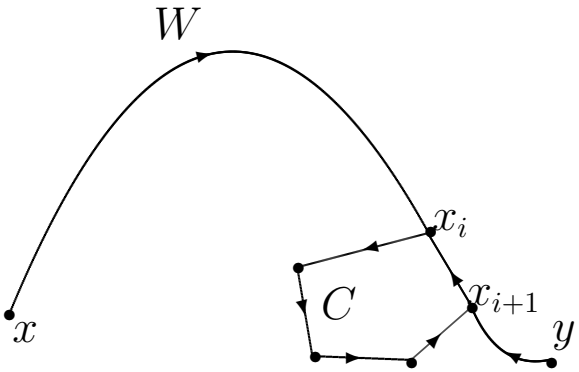






Định lý 7.18. *Một đồ thị liên thông G là một đồ thị liên thông mạnh khi và chỉ khi một cung tùy ý của G nằm trong ít nhất một chu trình.*

Chứng minh. Giả sử G là một đồ thị có hướng cho trước sao cho mỗi cung bất kì của nó nằm trong một chu trình. Để chỉ ra một mâu thuẫn, ta giả sử rằng trong G tồn tại hai đỉnh x và y sao cho không tồn tại một đường đi nối x với y. Với $\mathcal{M}$ ta kí hiệu tập hợp các đỉnh y được nối với đỉnh x bởi một đường đi $x - y$ nào đó. Do G là đồ thị liên thông, tồn tại một đường đi vô hướng $K = (x = x_1, x_2, ..., x_k = y)$ nối x với y. Rõ ràng



Hình 7.30: Đường đi từ x tới x_{i+1}.

$x_1 = x \in \mathcal{M}$ và $x_k = y \notin \mathcal{M}$. Giả sử i là số lớn nhất sao cho $x_i \in \mathcal{M}$

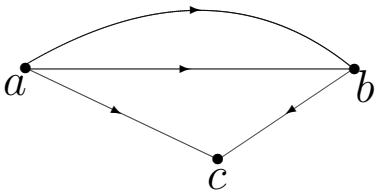
(ta đi từ đỉnh x tới đỉnh x_i bởi một đường đi W nào đó) và $x_{i+1} \notin \mathcal{M}$. Khi đó cung giữa hai đỉnh x_i và x_{i+1} có hướng đi từ đỉnh x_{i+1} tới đỉnh x_i, ta kí hiệu $u = [x_{i+1}, x_i]$. Theo giả thiết cung u nằm trên một chu trình C. Ta đi từ đỉnh x tới đỉnh x_i dọc theo W và sau đó dọc theo chu trình C tới đỉnh x_{i+1}. Trong những lộ trình đi từ x tới x_{i+1} như vậy thì lộ trình với ít cung nhất phải là một con đường. Vậy có $x_{i+1} \in \mathcal{M}$, vô lý. Mâu thuẫn này chứng tỏ rằng G là đồ thị liên thông mạnh.

Bây giờ giả sử rằng cho trước một đồ thị liên thông mạnh G và một cung u tùy ý của nó. Nếu u là một khuyên, thì cung u đã nằm trên một chu trình. Nếu đỉnh xuất phát x và đỉnh đích y của cung u khác nhau, thì tồn tại trong đồ thị G một đường đi W nối đỉnh y với đỉnh x. Khi đó $W \cup \{u\}$ là một chu trình chứa cung u, là điều cần chứng minh. □

Một số đồ thị có hướng đặc biệt

Có một loại đồ thị có hướng khá đơn giản là đồ thị không có chu trình. Một đồ thị G được gọi là đồ thị phản chu trình, nếu như nó không chứa chu trình nào cả. Mọi đồ thị phản chu trình không có khuyên và mỗi đồ thị con của đồ thị phản chu trình lại là một đồ thị phản chu trình.

Một đỉnh x của một đồ thị có hướng G được gọi là đỉnh nguồn của đồ thị G, nếu như đỉnh x không phải là đỉnh đích của cung của G nào cả Tất nhiên mỗi đỉnh cô lập cũng được coi là một đỉnh nguồn. Bằng cách tương tự ta định nghĩa một đỉnh hạ lưu của một đồ thị là một đỉnh y với tính chất là nó không phải là đỉnh xuất phát của bất cứ cung nào của đồ thị G cả.



Hình 7.31:

Ví dụ. Đỉnh a trong hình 7.20 là đỉnh nguồn còn đỉnh b là đỉnh đích.

Đỉnh nguồn và đỉnh đích đóng vai trò quan trọng trong lý thuyết vận tải. Những vấn đề vận tải trên đồ thị có hướng và cạnh có hạn chế lượng vận tải tối đa đã được Fork và Funkelson giải quyết, mà ta không xét ở đây. Về sự tồn tại của đỉnh nguồn và đỉnh hạ lưu trong đồ thị phản chu trình, ta có định lý sau:

Định lý 7.19. *Trong mỗi đồ thị phản chu trình luôn tồn tại ít nhất một đỉnh nguồn và một đỉnh hạ lưu.*

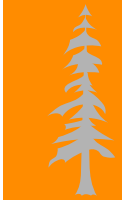
Chứng minh. Rõ ràng chỉ cần chứng minh rằng trong mỗi đồ thị phản chu trình luôn tồn tại một đỉnh nguồn. Sự tồn tại của đỉnh hạ lưu hoàn toàn

tương tự.

Ta chọn trong một đồ thị phản chu trình G cho trước một đỉnh tùy ý x_1. Nếu đỉnh x_1 là một đỉnh nguồn, thì định lý được chứng minh. Nếu đỉnh x_1 không phải là đỉnh nguồn, thì tồn tại trong đồ thị phản chu trình G ít nhất là một cung u với đỉnh đích là x_1. Gọi đỉnh xuất phát của cung này là đỉnh x_2, thì $x_2 \neq x_1$ do G là đồ thị phản chu trình. Giả sử rằng đỉnh x_2 là một đỉnh nguồn, thì định lý được chứng minh. Trong trường hợp đỉnh x_2 không phải là đỉnh nguồn thì tồn tại trong G một cung với đỉnh đích là đỉnh x_2. Giả sử đỉnh xuất phát của cung này là đỉnh x_3, thì ta có $x_3 \notin \{x_1, x_2\}$ vì G là một đồ thị phản chu trình. Bằng cách tương tự, chúng ta cứ tiến hành như vậy và thu được một dãy các đỉnh x_1, x_2, ..., x_k, ..., trong đó không có đỉnh nào xuất hiện hai lần. Do tính hữu hạn của tập hợp đỉnh, dãy này không thể kéo dài vô hạn mà phải kết thúc tại một đỉnh x_0 nào đó là đỉnh nguồn trong đồ thị G. Định lý được chứng minh. □

Định lý dưới đây cho chúng ta sự mô tả về đồ thị phản chu trình:

Định lý 7.20. *Một đồ thị có hướng $G = [X, E]$ với n đỉnh là đồ thị phản chu trình khi và chỉ khi chúng ta có thể đánh số các đỉnh của đồ thị bởi các số 1, 2, ..., n sao cho mỗi cung $[i, j]$ thỏa mãn quan hệ $i < j$.*



Chứng minh. Giả sử rằng ta có thể đánh số các cạnh của đồ thị có hướng G bởi các số 1, 2, ..., n sao cho mỗi cung $[i, j]$ thỏa mãn quan hệ $i < j$. Rõ ràng khi đó G không có khuyên. Bây giờ ta chứng minh rằng G không có chu trình. Thực vậy, giả sử đồ thị G có chu trình với các đỉnh được đánh số là i_1, i_2, i_3, ..., i_n dọc trên chu trình theo một chiều đi. Theo giả thiết đầu bài thì $i_1 < i_2 < i_3 .... < i_n$ và $i_n < i_1$. Từ đó suy ra $i_1 < i_1$, vô lí. Mâu thuẫn đó chứng tỏ rằng đồ thị đã cho phản chu trình.

Bây giờ cho trước một đồ thị phản đối xứng G với n đỉnh. Chúng ta sẽ chứng minh rằng các đỉnh của đồ thị G có thể được đánh số bởi các số 1, 2, ..., n sao cho với mỗi cung $[i, j]$ ta có quan hệ $i < j$. Theo định lý 7.19, tồn tại trong đồ thị phản chu trình G một đỉnh nguồn, mà ta đánh số 1. Trong trường hợp đồ thị G có còn các đỉnh khác, ta khảo sát đồ thị $G_1 = G - \{1\}$, là đồ thị thu được khi bỏ đỉnh 1 đi. Đồ thị thu được G_1 dĩ nhiên cũng là một đồ thị phản chu trình, và ta có thể đánh số đỉnh nguồn của đồ thị này bởi số 2. Nếu đồ thị G_1 còn có nhiều đỉnh khác nữa, thì ta lại có thể xét đồ thị $G_2 = G_1 - \{2\}$ .... và cứ bằng cách này chúng ta thu được đồ thị chỉ có đúng một đỉnh, mà ta sẽ đánh số đỉnh này với số n. Vậy các đỉnh của đồ thị được đánh số bởi các số 1, 2, ..., n sao cho mỗi cung $[i, j]$ thỏa mãn quan hệ $i < j$. Định lý như vậy được chứng minh. □

Một trong những ứng dụng lý thú của đồ thị phản chu trình được Oystein Ore (trong năm 1960) xây dựng một loại cây gia phả, trong đó quan hệ bố và con được biểu diễn bởi một cung nối. Nếu chúng ta quan sát một tập hợp các loài sinh vật sống của một loài nào đó, chúng ta có thể gán cho mỗi cá thể một đỉnh của một đồ thị có hướng; cung $[x, y]$ được nối từ cá thể bố x tới cá thể y, nếu như cá thể y là con đẻ của cá thể x. Vì không có cá thể tự bản thân là con đẻ của mình, cho nên đồ thị thu được là một đồ thị không có khuyên. Ngoài ra dễ thấy đồ thị thu được là một đồ thị phản chu trình đặc biệt: mỗi đỉnh là đỉnh đích của đúng hai cung.

Bây giờ chúng ta muốn làm quen với một đồ thị có hướng đặc biệt được gọi là Turnier. Về Turnier có cả một lý thuyết vượt ra khỏi giới hạn của cuốn sách này. Người ta hiểu một Turnier là một đồ thị có hướng sao cho nếu bỏ hướng các cạnh đi thì ta thu được một đồ thị đầy đủ. Ta còn có thể gọi một Turnier là một đồ thị đầy đủ có hướng (hình 7.32 là một 9-Turnier) và được kí hiệu bởi T_n. Một đồ thị con sinh bởi m đỉnh của một n-Turnier T_n được gọi là một m-Turnier con.

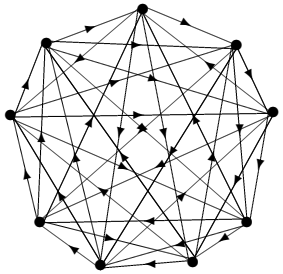
Trong một n-Turnier T_n ta có một cách hiển nhiên:
1) $d^+(x) + d^-(x) = n - 1$ cho đỉnh x tùy ý của T_n.








2) T_n có không quá một đỉnh nguồn và không quá một đỉnh hạ lưu.



Hình 7.32: Turnier 9 đỉnh.

Tên gọi Turnier bắt nguồn từ thể thao: Nếu có một đại hội thể thao (mà trong tiếng Đức được gọi là Turnier) gồm có n đấu thủ $x1$, $x2$, ..., xn tham gia, trong đó mỗi người phải gặp một người khác đúng một lần. Không có trận đấu nào được kết thúc với kết quả hòa - chẳng hạn như đối với các trận đấu bóng bàn hoặc bóng rổ. Khi đó ta có thể gán mỗi đội với một trong các số không vượt quá n bằng phép gán $x_i \to i$ ta có một biểu diễn n đội bằng các đỉnh 1, 2, .., n của một đồ thị đầy đủ K_n.

Các cung (i, j) với hai đỉnh khác nhau i, j của đồ thị K_n được gán cho một chiều đi thông qua một mũi tên từ i tới j, nếu như đội x_i thắng đội x_j.

Cuối cùng sau trận đấu, người ta thu được một đồ thị đầy đủ có hướng, mà nhìn vào đồ thị này người ta biết ngay kết quả từng trận đấu riêng lẻ.

Một Turnier với n đỉnh được gọi tắt là một n-Turnier (hình 7.32 là T_9).

Người ta đã chứng minh định lý sau:.

Định lý 7.21. *Trong mỗi một Turnier luôn tồn tại ít nhất một đường Hamilton. Một Turnier có đúng một đường Hamilton nếu nó là đồ thị có hướng phản chu trình.*

Chứng minh. Để chứng minh khẳng định thứ nhất của bài toán ta giả sử rằng tồn tại một Turnier không có đường Hamilton, và sẽ chỉ ra một mâu thuẫn. Giả sử rằng T_n là một Turnier và $B = (x_1, x_2, ..., x_k)$ là một đường đi có độ dài lớn nhất với $k < n$. Ta chọn một đỉnh tùy ý $x \in \{x_1, x_2, ..., x_k\}$. Do tính cực đại của B đỉnh x không phải là đỉnh xuất phát của cung nối x với x_1 và không phải là đỉnh đích của cung nối đỉnh x với đỉnh x_k. Giả sử rằng $i_0 = \max\{i : (x_i, x)\text{là một cung trong}T_n\}$. Khi đó ta có $1 \leq i_0 \leq k-1$. Rõ ràng rằng $B' = (x_1, x_2, ..., , x, , ..., x_k)$ là một đường đi trong Turnier T_n và dài hơn B, mâu thuẫn với tính cực đại của B.

Để chứng minh phần thứ hai của định lý, ta nhận thấy rằng nếu T_n phản chu trình, thì T_n có đúng một đương đi Hamilton. Khẳng định này được suy ra trực tiếp từ định lý 7.20, vì một đường Hamilton chỉ có thể có các đỉnh xếp theo thứ tự 1, 2, 3, ..., n với cách đánh số các đỉnh của T_n theo thứ tự như kết luận của định lý 7.20. Như vậy đường Hamilton là được xác

định duy nhất. □

Một Turnier T_n với $n \geq 3$ đỉnh được gọi là bắc cầu, nếu như T_n không có chu trình độ dài 3. Từ "bắc cầu" có nghĩa là với các cung (x_1, x_2) và (x_2, x_3) nối x_1 tới x_2 và từ x_2 tới x_3 thì cũng tồn tại cung (x_1, x_3) từ x_1 tới x_3. Rõ ràng rằng mỗi Turnier phản chu trình đều bắc cầu. Ngược lại cũng đúng, như ta sẽ chứng minh trong định lý sau đây.

Định lý 7.22. *Mỗi Turnier T_n với $n \geq 3$ đỉnh là bắc cầu khi và chỉ khi T_n phản chu trình.*

Chứng minh. Áp dụng định lý 7.20 ta thấy mỗi đồ thị phản chu trình đều là đồ thị bắc cầu nhờ cách đánh số các đỉnh thỏa mãn định lý 7.20. Do đó ta chỉ cần chứng minh điều ngược lại là mỗi Turnier bắc cầu là một đồ thị có hướng phản chu trình. Ta chứng minh bằng phản chứng.

Giả sử rằng $Z = (x_1, x_2, ..., x_k, x_1)$ là một chu trình có độ dài nhỏ nhất. Dễ thấy $n \geq 4$ do kết luận định lý hiển nhiên đúng cho T_3. Vì T_n có tính bắc cầu, chu trình Z có độ dài > 3, nghĩa là $k \geq 4$. Bây giờ ta xét các đỉnh x_1, x_2 và x_3. Vì T_n bắc cầu, cung nối hai đỉnh x_1 và x_3 với nhau phải đi từ x_1 tới x_3, do đó $Z' = (x_1, x_3, x_4, ..., x_k, x_1)$ là một chu trình có

độ dài nhỏ hơn độ dài của chu trình Z, mâu thuẫn giả thiết Z có độ dài nhỏ nhất. Vậy T_n là đồ thị phản chu trình. Định lý được chứng minh. □

Khác với đồ thị vô hướng, câu trả lời về sự tồn tại chu trình hamilton trong Turnier được giải quyết trọn vẹn trong định lý sau.

Định lý 7.23. *Giả sử T_n là một Turnier với $n \geq 3$ đỉnh. Các mệnh đề sau tương đương với nhau:*
1) T_n có một chu trình hamilton.
2) T_n là đồ thị liên thông mạnh.
3) Với mỗi giá trị $k = 3, 4, \ldots, n$ thì Turnier T_n có một chu trình độ dài k.

Chứng minh. Rằng 1)⇒ 2) và 3)⇒ 1) thỏa mãn, ta có thể thấy dễ dàng. Ta chỉ còn phải chứng minh 2)⇒ 3). Ta chứng minh bằng qui nạp theo k. Giả sử rằng T_n liên thông mạnh. Khi đó rõ ràng T_n không phải đồ thị phản chu trình, và theo định lý 7.22 thì T_n không phải là đồ thị bắc cầu và do đó T_n có một chu trình độ dài 3. Khẳng định được chứng minh cho $k = 3$.

Bây giờ giả sử rằng với một $k \leq n$ tồn tại một chu trình độ dài k. Nếu như có $k \leq n - 1$, ta sẽ chỉ ra rằng tồn tại một chu trình độ dài $k + 1$. Giả sử rằng ta có $C = (x_1, x_2, ..., x_k, x_1)$ là một chu trình độ dài k. Vì

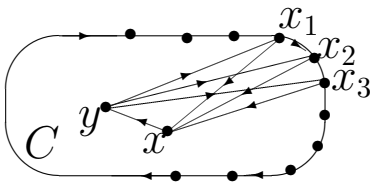


$k \leq n-1$, tồn tại một đỉnh $x \notin \{x_1, x_2, ..., x_k\}$. Ta phân biệt hai trường hợp sau:
a) Tồn tại một đỉnh $x \notin \{x_1, x_2, ..., x_k\}$ và hai cung $[x_i, x]$ và $[x, x_j]$, chẳng hạn $1 \leq i < j \leq k$ trong T_n. Trong trường hợp này chúng ta có thể thấy rằng tồn tại một giá trị $i \leq t \leq j-1$, sao cho $[x_t, x]$ và $[x, x_{t+1}]$ là các cung trong Turnier T_n. Bây giờ

$$C' = (x_1, x_2, \ldots, x_t, x, x_{t+1}, \ldots x_k, x_1)$$

là một chu trình trong Turnier T_n với độ dài $k+1$.
b) Mỗi đỉnh $x \notin \{x_1, x_2, ..., x_k\}$ là đỉnh đích của tất cả các cung nối x với các đỉnh thuộc C hoặc là đỉnh xuất phát của tất cả các cung nối đỉnh x với tất cả các đỉnh của C.

Trong trường hợp này ta có thể chia tập đỉnh $T_n - C$ làm hai tập hợp A và B, sao cho mỗi đỉnh $x \in A$ chỉ là đỉnh đích của mỗi cung nối đỉnh x với một đỉnh tùy ý của C và mỗi đỉnh $y \in B$ chỉ là đỉnh xuất phát của các cung nối đỉnh y với các đỉnh của C. Do $k \leq n-1$ thì $A \cup B$ không phải tập hợp rỗng. Do T_n liên thông mạnh, các tập hợp A và tập hợp B cùng



Hình 7.34:

không rỗng và tồn tại ít nhất một cung nối từ A tới B, chẳng hạn cung $[x, y]$ với $x \in A$ và $y \in B$. Rõ ràng là $C' = (x_1, x, y, x_3, \ldots, x_k, x_1)$ là một chu trình với độ dài $k+1$ trong T_n. Vậy khẳng định của ta cũng đúng cho giá trị $k+1$. Định lý được chứng minh. □

Bài tập:

1. *Trong một đồ thị phản chu trình T_n không tồn tại hai đỉnh có cùng số cạnh xuất phát. Hãy chứng minh điều đó.*

2. *Chứng minh rằng hai khẳng định sau tương đương với nhau:*

   (a) Ta có thể đánh số n đỉnh của một Turnier T_n với các số 1, 2, ..., n, sao cho cho với mọi đỉnh i ta có $d^+(i) = n - i$.

   (b) T_n là đồ thị phản chu trình.

3. *Chứng minh rằng một Turnier có duy nhất một đường Hamilton khi và chỉ khi nó là đồ thị phản chu trình.*

