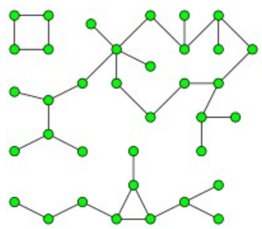


# TOÁN RỜI RẠC

**NGUYỄN ĐÌNH CƯỜNG**

**BỘ MÔN KỸ THUẬT PHẦN MỀM**

**KHOA CNTT-ĐHNT**

## LÝ THUYẾT TỔ HỢP

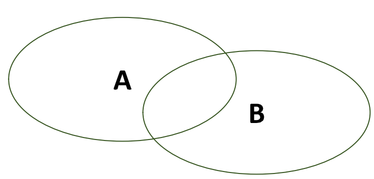
### Các phép toán tập hợp

**Phần bù**

$$\bar{A} = \{x \in \mathrm{X}: x \notin A\}$$

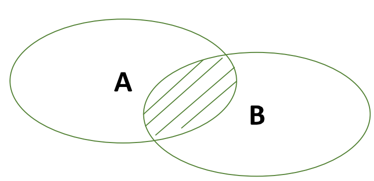
**Hợp của A và B**

$$A \cup B = \{x: x \in \mathrm{A} \text{ hoặc } x \in B\}$$




**Giao của A và B**

$$A \cap B = \{x: x \in \mathrm{A} \text{ và } x \in B\}$$




**Kết hợp**

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

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

**Giao hoán**

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

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

**Phân bố**

$$\mathrm{A} \cup (\mathrm{B} \cap \mathrm{C}) = (A \cup B) \cap (A \cup C)$$

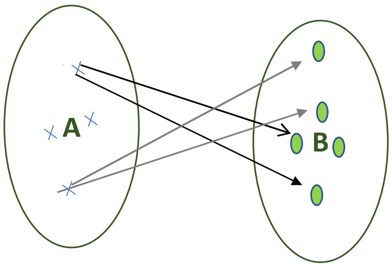
$$\mathrm{A} \cap (\mathrm{B} \cup \mathrm{C}) = (A \cap B) \cup (A \cap C)$$

**Đối ngẫu**

$$\overline{AUB} = \bar{A} \cap \bar{B}$$

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

**Tích Đềcát của các tập hợp**



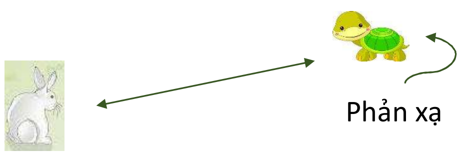

$$A \times B = \{(\mathrm{a}, \mathrm{b}) | \mathrm{a} \in \mathrm{A}, \mathrm{b} \in B\}$$

$$A_1 \times A_2 \times \cdots A_k = \{(a_{11} a_2, \cdots, a_k) | a_i \in A_i = 1,2, \dots, k\}$$

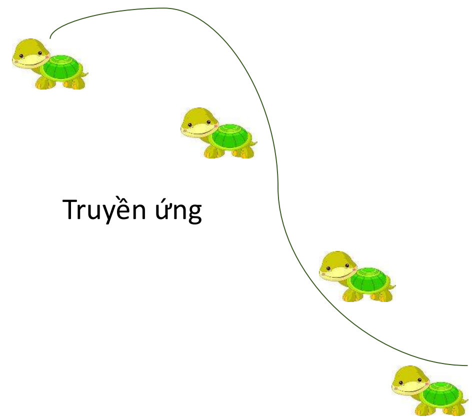
$$A^k = A \times A \times A \dots \times A \qquad k \text{ lần}$$

**Quan hệ tương đương và phân hoạch**

Đối xứng



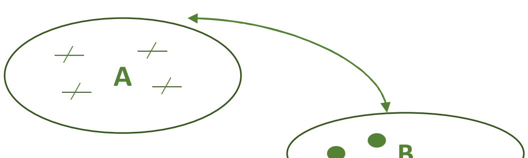
Phản xạ



Truyền ứng

Tính chất vật lý và sự hữu hình các phần tử trong tập hợp

**Nguyên lý cộng**




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

Một đoàn vận động viên gồm 2 môn bắn súng và bơi được cử đi thi đấu ở nước ngoài. Nam có 10 người. Số vận động viên thi bắn súng kể cả nam và nữ là 14. Số nữ vận động viên thi bơi bằng số nam vận động viên thi bắn súng. Số nữ vận động viên thi bơi bằng số nam vận động viên thi bắn súng. Hỏi toàn đoàn có bao nhiêu người.

Trả lời :

Toàn đoàn có 10 + 14 = 24 người

**Nguyên lý cộng**

$n_1 = 10$
$n_2 = 20$
$n_3 = 30$
$k$=0
for $i_1$=1 to $n_1$ do $k = k + 1$
for $i_2$=1 to $n_2$ do $k = k + 1$
for $i_3$=1 to $n_3$ do $k = k + 1$

Giá trị $k$ sẽ bao nhiêu sau khi đoạn chương trình này thực hiện

**Nguyên lý nhân**

Có bao nhiêu chuỗi kí tự có độ dài 10 chỉ chứa 2 chữ cái A, B, bắt đầu bởi AAA hoặc ABA

$n_1 = 10$
$n_2 = 20$
$n_3 = 30$
$k$=0
for $i_1$=1 to $n_1$ do
for $i_2$=1 to $n_2$ do
for $i_3$=1 to $n_3$ do $k = k + 1$

**Các cấu hình tổ hợp đơn giản**

Chỉnh hợp lặp

Định nghĩa 1. *Một chỉnh hợp lặp chập k của n phần tử là một bộ có thứ tự gồm k thành phần lấy từ n phần tử đã cho. Các phần tử có thể lặp lại.*

- Tính số dãy nhị phân độ dài n, 1001000000100000000100000000001

Trả lời $2^N$

- Tính số tập con của một *n*-tập

X= $\{x_1 x_2, \cdots, x_n\}$

Biểu diễn mỗi tập con bằng dãy nhị phân độ dài n

b= $\{b_1 b_2, \cdots, b_n\}$

Chỉnh hợp không lặp

Định nghĩa 2. *Một chỉnh hợp lặp chập k của n phần tử là một bộ có thứ tự gồm k thành phần lấy từ n phần tử đã cho. Các phần tử không được lặp lại.*

**Các cấu hình tổ hợp đơn giản**

**Hoán vị**

Định nghĩa 3. *Ta gọi một hoán vị n phần tử là một cách xếp thứ tự các phần tử đó.*

- 6 người đứng xếp thành hàng ngang để chụp ảnh. Hỏi có thể bố trí bao nhiêu kiểu

Giải
Mỗi kiểu ảnh là một hoán vị của 6 người. Từ đó nhận được số kiểu ảnh có thể bố trí là 6! = 720

**Tổ hợp**

Định nghĩa 4. Tổ hợp chập $k$ của $n$ phần tử là một bộ không kể thứ tự gồm $k$ thành phần khác nhau lấy từ $n$ phần tử đã cho.

$$C_n^k = \frac{n!}{k!\,(n-k)!} \cdot$$

Có n đội bóng thi đấu vòng tròn. Hỏi phải tổ chức báo nhiêu trận

Giải. Cứ 2 đội thì có một trận, từ đó suy ra số trận đấu bằng số cách chọn 2 đội từ n đội

$$C_n^2 = \frac{n(n-1)}{2}$$

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1

$$c_n^0 + c_n^1 + c_n^2 + \cdots + c_n^n = 2^n$$

Các hệ số trong tam giác PASCAL

Viết chương trình máy tính in ra tam giác PASCAL

## BÀI TOÁN ĐẾM

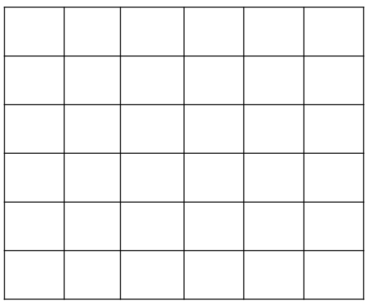
- Thường sử dụng nguyên lý cộng và nguyên lý nhân

Thí dụ

Có bao nhiêu cách xếp 5 người thành hàng ngang sao cho A không đứng cạnh B.

Giải

Xem A và B là một chổ , ta có 4! =24 cách xếp. Toàn bộ có 5!=120 Cách xếp, từ đó nhận được số cách xếp là 120 – 48 = 72



| A | B | X | X | X |
|---|---|---|---|---|

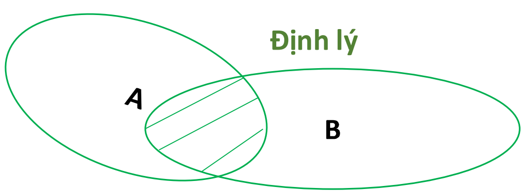
Hỏi có bao nhiêu đường đi khác nhau từ nút (0, 0) đến nút (n, m)

for $\mathrm{i} = 2$ to $n\ do$
for $j = n\ down\ to\ i\ d$
if a[j-1] > a[j] then Swap(a[j-1], a[j]);

## BÀI TOÁN ĐẾM

Nguyên lý bù trừ

**Định lý**




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

$$|A_1 U A_2 \cup A_3 \ldots A_m| = N_1 - N_2 + \cdots + (-1)^{n-1} N_m$$

Thí dụ

Hỏi trong tập X={1, 2, ...., 10000} có bao nhiêu số không chia hết cho bất cứ số nào trong các số 3, 4, 7.

$$A_i = \{x \in X : chia\ hết\ cho\ i\}, i = 3, 4, 7$$

$$A_3 \cup A_4 \cup A_7$$

Số lượng các số cần đếm =$|X| - |A_3 \cup A_4 \cup A_7| \;= |X| - (N_1 - N_2 + N_3)$

$N_1 = |A_3|$+ $|A_4|$+ $|A_7|$=$\left[\frac{10000}{3}\right] + \left[\frac{10000}{4}\right]$+ $\left[\frac{10000}{7}\right]$=3333+2500+1428=7261

$N_2 = |A_3 \cap A_4|$+ $|A_3 \cap A_7|$+ $|A_4 \cap A_7|$=$\left[\frac{10000}{3\times4}\right] + \left[\frac{1000}{3\times7}\right]$+ $\left[\frac{1000}{4\times7}\right]$ $=$ 833 + 476 + 357=1666

$N_3 = |A_3 \cap A_4 \cap A_7|$=$\left[\frac{10000}{3\times4\times7}\right] = 119$

Số lượng các số cần đếm =10000 -7261+ 1666 -119=4286

**Có bao nhiêu xâu nhị phân bắt đầu bởi 00 và kết thúc 11**

## Tổ hợp lặp

Một sinh viên mua r=4 cây bút chì chọn trong n=3 màu khác nhau là xanh, đỏ và vàng. Hỏi có bao nhiêu cách chọn mua hang khác nhau.

<table>
<tr><th>Cách mua</th><th>Xanh</th><th>Đỏ</th><th>Vàng</th><th>Biểu diễn</th></tr>
<tr><td>4 bút màu xanh</td><td>****</td><td></td><td></td><td>****||</td></tr>
<tr><td>1 xanh, 3 vàng</td><td>*</td><td></td><td>***</td><td>*||***</td></tr>
<tr><td>2 đỏ, 2 vàng</td><td></td><td>**</td><td>**</td><td>|**|**</td></tr>
</table>

Xếp 4 dấu * và 2 dấu | vào 6 vị trí : $c_6^4$

Phương trình $x_1 + x_2 + x_3 = 10$ có bao nhiêu nghiệm nguyên không âm

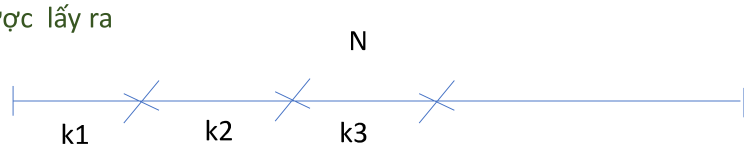
<table>
<tr><th>Nghiệm</th><th>x1</th><th>x2</th><th>x3</th><th>Biểu diễn</th></tr>
<tr><td>(4, 0, 6)</td><td>****</td><td></td><td>******</td><td>****||******</td></tr>
<tr><td>(3, 4, 3)</td><td>***</td><td>****</td><td>***</td><td>***|****|***</td></tr>
</table>

**Định nghĩa 5.** Một tổ hợp lặp chập r của n phần tử cho trước là việc chọn r phần tử trong n phần tử, trong đó mỗi phần tử có thể được chọn lại nhiều lần.

**Định nghĩa 6.** Số tổ hợp lặp chập r của n phần tử bằng số tổ hợp chập r của r+n-1 phần tử.

$$c_{r+\mathrm{n}-1}^{r} = \frac{(n+r-1)!}{r!\,(n-1)!}$$

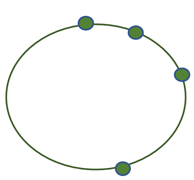
### Bài toán 1. Có bao nhiêu cách lấy k phần tử trong n phần tử xếp trên đường thẳng sao cho không có 2 phần tử kề nhau cùng được lấy ra




- Số phần tử còn lại n-k
- Số đoạn trống có thể có k1 + k2 +k3 +1= N-k +1

Mỗi cách lấy k khoảng từ các khoảng này, sẽ tương ứng với một cách chọn k phần tử t hỏa mãn yêu cầu đã nêu. Vậy số cần tìm. $C_{N-k+1}^{k}$

### Bài toán 2. Có bao nhiêu cách lấy k phần tử trong n phần tử xếp trên vòng tròn sao cho không có 2 phần tử kề nhau cùng được lấy ra



Vì trên vòng tròn nên ta có thể cố định phần tử a trong n phần tử. Chia ra 2 lớp bài toán a được chọn và a không được chọn

- Nếu chọn a, khi đó 2 phần tử kề a sẽ không được chọn. Như vậy sẽ lấy k-1 phần tử từ n-3 phần tử còn lại. Các phần tử này được xem như trên đường thẳng. Theo bài toán số 1, số cách là $C_{n-k-1}^{k-1}$
- a không được chọn, bỏ a đi, ta đưa về bài toán lấy k phần tử từ n-1 phần tử xếp trên đường thẳng. Theo bài toán 1 số cách là

Số cách cần tìm $C_{n-k-1}^{k-1} + C_{n-k}^{k} = \frac{n}{n-k} C_{n-k}^{k}$

## Hoán vị lặp

Có thể nhận được bao nhiêu từ khác nhau bằng cách sắp xếp lại các chữ cái của từ **SUCCESS**

| | | | | | | |
|---|---|---|---|---|---|---|
| S | S | S | C | C | E | U |

- Có 3 chữ S, 2 chữ C và 1 chữ U
- Chọn vị trí cho chữ S: $c_7^3$
- Có 2 vị trí cho 2 chữ C: $c_4^2$
- Đặt chữ U bằng $c_2^1$
- Đặt chữ E, $c_2^1$ t

$$C_7^3 \times C_4^2 \times C_2^1 \times C_1^1 = \frac{7!}{3!\,21!\,1!}$$

Trong không gian Oxyz, một robot di chuyển bằng cách nhảy từng bước dài một đơn vị theo hướng dương của trục x, y hoặc z. Tính số cách khắc nhau mà robot có thể di chuyển từ tọa độ (0, 0, 0) đến điểm (4, 3, 5)

Ví dụ: Biểu diễn một đường đi XXXXYYYYXZZZ

$$C_{12}^4 \times C_8^3 \times C_5^5 = \frac{12!}{4!\,3!\,5!}$$

**Định nghĩa 7.** Hoán vị của n phần tử trong đó có $n_1$ *phần tử giống nhau thuộc loại* $1, n_2$ phần tử giống nhau thuộc loại 2,…và $n_k$ phần tử giống nhau thuộc loại $k$ $(n_1 + n_2 + \cdots + n_k = n)$ . Được gọi là hoán vị lặp chập $k$ của $n$ *phần tử*.

**Định nghĩa 8.** Số hoán vị lặp chập k của n phần tử trong đó có $n_1$ *phần tử giống nhau thuộc loại* $1, n_2$ phần tử giống nhau thuộc loại 2,…và $n_k$ phần tử giống nhau thuộc loại $k$ .

$$\frac{n_1}{n_1!\,n_2!\ldots n_k!}$$

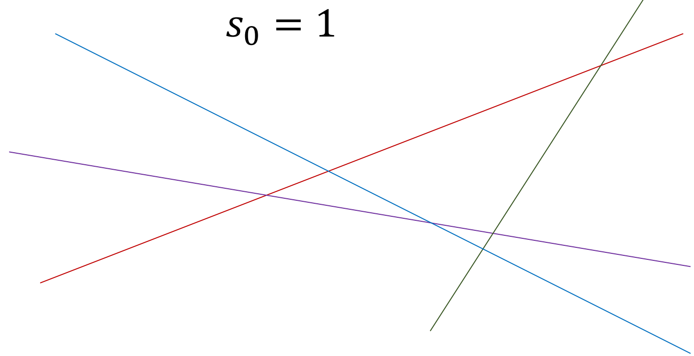
### Công thức truy hồi

Ví dụ: Trên mặt phẳng, kẻ n đường thẳng sao cho không có 2 đường thẳng nào song song và 3 đường nào đồng quy. Hỏi mặt phẳng được chia làm mấy phần.

Giải. Gọi số phần mặt phẳng được chia bởi n đường thẳng là $S_n$. Giả sử đã kẻ n-1 đường thẳng, kẻ thêm đường thẳng thứ n. Thì số phần tử được thêm bằng số giao điểm được thêm cộng với 1.

- Số giao điểm được thêm là số giao điểm đường thẳng vừa kẻ cắt n-1 đường thẳng cũ, nghĩa là bằng n-1. Từ đó nhận được công thức truy hồi

$$s_n = s_{n-1} + \mathrm{n}, \mathrm{n} \geq 1$$
$$s_0 = 1$$



Phương pháp tìm nghiệm của hệ thức truy hồi

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-1}$$

Trong đó $c_1, c_2, \dots, c_k$ là các hằng số và $c_k \neq 0$

$a_n = 1.1\, a_{n-1}$, *truy hồi tuyến tính thuần nhất cấp* 1.

$a_n = a_{n-1} + a_{n-2}$ *Hệ thức truy hồi tuyến tính thuần nhất bậc* 2.

$a_n = a_{n-1} + a_{n-4}$ *Hệ thức truy hồi tuyến tính thuần nhất bậc* 4.

- $a_n = 2a_{n-1} + 1$ không thuần nhất
- $a_n = a_{n-1} + a_{n-4}^2$ không tuyến tính
- Hệ thức $a_n = \mathrm{n}a_{n-1} + a_{n-2}$

Cần tìm nghiệm $a_n$ cho dãy số $\{a_n\}$

Phương trình đặc trưng

$$r^n - c_1 r^{n-1} - c_2 r^{n-2} - \cdots - c_k r^{\mathrm{n}-k} = 0$$

- Xét trường hợp k =2

Định lý. Cho $c_1, c_2$ là các hằng số thực, giả sử phương trình: $r^2 - C_1 r - C_2 = 0$.

Có hai nghiệm phân biệt $r_1, r_2$. Khi đó dãy số $\{a_n\}$ là nghiệm của hệ thức truy hồi.

$$a_n = c_1 a_{n-1} + c_2 a_{n-2}$$

Khi và chỉ khi $$a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$$

Với các n=0, 1, 2,…và $\alpha_1$, $\alpha_2$ là hằng số.

Cho lần lượt n=0, 1 trong hệ thức truy hồi

$$\begin{cases} \bar{c}_0 = \alpha_1 + \alpha_2 \\ \bar{c}_1 = \alpha_1 r_1 + \alpha_2 r_2 \end{cases}$$

- Giải hệ phương trình trên ta được $$\alpha_1 = \frac{\bar{c}_1 - \bar{c}_0 r_2}{r_1 - r_2} \qquad \alpha_2 = \frac{\bar{c}_0 r_1 - \bar{c}_1}{r_1 - r_2} \qquad r_1 \neq r_2$$

Xác định nghiệm của dãy số fibonaxi

$$\begin{cases} a_0 = a_1 = 1 \\ a_n = a_{n-1} + a_{n-2} \quad n \geq 2 \end{cases}$$

Phương trình đặc trưng

Giải. $C_1 = C_2 = 1,\ \overline{C_0} = \bar{C_1} = 1$

$$r^2 - \mathrm{r} - 1 = 0$$

$$r_1 = \frac{1+\sqrt{5}}{2} \qquad r_2 = \frac{1-\sqrt{5}}{2}$$

$$\alpha_1 = \frac{1+\sqrt{5}}{2\sqrt{5}} \quad \alpha_2 = \frac{1-\sqrt{5}}{2\sqrt{5}}$$

Kết quả $$a_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} + \left(\frac{1-\sqrt{5}}{2}\right)^{\mathrm{n}+1}\right]$$

$r^2 - \mathrm{r} - 1 = 0 \qquad r_1 = 2, r_2 = -1$

$\alpha_1 = 3$

$\alpha_2 = -1$

$$a_n = \alpha_1.2^n + \alpha_2(-1)^n$$

$$\begin{cases} a_0 = 2, \quad a_1 = 7 \\ a_n = a_{n-1} + 2a_{n-2} \quad n = 2, 3, \ldots \end{cases}$$

$$\begin{cases} \alpha_1 + \alpha_2 = 2 \\ 2\alpha_1 - \alpha_2 = 7 \end{cases} \qquad a_n = 3.2^n - (-1)^n$$

$c_1, c_2$ *là các số thực* $c_2 \neq 0$ $\qquad r^2 - c_1\mathrm{r} - \mathrm{c}_2 = 0$

Có nghiệm kép $r_0$.

$a_n$ là nghiệm của hệ thức truy hồi $a_n = c_1 a_{n-1} + c_2 a_{n-2}$

Khi và chỉ khi $a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$

Định lý. Cho $c_1$, $c_2$,...., $c_\mathrm{n}$. *Giả sử phương trình đặc trưng*

$$r^k - C_1 r^{k-1} - C_2 r^{k-2} - \cdots . - c_k = 0$$

Có k nghiệm phân biệt $r_1, r_2, \ldots, r_\mathrm{k}$ *khi* đó $\{a_n\}$,

là *nghiệm hệ thức truy hồi*, $a_n = c_1 a_{n-1} + \cdots + c_k a_{n-k}$

Khi và chỉ khi $a_n = \alpha_1 \mathrm{r}_1^n + \cdots + \alpha_\mathrm{k} \mathrm{r}_{\mathrm{n-k}}^n$, .......với n=0, 1, 2, 3. $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_\mathrm{k}$ là *hằng số*

Nhận xét bước quan trọng trong việc xác định nghiệm của hệ thức truy hồi bậc $k$. Việc làm này không phải lúc nào cũng thực hiện được khi $k \geq 5$

$$r^k - C_1 r^{k-1} - C_2 r^{k-2} - \cdots . - c_k = 0$$

**LIỆT KÊ**

Bài toán hình chữ nhật la tinh.

Một hình chữ nhật la tinh trên S là một bảng p dòng, q cột, sao cho mỗi dòng của nó là một chỉnh hợp không lặp chập q của S và mỗi cột của nó là một chỉnh hợp không lặp chập p của S.

Thí dụ S= {1, 2, 3, 4, 5, 6, 7}

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 3 | 4 | 5 | 6 | 7 | 1 |
| 3 | 4 | 5 | 6 | 7 | 1 | 2 |

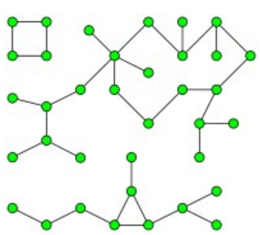
$$L(p,n) = n! \cdot K(p,n)$$

Gọi L(p, n) là số hình chữ nhật la tinh pxn, còn K(p,n) là số hình chữ nhật latinh chuẩn pxn. Ta có

Riordan J.(1946) đã chứng minh công thức

$$L(p,n) = \mathrm{n}! \cdot K(p,n)$$

Bài toán đếm số hình chữ nhật la tinh với số dòng nhiều hơn cho đến nay chưa được giải quyết. Người ta mới đưa ra một vài dạng tiệm cận( Erdos P.(1946), Yamamoto K. (1951)).



TOÁN RỜI RẠC

**NGUYỄN ĐÌNH CƯỜNG**

**BỘ MÔN KỸ THUẬT PHẦN MỀM**

**KHOA CNTT-ĐHNT**

**BÀI TOÁN TỒN TẠI**

Bài toán về 36 sĩ quan

Bài toán này được Euler đề nghị, nội dung nó như sau: Có một lần người ta triệu tập từ 6 trung đoàn mỗi trung đoàn 6 sĩ quan thuộc 6 cấp bậc khác nhau: thiếu úy, trung úy, thượng úy, đại úy, thiếu tá, trung tá về tham gia duyệt binh ở sử đoàn.

Dùng các chữ cái A, B, C, D, E, F để chỉ các phiên hiệu trung đoàn các chữ cái thường a, b, c, d, e, f.

| **Ab** | **Db** | **Ba** | **Cc** |
|---|---|---|---|
| Bc | Ca | Ad | Db |
| Cd | Bb | Dc | Aa |
| Da | Ac | Cb | Bd |

N=5

| **Aa** | **Bb** | **Cc** | **Dd** | **Ee** |
|---|---|---|---|---|
| Cd | De | Ea | Ab | Bc |
| Eb | Ac | Bd | Ce | Da |
| Be | Ca | Db | Ec | Ad |
| Dc | Ed | Ae | Ba | Cb |

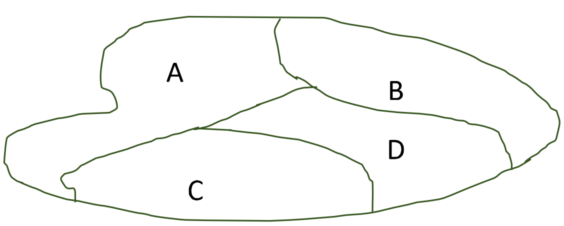
1960 Boce, Parker, Srikanda lời giải n=10.

Chỉ ra phương pháp xây dựng hình vuông la tinh trực giao cho mọi $n = 4k + 2$, với k>1

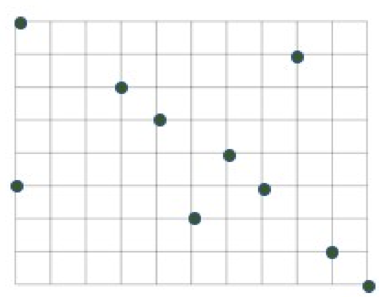
Bài toán 4 màu

Bài toán có thể phát biểu trực quan như sau:

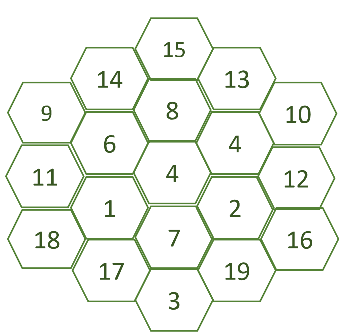
Chứng minh rằng mọi bản đồ trên mặt phẳng đều có thể tô bằng 4 màu sao cho không có hai nước láng giềng nào bị tô bởi cùng một màu




Bài toán chọn 2n điểm trên lưới nxn điểm



Hình lục giác thần bí



Phương pháp phản chứng

Giả thiết điều định chứng minh là sai, từ đó dẫn đến mâu thuẫn

Thí dụ.

Cho 7 đoạn thẳng có độ dài lớn hơn 10 và nhỏ hơn 100. Chứng minh rằng luôn tìm được 3 đoạn để có thể ghép thành 1 tam giác.

$$a_1 + a_2 \leq a_3$$
$$a_2 + a_3 \leq a_4$$
$$a_3 + a_4 \leq a_5$$
$$a_4 + a_5 \leq a_6$$
$$a_5 + a_6 \leq a_7$$

Bất đẳng thức cuối cùng mâu thuẫn với giả thiết các đoạn thẳng nhỏ hơn 100.

Thí dụ.

Chứng minh rằng không thể nối 31 máy vi tính thành một mạng sao cho mỗi máy nối với 5 máy khác.

Giải. Giả sử ngược lại là tìm được cách nối 31 máy cho cho mỗi máy được nối đúng với 5 máy khác. Khi đó số lượng kênh nối là 5x31/2 = 75.5.

**Nguyên lý Dirichlet**

Nguyên lý Dirichlet Nếu đem xếp nhiều $n$ đối tượng vào $n$ cái hộp, thì luôn tìm được một cái hộp chứa không ít hơn 2 đối tượng.

Nguyên lý Dirichlet tổng quát Nếu đem xếp $n$ đối tượng vào $k$ cái hộp, thì luôn tìm được một cái hộp chứa không ít hơn $\frac{n}{k}$ đối tượng.

Thí dụ 1. Trong số 367 người bao giờ cũng tìm được hai người có ngày sinh nhật giống nhau.

Thí dụ 2. Trong một phòng họp bao giờ cũng tìm được hai người có số người quen trong số những người dự họp là bằng nhau.

Thí dụ 3. Trong một tháng gồm 30 ngày một đội bóng chuyền thi đấu mỗi ngày ít nhất một trận, nhưng không chơi quá 45 trận. Hãy chứng minh rằng phải tìm được một giai đoạn gồm một số ngày liên tục nào đó trong tháng sao cho trong giai đoạn đó đội chơi đúng 14 trận.

Hệ đại diện phân biệt

Giả sử $s_1, s_2, \ldots, s_{\mathrm{m}}$ là một họ tập con của một tập hợp S( các $s_{\mathrm{i}}$ $không\ nhất\ thiết\ khác\ nhau$).

Ta gọi một bộ có thứ tự $a_1, a_2, \ldots, a_{\mathrm{m}}$. Là một hệ đại diện phân biệt của họ này nếu $a_{\mathrm{i}}\ \epsilon s_i$ $v$à $a_{\mathrm{i}} \neq a_{\mathrm{j}}.(i \neq j)$.

Hệ đại diện phân biệt được biết tắt là TRAN (Tranversal) và thành phần $a_{\mathrm{i}}$ của hệ được gọi là đại diện của tập con $s_{\mathrm{i}}$ ($i$=1, …, $m$)

Thí dụ. S= {1, 2, 3, 4, 5}, $S_1 = \{2, 5\}$, $S_2 = \{2, 5\}$, $S_3 = \{1, 2, 3, 4\}$, $S_4 = \{1, 2, 5\}$

có TRAN là (2, 5, 3, 1). Một TRAN khác của họ này là (5, 2, 4, 1)

Bài toán liệt kê

Thuật toán và độ phức tạp tính toán

Thuật toán giải bài toán đặt ra là một thủ tục xác định bao gồm một dãy hữu hạn các bước cần thực hiện để thu được lời giải bài toán.

Thí dụ

Cho 3 số nguyên a, b, c. Mô tả tìm số lớn nhất trong 3 số đã cho.

Giải. Thuật toán gồm các bước sau:

Bước 1. Đặt x = a;

Bước 2. Nếu b > x thì x = b;

Bước 3. Nếu c > x, thì đặt x=c;

Thuật toán Euclide

Đầu vào a và b là hai số nguyên dương

Đầu ra: Ước số chung lớn nhất của a và b

```
function gcd(a, b);
begin
    if a<b then <đổi chổ a và b>;
    while b ≠ 0 do
        begin
            lấy a chia b thu được a=bq + r, 0 ≤ r<b;
            a=b;
            b=r;
        end
        gcd=a;
end
```

Đánh giá độ phức tạp thuật toán

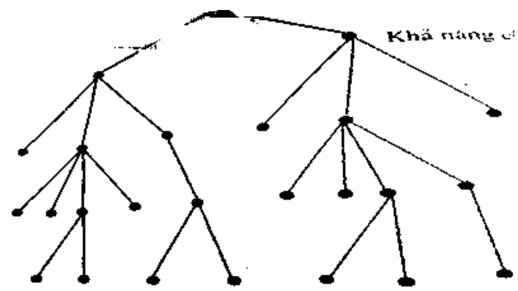
| **Dạng đánh giá** | **Tên gọi** |
|---|---|
| $\Theta(1)$ | Hằng số |
| $\Theta(lg\ lg\ n)$ | Log log |
| $\Theta(lg)$ | Logarithm |
| $\Theta(n)$ | Tuyến tính |
| $\Theta(nlgn)$ | N log n |
| $\Theta(n^2)$ | Bậc hai |
| $\Theta(n^3)$ | Bậc ba |
| $\Theta(n^m)$ | Đa thức |
| $\Theta(m^n), m \geq 2$ | Hàm mũ |
| $\Theta(n!)$ | Giai thừa |

| **Đánh giá** | **Thời gian tính , sec nếu n =** <br> **6** | **12** | **50** | **100** |
|---|---|---|---|---|
| **1** | $10^{-6}$ | $10^{-6}$ | $10^{-6}$ | $10^{-6}$ |
| $lg\ lg\ n$ | $10^{-6}$ | | | |
| $lgn$ | $3x10^{-6}$ | | | |
| $nlgn$ | $2x10^{-6}$ | | | |
| $n$ | $6x10^{-6}$ | | | |
| $n^2$ | $4x10^{-5}$ | | | |
| $n^3$ | $2x10^{-4}$ | | | |
| $2^n$ | $6x10^{-5}$ | | 36 năm | $4x10^6$ năm |

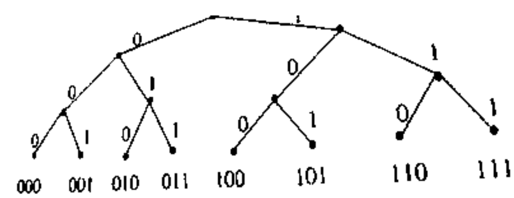
**Phương pháp sinh**

- Có thể xác định được một thứ tự trên tập các cấu hình tổ hợp cần liệt kê. Từ đó có thể xác định cấu hình đầu tiên và cấu hình cuối cùng trong thứ tự đã xác định. Xây dựng được thuật toán từ cấu hình chưa phải là cuối cùng trong thứ tự đã xác định.
- Xây dựng được thuật toán từ cấu hình chưa phải là cuối cùng đang có, đưa ra cấu hình kế tiếp nó.

procedure generate;
begin
    <Xây dựng cấu hình ban đầu>;
    Stop = false;
    while not stop do
    begin
        <Đưa ra cấu hình đang có>;
        Sinh_kế_tiếp;
    end
end



**Liệt kê các dãy nhị phân độ dài N**




- Tìm i đầu tiên( theo thứ tự i=n, n-1,…,1) thỏa mãn $b_i = 0$
- Gán lại $b_i$=1 và $b_i$=0 với tất cả $j > \mathrm{I}$

Ví dụ

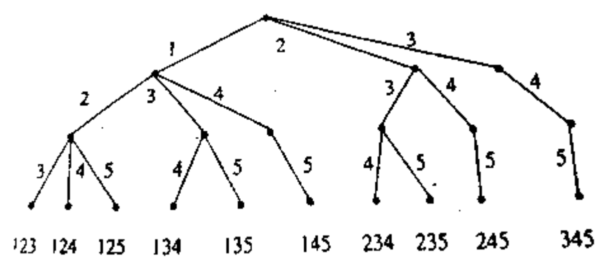
Xét dãy nhị phân độ dài 10: b=1101011111. Ta có i=5. Do đó, đặt $b_5 = 1$, và $b_i = 0$, i=6, 7, 8, 9, 10 ta thu được xâu nhị phân kế tiếp là 1101100000.

| b | p(b) |
|---|---|
| 0 0 0 | 0 |
| 0 0 1 | 1 |
| 0 1 0 | 2 |
| 0 1 1 | 3 |
| 1 0 0 | 4 |
| 1 0 1 | 5 |
| 1 1 0 | 6 |
| 1 1 1 | 7 |

procedure next_bit_string;
// sinh xâu nhị phân kế tiếp theo thứ tự từ điển từ xâu đang có
// xâu đang có $b_1 b_2 b_3 \ldots b_\mathrm{n}$
begin
    i=n;
    while $b_\mathrm{i}$=1 do
    begin
        $b_\mathrm{i} = 0$;
        i= i – 1;
    end
    $b_\mathrm{i}$=1;
end

```
var
n, i : integer;
b: array [1..20] of 0..1;
count word;
stop Boolean;

procedure init;
var  i: integer;
begin
        write('cho biết độ dài dãy nhị phân');
        readln(n);
        for i=1 to n do b[i]=0;
        stop=false;
        count;
end

procedure next_bit_string;
var i:integer;
begin
{sinh dãy nhị phân kế tiếp}
i=n;
while (i>=1) and (b[i]=1) do
begin
        b[i]=0;
        i=i-1;
end
If i<1 then stop =true
else b[i] =1;
end

begin
init;
while not stop do
begin
        // đưa ra dãy nhị phân hiện tại
        count=count +1;
        write(count:5, '.');
        for i=1 to n do write(b[i]:2); writeln;
        next_bit_string;
end
write('Gõ Enter để kết thúc…'); readln;
end
```

**Liệt kê các tập con m phần tử của tập n phần tử**




- Tìm từ bên phải dãy $a_1, a_2, a_3, a_4 \dots, a_m$ *phần tử* $a_1 \neq$ n-m+ i
- Thay $a_i$ bởi $a_{i+1}$
- Thay $a_j$ *bởi* $a_i + j - i$, *với* $j = i+1, i+2, \dots, m$

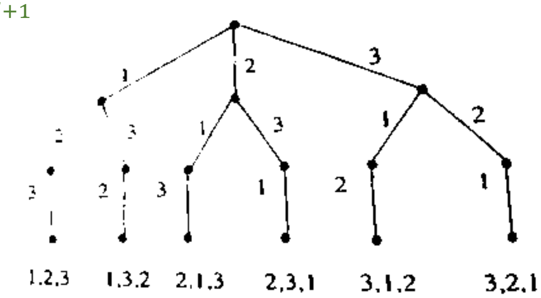
Thí dụ

n=6, m=4 . Giả sử có tập con (1, 2, 5, 6) cần xây dựng tập con kế tiếp nó trong thứ tự từ điển.

Ta có i=2 thay $a_2 = 3, a_3 = 4, a_4 = 5$. Ta có tập con kế tiếp (1, 3, 4, 5)

procedure next_combination;
begin
        i=m;
        while $a_i$ = n-m + I do i=i -1;
        $a_i$= $a_i$+1;
        for j=i+1 to m do $a_j$= $a_i + j - i$;
end

## Liệt kê các hoán vị tập n phần tử




- Tìm từ phải qua trái hoán vị đang có chỉ số $j$ đầu tiên thỏa mãn $a_j < a_{j+1}$

(nói cách khác j là chỉ số lớn nhất thỏa mãn $a_j < a_{j+1}$)

- Tìm $a_k$ là số nhỏ nhất còn lớn hơn $a_j$ trong các số bên phải $a_j$
- Đổi chổ $a_j$ *với* $a_k$
- Lật ngược đoạn $a_j$ đến $a_n$

Thí dụ

Giả sử đang có hoán vị (3, 6, 2, 5, 4, 1) cần xây dựng tập con kế tiếp nó trong thứ tự từ điển.

Ta có j=3 thay ($a_3 = 2 < a_4 = 5$) . *Số nhỏ nhất còn lớn hơn* $a_3$ *trong các số bên phải của* $a_3$ là $a_5 = 4$. Đổi chổ $a_3$ và $a_5$ ta thu được (3, 6, 4, 5, 2, 1) và cuối cùng lật ngược đoạn thứ tự $a_4\ a_5\ a_6$ *ta thu* được *hoán vị kế tiếp* (3, 6, 4, 1, 2, 5).

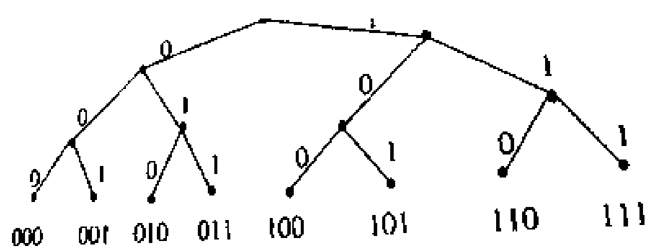
procedure next_permutation;<br>
// sinh hoán vị kế tiếp thứ tự từ điển<br>
begin<br>
// tìm j là chỉ số lớn nhất thỏa $a_j < a_{j+1}$<br>
j=n-1;<br>
while $a_j > a_{j+1}$ *do* $j = j - 1$;<br>
// tìm $a_k$ *là số nhỏ nhất còn lớn hơn* $a_j$ *ở bên phải* $a_j$<br>
k=n;<br>
while $a_j > a_k$ *do* $k = k - 1$;<br>
swap($a_j$, $a_k$);<br>
// lật ngược đoạn $a_{j+1}$ đến $a_n$<br>
r=n;<br>
s=j+1;<br>
while r>s do<br>
begin<br>
        swap($a_r$, $a_s$);<br>
        r=r-1;<br>
        s=s+1;<br>
end<br>
end

**Thuật toán quay lui**

procedure Try(i: integer);
var j:integer;
begin
for j=1 to $n_i$ $do$
if <chấp nhận j> then
begin
<xác nhận $x_i$ $theo$ $j$>;
if i= n then <ghi nhận một cấu hình>
else
Try (i+1);
end
end

begin
Try(1);
end

procedure Try(i:integer);
var j:integer;
begin
for j=0 to 1 do
begin
b[i]=j;
if i=n then Result else Try(i+1);
end
end

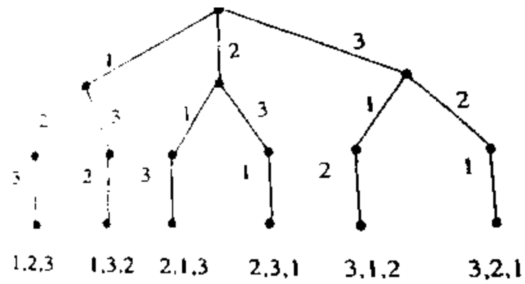



begin
Try(1);
end

**Liệt kê các hoán vị từ 1 đến N**

Biểu diễn hoán vị dưới dạng $p_1, p_2, p_3, p_n$, trong đó $p_i$ nhận giá trị từ 1 đến n và $p_i \neq p_j$ với i≠ $j$. Các giá trị từ 1 đến n lần lượt đề cử cho $p_i$, trong đó giá trị $j$ được chấp nhận nếu nó chưa được dung. Vì thế cần phải ghi nhớ đối với mỗi giá trị $j$ xem nó được dùng hay chưa. Điều này thực hiện nhờ vào một dãy biến logic $b_j$ , trong đó $b_j$ bằng *true* nếu $j$ chưa được dùng. Các biến này cần phải được khởi gán giá trị *true . Sau khi* gán $j$ cho $p_i$ *cần* ghi nhận false cho $b_j$ và phải gán lại true khi thực hiện xong Result hay Try(i+1).

```
procedure Try(i : integer);
var j: integer;
vegin
          for j=1 to n do
          if b[j] then      { chấp nhận j }
           begin
                    p[i] = j;
                    b[j] = false; {ghi nhận trạng thái mới }
                    if i= n then Result else Try(i + 1);
                    b[j]=true;
          end
end
```

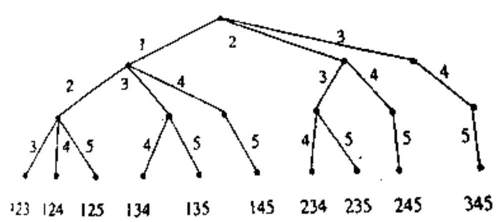



**Liệt kê các tổ hợp chập m của {1, 2, …, n}**

Biểu diễn tổ hợp dưới dạng $C_1C_2 \dots C_m$, trong đó $1 \leq C_1 < C_2 < \dots < C_m \leq n$

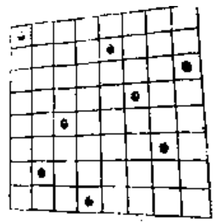
Từ đó suy ra các giá trị đề cử cho $C_i$ là từ $C_{i-1} + 1$ đến n-m + i. Để điều này đúng cho trường hợp i=1, cần thêm vào $C_0$ $với C_0$=0.

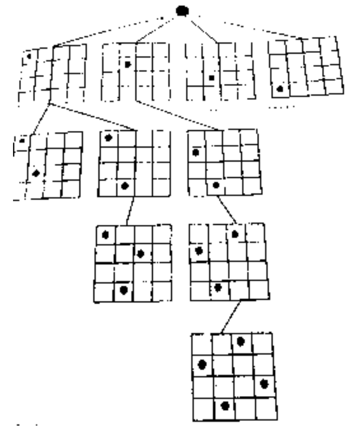
```
procedure Try(i: integer);

var j: integer;

begin

          for j= c[i-1] + 1 to n-m+i do

          begin

                    c[i]=j;

                    if i=m then Result else Try(i+1);

          end

end
```



**Bài toán xếp 8 quân hậu trên bàn cờ**

```
procedure Try(i: integer);
var j: integer;
begin
          for j=1 to n do
          if a[j] and b[i+j] and c[i-j] then
          begin       {chấp nhận j}
                      x[i] = j;
                      { ghi nhận trạng thái mới }
                      a[j] = false;
                      b[i+j] = false;
                      c[i-j] = false;
                      if i= n then Result  else Try(i+1);
                      // Trả lại trạng thái cũ
                      a[j]= true;
                      b[i+j] = true;
                      c[i-j]=true;
          end
end
```





Số cách xếp quân hậu trên bàn cờ theo N

| N | 4 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|
| H | 2 | 40 | 92 | 352 | 724 | 2680 | 14200 | 73712 | 365596 |

**BÀI TOÁN TỐI ƯU**

Hãy lựa chọn trong số các cấu hình tổ hợp chấp nhận được cấu hình có giá trị sử dụng tốt nhất. Các bài toán như vậy gọi là bài toán tối ưu tổ hợp. Dưới dạng tổng quát bài toán tối ưu tổ hợp có thể phát biểu như sau:

Tìm cực tiểu (hay cực đại) của phiếm hàm

$$f(x) \rightarrow \min(max)$$

Với điều kiện $x \in D. Trong$ đó $D$ là $tập$ $hữu$ $hạn$ $phần$ $tử$.

Hàm $f$(x) , được gọi là hàm mục tiêu của bài toán, mỗi phần tử $x \in D$. Được gọi là một phương án còn tập $D$ gọi là tập phương án của bài toán.

Phương án $x^* \in D$ đem lại giá trị nhỏ nhất ( lớn nhất) cho hàm mục tiêu được gọi là phương án tối ưu, khi đó $f^* = f(x^*)$

**Bài toán người du lịch**

Một người du lịch muốn đi tham quan n thành phố

$T_1, T_2, T_3, \ldots, T_n. Xuất\ phát$ từ một thành phố nào đó người du lịch muốn đi qua các thành phố còn lại, mỗi thành phố đúng một lần, rồi quay trở lại thành phố xuất phát. Biết $c_{ij}$ là $chi\ phí$ $đi$ $từ$ $thành\ phố\ i$ $tới$ $thành\ phố\ j$ hãy tìm hành trình (Một cách đi thỏa mãn điều kiện đặt ra). Với tổng chi phí là nhỏ nhất.

Hành trình

$$T_{\Pi(1)} \rightarrow T_{\Pi(2)} \rightarrow \ldots \rightarrow T_{\Pi(n)} \rightarrow T_{\Pi(1)}$$

Với một hoán vị $\Pi = (\Pi(1), \Pi(2), \ldots, \Pi(\mathrm{n}))$của n số tự nhiên

$$f(\Pi) = c_{\Pi(1),\Pi(2)} + \cdots + c_{\Pi(\mathrm{n}-1),\Pi(\mathrm{n})} + c_{\Pi(\mathrm{n}),\Pi(1)}$$

Khi đó bài toán người du lịch, có thể phát biểu dưới dạng sau

$$\min\{ f(\pi) : \pi \in \pi\}$$

Có thể thấy rằng tổng số hành trình của người du lịch là $n!$

**Bài toán cái túi**

X =($x_1, x_2, \ldots, x_n$). $Trong$ đó $x_j = 1$ $nghĩa$ là đồ $vật$ $thứ$ $j$ được $mang$ $theo$ $v$à $x_j$=0 có nghĩa là trái lại.

Với phương án $x$, giá trị đồ vật đem theo là

$$f(x) = \sum_{j=1}^{n} c_j x_j$$

Tổng trọng lượng của đồ vật mạng theo là

$$g(x) = \sum_{j=1}^{n} a_j x_j$$

Trong các vecto nhị phân độ dài n thỏa mãn điều kiện $g$(x)<b , hãy tìm vecto $x^*$ $cho$ $giá$ $trị$ $nhỏ$ $nhất$của hàm mục tiêu $f$(x)

$$min\ \{f(\mathrm{x}) : g(\mathrm{x}) \leq b\}$$

**Bài toán cho thuê máy**

Một ông chủ có một cái máy để cho thuê. Đầu tháng ông nhận được yêu cầu thuê máy của m khách hàng. Mỗi khách hàng i sẽ cho biết tập $N_i$ *các* ngày trong tháng sử dụng máy($i = 1, 2, 3, \dots, m$).
Ông chủ chỉ có quyền hoặc là từ chối yêu cầu của khách hàng i, hoặc là nếu nhận thì phải bố trí đúng những ngày mà khách hàng này yêu cầu. Hỏi rằng ông chủ phải tiếp nhận các yêu cầu của khách như thế nào để cho tổng số ngày sử dụng máy tính là lớn nhất.

Ký hiệu $I$={1, 2, 3, …, $m$} là tập chỉ số khách hàng, S là tập hợp các tập con của I. Khi đó tập hợp các phương án cho thuê máy là $D$= {$J \subset S$: $N_k \cap N_p = \emptyset, \forall k \neq p, k, p \epsilon J$ và với mỗi phương án J$\epsilon D$

$$f(J) = \sum_{j \in J} |N_j|$$

Sẽ là tổng số ngày sử dụng máy tính theo phương án đó. Bài toán đặt ra có thể phát biểu dưới dạng bài toán tối ưu tổ hợp sau:

$$max\ \{f(J): J \in D\}$$

**Bài toán phân công**

Có n công việc và n thợ. Biết $C_{ij}$ *là chi phí cần trả để thợ i hoàn thành công việc j* $(i, j\ 1, 2, \dots, n)$.
*Cần phải thuê thợ sao cho các công việc đều hoàn thành* và mỗi thợ chỉ thực hiện một công việc. Mỗi công việc chỉ do một thợ thực hiện. Hãy tìm cách thuê sao cho tổng chi phí thuê thợ là nhỏ nhất.

| Công việc | Thợ thực hiện |
|---|---|
| 1 | $\pi(1)$ |
| 2 | $\pi(2)$ |
| … | |
| N | $\pi(N)$ |

Tương ứng với một hoán vị $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ của n số tự nhiên 1, 2, …, n. Chi phí theo phương án bố trí trên là

$$f(\pi) = C_{\pi(1),1} + C_{\pi(2),2} + \cdots + C_{\pi(n),n}$$

Bài toán đặt ra, được dẫn về bài toán tối ưu tổ hợp

$$min\ \{f(\pi):\ \pi\ \in\ \Pi\ \}$$

**Bài toán lập lịch**

Mỗi chi tiết trong số n chi tiết $D_1$, $D_2$,..., $D_n$ cần phải được lần lượt gia công trên m máy $M_1$, $M_2$, ..., $M_m$. Thời gian gia công chi tiết $D_i$ trên máy $M_j$ là $t_{ij}$.

Một lịch gia công các chi tiết trên máy tương ứng với một hoán vị $\pi = \big(\pi(1), \pi(2), \dots, \pi(n)\big)$. Của n số tự nhiên 1,2,...,$n$.

Thời gian hoàn thành theo lịch được tính bởi.

$$f(\pi) = \sum_{j=1}^{n-1} C_{\pi(j),\pi(j+1)} + \sum_{k=1}^{m} t_{k,\pi(n)}$$

Trong đó $c_{ij}$= $S_j - S_i$, $S_j$ thời điểm bắt đầu thực hiện gia công chi tiết $j$ $(i, j = 1, 2, , \dots, n)$.

$c_{ij}$ tổng thời gian gián đoạn( được tính khi bắt đầu gia công chi tiết $i$ ) $gây\ ra$ bởi chi tiết $j$ khi nó được gia công sau chi tiết $i$ trong lịch gia công.

$$min\ \{f(\pi)\}: \pi\ \in\ \Pi\}$$

Thuật toán nhánh cận

procedure Try(k);
// phát triển phương án bộ phận $a_1$, $a_2$, $a_3$,..., $a_{k-1}$
theo thuật toán quay lui có kiểm tra cận dưới trước khi tiếp tục phát triển phương án

begin
for $a_k \in A_k\ do$
if <chấp nhận $a_k$ > then
begin
$X_k = a_k$;
if k=n then <Cập nhật kỷ lục>
else
if g $(a_1, a_2, ...., a_k) \le \bar{f}$ $then$ Try (k+1);
end
end

**Bài toán cái túi**

Mô hình toán học có dạng như sau

$$f^* = \max\{f(x) = \sum_{j=1}^{n} C_j x_j : \sum_{j=1}^{n} C_j x_j \le b, x_j \in Z_+, \qquad j = 1, 2, 3, \dots, n\}$$

Kí hiệu D là tập phương án

$$D=\{x=(x_1, x_2, \dots, x_n)\ \sum_{j=1}^{n} a_j x_j \le b, x_j \in Z_+, j=1, 2, \dots, n\}$$

Xếp thứ tự

$$C_1/a_1 \ge C_2/a_2 \dots . \ge C_n/a_n$$

Tìm

$$g^* = \max\{f(x) = \sum_j^n c_j\, x_j : \sum_j^n c_j\, x_j \le b,\ x_j \ge 0, j = 1,2,3 \dots . . , n\}$$

Phương án tối ưu của bài toán là vecto $\bar{x} = (\overline{x_1}, \overline{x_2}, \dots . , \overline{x_n})$ các thành phần được xác định bởi công thức

$\overline{x_1} = b/a_1$ , $\overline{x_2} = \overline{x_3} = \cdots = \overline{x_2} = 0$ và giá trị tối ưu là $g^* = C_1 b_1/a_1$

Chứng minh

Xét $x=(x_1, x_2, \dots, x_n)$ là một phương án tùy ý $x_j \ge 0$, ta suy ra

$$c_j/x_j \ge (C_1/a_1)\, a_j/x_j, \quad j=1, 2, \dots, n$$

$$\sum_{j=1}^{n} C_j x_j \le \sum_{j=1}^{n} (C_1/a_1)\, a_j x_j = (C_1/a_1) \sum_{j=1}^{n} a_j x_j = (C_1/a_1) \sum_{j=1}^{n} a_j x_j \le \left(\frac{C_1}{a_1}\right) b = g^*$$

Mệnh đề được chứng minh

Giả sử có phương án bộ phận cấp k; $(u_1, u_2, u_3, \dots, u_k)$. Khi đó giá trị sử dụng các đồ vật trong túi là

$$\sigma_k = c_1\, u_1 + c_2\, u_2 + \dots . + c_k\, u_k$$

Và trọng lượng còn lại của cái túi là

$$b_k = b - a_1 u_1 + a_2 u_2 + \dots + a_k u_k$$

$$max\ \{f(x) : x \in D, x_j = u_j, j = 1, 2, \dots . , k\}$$

$$= max\{\sigma_k + \sum_{j=k+1}^{n} C_j x_j : \sum_{j=k+1}^{n} a_j x_j \le b_k, x_j \in Z_+, j=k+1, k+2, \dots, n\}$$

$\le \sigma_k + \max\ \{\sum_{j=k+1}^{n} c_j x_j : \sum_{j=k+1}^{n} a_j x_j \le b_k, x_j \ge 0, j = k+1, k+2, \dots, n\ \}$

(theo mệnh đề giá trị số hạng thứ hai là $C_{k+1} b_k/a_{k+1}$ )

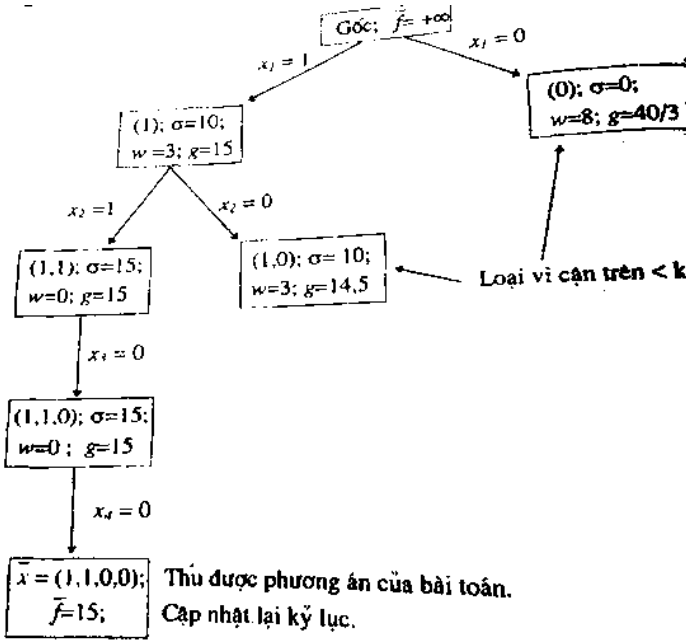
$= \sigma_k + C_{k+1} b_k/a_{k+1}$

Vậy ta có thể tính cận cho phương án bộ phận bởi công thức $(u_1, u_2, u_3, \dots, u_k)$

$$g(u_1, u_2, u_3, \dots, u_k) = \sigma_k + C_{k+1} b_k/a_{k+1}$$

Ví dụ: giải bài toán cái túi theo thuật toán nhánh cận

$$f(x) = 10x_1 + 5x_2 + 3x_3 + 6\ 10x_4 \quad \rightarrow max$$
$$5\ x_1 + 3\ x_2 + 2\ x_3 + 4\ x_4 \leq 8$$
$$x_j \in Z_+ ,\ j=1, 2, 3, 4.$$




Kết thúc ta thu được phương án tối ưu

$x^* = (1, 1, 0, 0)$ và giá trị tối ưu $f^* = 15$

```
procedure branch_and_bound(i : integer);
var j, t: integer;
begin

          t=trunc ((w-weight)/a[i]);
          for j= t downto 0 do
          begin
                    x[i]=j;
                    weight= weight + a[i]*x[i];
                    cost = cost + c[i]*x[i];
                    if i=n then ghinhanketqua
                    else
                    if cost +  c[i+1]*(w-weight)/a[i+1] > fopt
                    then branch_and_bound(i+1);
                    weight=weight – a[i]*x[i];
                    cost = cost – c[i] * x[i];
          end
end
```

Bài toán người du lịch

Tìm cựu tiểu của hàm

$$f(x_2, x_3, \ldots, x_n) = c[1, x_2] + c[x_2, x_3] + \ldots + c[x_{n-1}, x_n] + c[x_n, 1] \rightarrow \min$$

Với điều kiện

$(x_2, x_3, \ldots, x_n)$ là hoán vị của các số 2, 3, …,$n$.

Ký hiệu

$$c_{min} = \min\{c[i,j], i,j = 1,2,\ldots,n, i \neq j\}$$

Giả sử ta có phương án bộ phận $(u_1, u_2, u_3, \ldots, u_k)$. Phương án này tương ứng với hành trình bộ phận qua $k$ thành phố

$$T_1 \rightarrow T(u_2) \rightarrow \ldots \rightarrow T(u_{k-1}) \rightarrow T(u_k)$$

Vì vậy, chi phí phải trả theo hành trình bộ phận qua $k$ thành phố

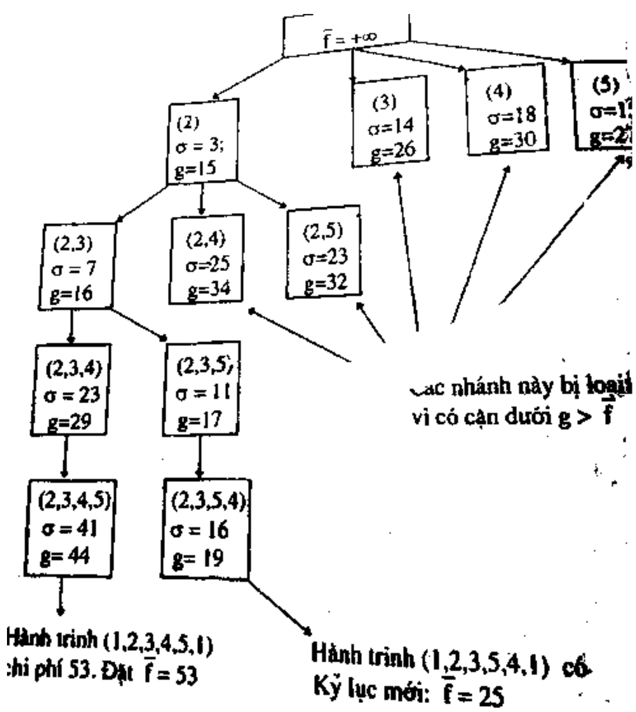
$$\sigma = c[1, u_2] + c[u_2, u_3] + \ldots + c[u_{k-1}, u_k]$$

Cận dưới phương án bộ phận $g(u_1, u_2, u_3, \ldots, u_k) = \sigma + (n-k+1)c_{min}$

Giải bài toán người du lịch

$$c = \begin{vmatrix} 0 & 3 & 14 & 18 & 15 \\ 3 & 0 & 4 & 22 & 20 \\ 17 & 9 & 0 & 16 & 4 \\ 6 & 2 & 7 & 0 & 12 \\ 9 & 15 & 11 & 5 & 0 \end{vmatrix}$$

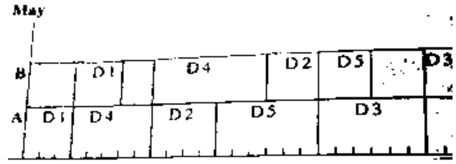
$$C_{min} = 3$$




```
procedure Try(i:integer);
var j: integer;
begin
for j=2 to n do
      if chuaxet[j] then
         begin
              a[i]=j;
              chuaxet [j] = false;
              chuaxet[j] = false;
              can = can + c[a[i-1], a[i]];
              if i=n then Ghinhan
              else
              if Can + (n-i+1)*Cmin < fopt then Try (i+1);
              can = can - c[a[i-1], a[i]];
              chuaxet [j] = true;
         end
end
```

**Bài toán lập lịch trên hai máy**

| Máy \ Chi tiết | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ |
|---|---|---|---|---|---|
| A | 3 | 4 | 6 | 5 | 6 |
| B | 3 | 3 | 2 | 7 | 3 |




Thuật toán JOHNSON

1. Chia các chi tiết thành 2 nhóm: $N_1$ $gồm\ các\ chi\ tiết\ D_i\ thỏa\ mãn\ a_i \leq b_i$ . $Nhóm\ 2\ \ D_i\ thỏa\ mãn\ a_i > b_i$

2. Sắp xếp các chi tiết $N_1$ theo chiều tăng của các $a_i$ và sắp xếp chi tiết $N_2$ theo chiều giảm $b_i$

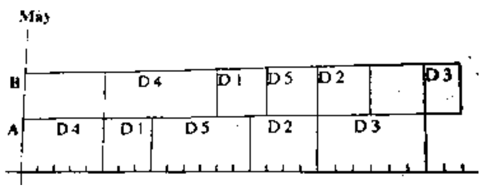
3. Nối $N_2$ vào đuôi $N_1$. Dãy thu được sẽ là lịch gia công tối ưu.

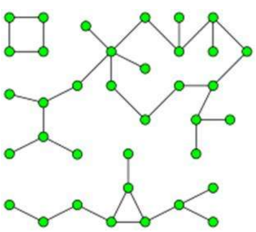
Các kết quả được tính sau:

- Chia nhóm $N_1$= $\{D_1, D_4\}$ $N_2$= $\{D_2, D_3, D_5\}$
- Sắp xếp các chi tiết $N_1$ theo chiều tăng của các $a_i$ và sắp xếp chi tiết $N_2$ theo chiều giảm $b_i$

$N_1$= $\{D_1, D_4\}$ $N_2$= $\{D_2, D_5, D_3\}$

- $\pi = (D_1, D_4, D_2, D_5, D_3)$

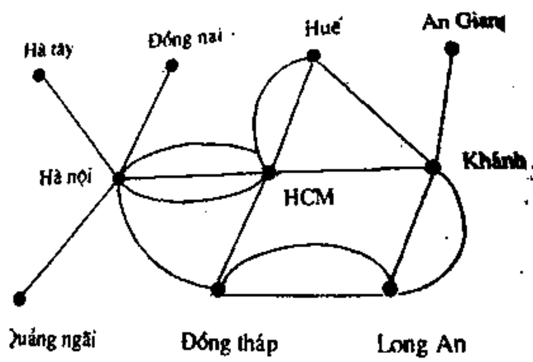


# LÝ THUYẾT ĐỒ THỊ

Định nghĩa 1. Đơn đồ thị vô hướng

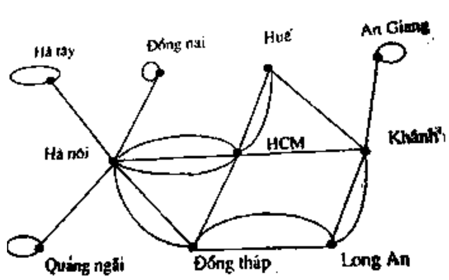
G=$(V, E)$, *bao gồm* $V$ *là tập các đỉnh*, *và* $E$ là tập các cặp không có thứ tự gồm hai phần tử khác nhau của V gọi là các cạnh.

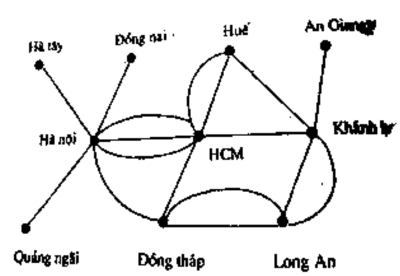



Định nghĩa 2. Đa đồ thị vô hướng

G=$(V, E)$, *bao* gồm V là các tập đỉnh, và E là họ các cặp không có thứ tự gồm 2 phần tử khác nhau của V gọi là các cạnh. Hai cạnh e1, e2 được gọi là cạnh lặp nếu chúng cùng tương ứng với một cặp đỉnh.

Định nghĩa 3. Giả đồ thị vô hướng G=(V, E) bao gồm V là tập đỉnh, và E là họ các cặp không có thứ tự gồm hai phần tử ( không nhất thiết phải khác nhau) của V gọi là các cạnh. Cạnh $e$ được gọi là khuyên nếu nó có dạng $e = (u, u)$




Định nghĩa 4. Đơnđồ thị vô hướng G=(V, E) bao gồm V là tập đỉnh, và E tập các cặp có thứ tự gồm hai phần tử khác nhau của V gọi là các cung.

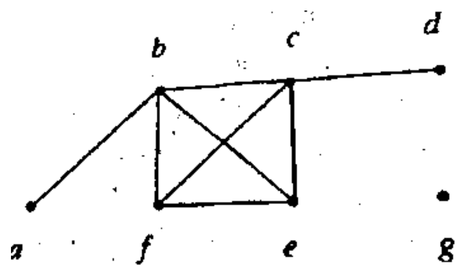
Định nghĩa 5. Đa đồ thị có hướng

$G = (V, E)$*bao gồm* V là tập các đỉnh, và E là họ các cặp có thứ tự gồm hai phần tử khác nhau của V gọi là các cung. Hai cung $e_1, e_2$ *tương* ứng với cùng một đỉnh gọi là cung lặp.

**Các thuật ngữ cơ bản**

Định nghĩa 6. Hai đỉnh $u$ và $v$ của đồ thị vô hướng G được gọi là kề nhau nếu $(u, v)$ là cạnh của đồ thị G. Nếu e=(u, v) là cạnh của đồ thị thì ta nói cạnh này là liên thuộc với 2 đỉnh u và v, hoặc cũng nói là cạnh e là nối đỉnh u và đỉnh v, đồng thời các đỉnh u và v sẽ được gọi là các đỉnh đầu của cạnh (u, v).

Định nghĩa 7. Ta gọi bậc của đỉnh v trong đồ thị vô hướng là số cạnh liên thuộc với nó và sẽ kí hiệu là deg(v)




Xét đồ thị trong hình 1, ta có
deg(a) = 1, deg(b) = 4, deg(c) = 4, deg(f) = 3, deg(d) = 1, deg(e) = 3, deg(g) = 0

Định lý 1. Giả sử $G = (V, E)$ là đồ *thị vô hướng với m* ạnh. *Khi* đó

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

Chứng minh. Rõ ràng mỗi cạnh

e= $(u, v)$ được *tính* một lần trong deg(u)và một lần trong deg(v). Từ đó suy ra tổng các bậc của các đỉnh bằng hai lần số cạnh

Thí dụ 2

Đồ thị n đỉnh và mỗi đỉnh có bậc là 6 có bao nhiêu cạnh ?

Giải: Theo định lý 1, ta có 2m=6n. Từ đó suy ra số cạnh của đồ thị là 3n.

Hệ quả. Trong đồ thị vô hướng, số đỉnh bậc lẻ( nghĩa là có bậc là số lẻ) là một số chẵn.

Chứng minh. Thực vậy, gọi $O$ *và* $U$ *tương ứng* là *tập đỉnh* bậc *lẻ* và *tập đỉnh bậc chẵn của* đồ *thị*. *Ta có*

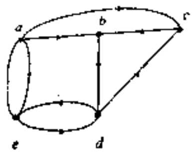
$$2m = \sum_{v \in V} \deg(v) = \sum_{v \in O} \deg(v) + \sum_{v \in U} \deg(v)$$

Định nghĩa 8.

Nếu $l$ là cung của đồ thị có hướng G thì ta nói hai đỉnh u và v là kề nhau, và nói cung (u, v) nối đỉnh u với đỉnh v hoặc cũng nói cung này là đi ra khỏi đỉnh u và đi vào đỉnh v. Đỉnh u(v)sẽ được gọi là đỉnh đầu ( cuối) của cung (u, v).

Định nghĩa 9

Ta gọi bán bậc ra (bán bậc vào) của đỉnh v trong đồ thị có hướng là số cung của đồ thị đi ra khỏi nó (đi vào nó) và kí hiệu là $deg^{+}(v)$, $deg^{-}(v)$




Định lý 2. Giả sử $G = (V, E)$ là đồ thị có hướng. Khi đó

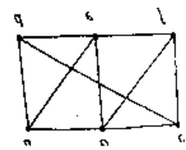
$$\sum_{v \in V} deg^+(v) = \sum_{v \in V} deg^-(v) = |E|$$

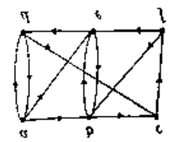
Đường đi, chu trình. Đồ thị liên thông

Định nghĩa 10. Đường đi độ dài n từ đỉnh u đến đỉnh v, trong đó n là số nguyên dương, trên đồ thị vô hướng $G = (V, E)$.

$x_0, x_1, \ldots, x_{n-1}, x_n$

Trong đó $u = x_0, v = x_n, (x_i, x_{i+1}) \in E, \quad i = 0, 1, 2, \ldots, n-1$





Đường đi nói trên có thể biểu diễn dưới dạng dãy các cạnh

$(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)$

Đỉnh u gọi là đỉnh đầu, còn đỉnh v gọi là đỉnh cuối của đường đi. Đường đi có đỉnh đầu trùng với đỉnh cuối (tức là u=v) được gọi là chu trình. Đường đi hay chu trình được gọi là đơn nếu như không có cạnh nào được lặp lại.

Định nghĩa 11.

Đường đi độ dài n từ đỉnh u đến đỉnh v, trong đó n là số nguyên dương, trên đồ thị có hướng $G = (V, A) là\ dãy$

$x_0, x_1, \ldots, x_{n-1}, x_n$

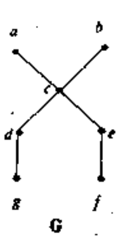
Trong đó $u = x_0, v = x_n, (x_i, x_{i+1}) \in A, \quad i = 0, 1, 2, \ldots, n-1.$

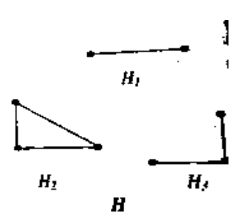
Đường đi nói trên còn có thể biểu diễn dưới dạng dãy các cung: $(x_0, x_1), (x_1, x_2), \ldots, (x_{n-1}, x_n)$

Đỉnh u được gọi là đỉnh đầu, còn đỉnh v gọi là đỉnh cuối của đường đi. Đường đi có đỉnh đầu trùng với đỉnh cuối (tức là u=v) được gọi là chu trình. Đường đi hay chu trình được gọi là đơn nếu như không có cung nào bị lặp lại.

Định nghĩa 12.

Đồ thị vô hướng $G = (V, E) được\ gọi\ là\ liên\ thông\ nếu\ luôn\ tìm\ được\ đường\ đi\ giữa\ hai\ đỉnh\ bất\ kì\ của\ nó$




Định nghĩa 13. Ta gọi đồ thị con của đồ thị $G = (V, E)$ là đồ thị $H = (W, F)$

Trong đó $W \subseteq V$ và $F \subseteq E$

Định nghĩa 14.

Đỉnh v được gọi là đỉnh rẻ nhánh nếu việc loại bỏ v cùng với các cạnh liên thuộc với nó khỏi đồ thị làm tang số thành phần liên thông của đồ thị. Cạnh e được gọi là cầu nếu việc loại bỏ nó khỏi đồ thị làm tăng số thành phần liên thông của đồ thị.

Định nghĩa 15.

Đồ thị có hướng $G = (V, A)$ được gọi là liên thông mạnh nếu luôn tìm được đường đi giữa hai đỉnh bất kì của nó.

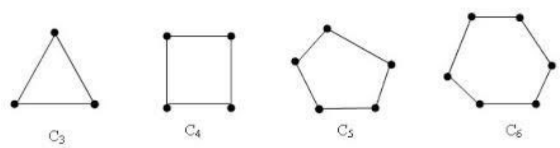
Định nghĩa 16.

Đồ thị có hướng $G = (V, A)$ được gọi là liên thông yếu nếu đồ thị vô hướng tương ứng với nó là đồ thị vô hướng liên thông.

Định lý 3.

Đồ thị vô hướng liên thông là định hướng được khi và chỉ khi mỗi cạnh của nó nằm trên ít nhất một chu trình.

Một số dạng đồ thị đặc biệt

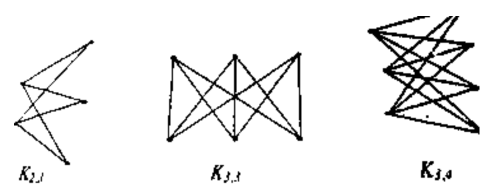



Đồ thị hai phía.

Đơn đồ thị $G = (V, E)$ *được gọi là hai phía* Nếu tập đỉnh V của nó có thể phân hoạch thành 2 tập X và Y. Sao cho mỗi cạnh của đồ thị chỉ nối một đỉnh nào đó trong X với một đỉnh nào đó trong Y. Khi đó ta sẽ sử dụng kí hiệu $G = (X \cup Y, E)$ để chỉ đồ thị hai phía với tập đỉnh X∪ $Y$.

Định lý 4

Đơn đồ thị là đồ thị hai phía và chỉ khi nó không chứa chu trình độ dài lẻ.




Đồ thị hai phía.

Đơn đồ thị

$G(V, E)$ *được gọi là đồ thị phẳng* nếu ta có thể vẽ nó trên mặt phẳng sao cho các cạnh của nó không cắt nhau ngoài ở đỉnh . Cách vẽ như vậy sẽ được gọi là biểu diễn phẳng của đồ thị.

Định lý 5(Kuratovski)

Đồ thị là phẳng khi và chỉ khi nó không chứa đồ thị con đồng cấu với $K_{3,3}$ *hoặc* $K_5$

Định lý 6 (công thức Euler)

Giả sử G là đồ thị phẳng liên thông với n đỉnh, m cạnh. Gọi m là số miền của mặt phẳng bị chia bởi biểu diễn phẳng của G. Khi đó

$$r = m - n + 2$$

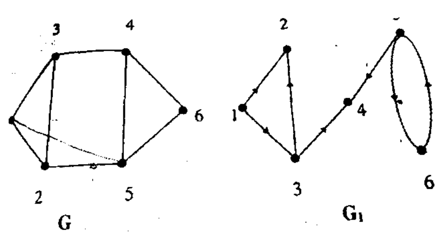
Thí dụ

Cho G là đồ thị phẳng liên thông với 20 đỉnh, mỗi đỉnh đều có bậc là 3. Hỏi mặt phẳng bị chia làm bao nhiêu phần bởi biểu diễn phẳng của đồ thị G.

Giải. Do mỗi đỉnh của đồ thị đều có bậc là 3, nên tổng bậc của đỉnh là 3x20=60. Từ đó suy ra số cạnh của đồ thị m=60/2=30. Vì vậy theo công thức Euler, số miền cần tìm là

$$R = 30 - 20 + 2 = 12$$

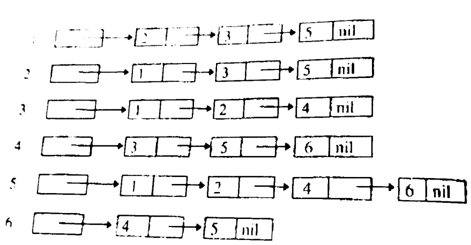
**BIỂU DIỄN ĐỒ THỊ TRÊN MÁY TÍNH**



$G = (V, E), với\ tập\ đỉnh\ V = \{1, 2, \dots, n\}.$
$Tập\ cạnh\ E = \{e_1, e_2, e_3, \dots, e_n\}$

A= $\{a_{ij} : i, j = 1, 2, \dots, n\}$

$a_{ij}$=0 nếu $(i, j) \notin E$ và $a_{ij}$=1, nếu $(i, j) \in E$ $\quad i, j = 1, 2, 3 \dots n$



$$A_{i_j} = \begin{vmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

Tùy từng trường hợp cụ thể c[i , j] = $\theta$ *có thể* được *đặt bằng các giá trị sau* $0, +\infty, -\infty$

**Tìm kiếm  theo chiều sâu trên đồ thị**

```
procedure DFS(v);
(* Tìm kiếm theo chiều sâu bắt đầu từ đỉnh v, các biến Chuaxet, Ke là biến toàn cục *)
begin
        Thăm_đỉnh(v);
        Chuaxet[v] = false;
        for u ∈ kề(v) do
        if chuaxet[u] then DFS(u);
end

begin
        for v ∈ V do chuaxet[v]=True;
        for v ∈ V do
                if Chuaxet[v] then DFS(v);
end
```

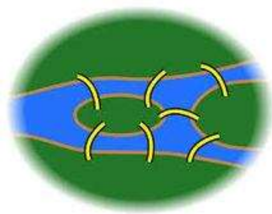
**Tìm kiếm  theo chiều rộng trên đồ thị**

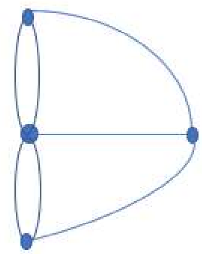
```
procedure BFS(v);
(* Tìm kiếm theo chiều sâu bắt đầu từ đỉnh v, các biến Chuaxet, Ke là biến toàn cục *)
begin
        QUEUE= Ø;
        QUEUE ⇐ v;
        Chuaxet[v]=false;
        while QUEUE ≠ 0 do
        begin
                p ⇐ QUEUE ;
                thămđỉnh(p);
`       for u ∈ kề(p) do
        if Chuaxet[u] then
        begin
                QUEUE ⇐ u;
                Chuaxet [u] = false;
        end
        end
end
```

**ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON**

Đồ thị Euler và các kết quả liên quan đã được nhà toán học Thụy Sĩ Leonhard Euler (1707-1783) nghiên cứu từ thế kỉ 18 khi ông tiến hành giải quyết bài toán nổi tiếng có tên gọi “Bài toán 7 chiếc cầu ở thành phố Kónigsberg”. Có thể đi dạo qua hết tất cả các cầu mỗi cầu chỉ qua một lần rồi trở về nơi xuất phát được không.

Bài toán xuất phát từ những bức thư của người dân vùng Kónigsberg (Nga) gửi cho Euler vào năm 1736 khi ông sang Nga làm việc. Euler đã chứng minh được rằng, một cuộc đi dạo như vậy không thể nào xảy ra. Euler đã biểu diễn bài toán trên bằng đồ thị và khẳng định, đồ thị như vậy không thể vẽ được một đường liền nét đi qua tất cả các cạnh sao cho mỗi cạnh chỉ vẽ một lần.





Hình  chiếc cầu ở thành phố Kónigsberg và biểu diễn thành đồ thị

**ĐỒ THỊ EULER VÀ ĐỒ THỊ HAMILTON**

Đồ thị Euler

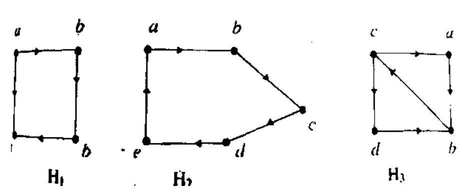
Định nghĩa 1. Chu trình đơn trong G đi qua mỗi cạnh của nó một lần được gọi là chu trình Euler. Đường đi đơn trong G đi qua mỗi cạnh của nó một lần được gọi là đường đi Euler. Đồ thị được gọi là đồ thị Euler nếu nó có chu trình Euler và gọi là đồ thị nữa Euler nếu nó có đường đi Euler.

Định lý 1(Euler). Đồ thị vô hướng liên thông G là đồ thị Euler khi và chỉ khi mọi đỉnh của $G$ đều có bậc chẵn.

Hệ quả . Đồ thị vô hướng liên thông $G$ *là nửa Euler khi* và chỉ khi nó không quá 2 đỉnh bậc lẻ.

Định lý 2. Đồ thị có hướng liên thông mạnh là đồ thị Euler khi và chỉ khi

$$deg^{+}(v) = deg^{-}(v)$$



```
procedure Euler_cycle;
begin
     STACK= Ø;
      CE=Ø;
      chọn u là một đỉnh nào đó của đồ thị;
      STACK ⇐ u;
      while STACK ≠ Ø do
      begin
            x= top (STACK);
            if Ke(x) ≠ Ø then
            begin
                  y=đỉnh đầu tiên trong danh sách Kề(x);
                  STACK ⇐ u;
                  (* Loại bỏ cạnh (x,y) khỏi đồ thị *)
                  Ke(x) = Ke(x) \ {y};
                  Ke(y)= Ke(y)\{x};
            end else
            begin
            x  ⇐ STACK;
            CE ⇐ u;
            end
      end
end
```

**Đồ thị Hamilton**

Định nghĩa 2

Đường đi qua tất cả các đỉnh của đồ thị mỗi đỉnh đúng một lần được gọi là đường đi Hamilton. Chu trình bắt đầu từ đỉnh v nào đó qua tất cả các đỉnh còn lại mỗi đỉnh đúng một lần rồi quay trở về v được gọi là chu trình Halmilton. Đồ thị G được gọi là đồ thị Halmilton nếu nó chứa chu trình Hamilton, và gọi là nửa Hamilton nếu nó chứa đường đi Hamilton.

Định lý 3(Dirak 1952)

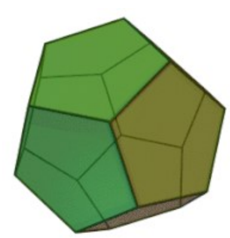
Đơn đồ thị vô hướng $G$ với n>2 đỉnh, mỗi đỉnh có bậc không nhỏ hơn n/2 là đồ thị Hamilton.

Định lý 4

Giả sử G là đồ thị có hướng liên thông mạnh với n đỉnh. Nếu

$$deg^{+}(v) \geq n/2, deg^{-}(v) \geq n/2, \forall v$$

Thì G là Hamilton



## Đồ thị Hamilton

Định lý 5 (tổng quát hơn Dirac)

Giả sử đồ thị G gồm $n \geq 3$ *đỉnh v*à mọi cặp đỉnh của nó đều có tổng bậc không bé hơn n. Khi đó G là đồ thị Halmilton.

Định lý 6(Ore, 1960) Giả sử G gồm $n \geq 3$ đỉnh và mọi cặp đỉnh không kề nhau của

*G đều có tổng bậc không bé hơn n. Khi đó G* là đồ thị Halmilton.

Định lý 7.

Giả sử G là đồ thị Halmilton. Khi đó

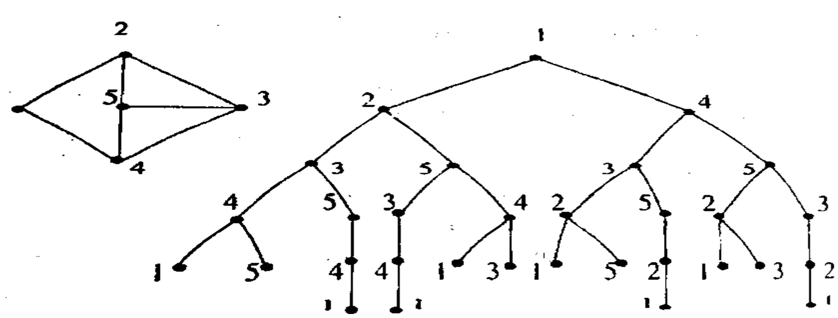
a. Mọi đỉnh của nó phải có bậc lớn hơn hoặc bằng 2.

b. Nếu một đỉnh có bậc bằng 2 thì hai cạnh của nó phải nằm trên một chu trình Halmilton

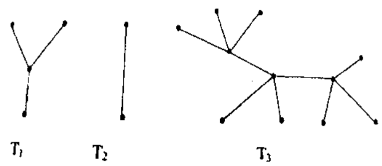
c. Nếu một đỉnh có bậc lớn hơn 2 và có hai cạnh của nó nằm trên một chu trình Hamilton thì các cạnh còn lại của nó không nằm trên chu trình Hamilton

**Thuật toán liệt kê 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 ∈ 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 ∈ V do chuaxet[v]=true;
        X[1]= v0;
        chuaxet[v0] = false;
        Hamilton(2);
end
```



## CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ




Định nghĩa 3

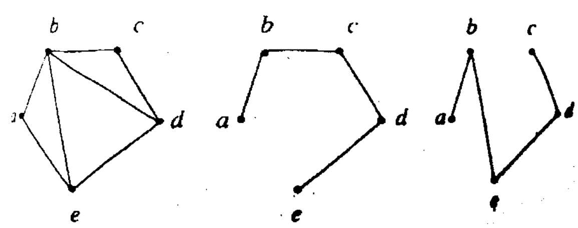
Ta gọi cây là đồ thị vô hướng liên thông không có chu trình. Đồ thị không có chu trình được gọi là rừng.

Định lý 8. Giả sử $G = (V, E)$ là đồ thị vô hướng $n$ đỉnh. Khi đó các mệnh đề sau đây là tương đương:

1. $T$ là cây
2. $T$ không chứa chu trình  và có $n - 1$ $cạnh$
3. $T$ liên thông và có $n - 1$
4. $T$ liên thông và mỗi cạnh của nó là cầu
5. Hai đỉnh bất kì của $T$ được nối với nhau bởi đúng một đường đi đơn
6. $T$ không chứa chu trình nhưng hễ cứ thêm vào nó một cạnh ta thu được đúng một chu trình

## CÂY VÀ CÂY KHUNG CỦA ĐỒ THỊ

Định nghĩa 4. Giả sử $G = (V, E)$ là đồ thị vô hướng liên thông. Cây T=(V, F) với $F$ ⊂E được gọi là cây khung của đồ thị $G$.




Định nghĩa 5 (Cayley). Số cây khung của đồ thị $K_n$ $là$ $n^{n-2}$

**Thuật toán xây dựng cây khung của đồ thị**

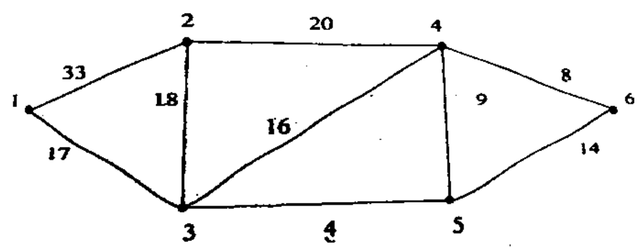
```
procedure cay_khung_DFS(v);
// Đồ thị G cho bởi danh sách kề
begin
        chuaxet[v]= false;
        for u ∈ ke(v) do
        if chuaxet[u] then
        begin
                T = T ∪ (v,u);
                Cay_khung_DFS(u);
        end
end
begin
        for u ∈ V do chuaxet[u] = true;
        T = ∅;
        Cay_khung_DFS(v);
        T là một cây khung;
end
```

**Thuật toán xây dựng cây khung của đồ thị**

```
procedure cay_khung_BFS(v);
// Đồ thị G cho bởi danh sách kề
begin
        Queue=∅;
        Queue ⇐ v;
        chuaxet[v]=false;
        while Queue ≠ ∅ do
        begin
        v ⇐ Queue;
        for u ∈ kề (v) do
        if chuaxet[u] then
        begin
                Queue ⇐ v;
                chuaxet[u]=false;
                T = T ∪ (v,u);
        end
end
end
begin
        for u ∈ V do chuaxet[u] = true;
        T = ∅;
        Cay_khung_BFS(v);
        T là một cây khung;
end
```

## BÀI TOÁN CÂY KHUNG NHỎ NHẤT

procedure Kruskal (1956);
begin
    $T = \emptyset$;
    while $|T| < n - 1 \text{ } and \text{ } E \neq \emptyset$ do
    begin
        Chọn $e$ là $cạnh \text{ } có \text{ } độ \text{ } dài \text{ } nhỏ \text{ } nhất \text{ } trong \text{ } E$;
        $E = E \setminus$ {e};
        if $T \cup \{e\}$ không chứa chu trình then $T = T \cup \{e\}$;
    end
    if $|T| < n - 1 \text{ } then$ Đồ $thị \text{ } không \text{ } liên \text{ } thông$;
end




## BÀI TOÁN CÂY KHUNG NHỎ NHẤT

**Thuật toán Prim (1957)**

Thuật toán Kruskal làm việc kém hiệu quả đối với những đồ thị dày ( đồ thị với số cạnh $m \approx \frac{n(n-1)}{2}$) .

Trong trường hợp đó thuật toán Prim tỏ ra hiệu quả hơn. Thuật toán Prim còn được gọi là phương pháp lân cận. Trong phương pháp này, bắt đầu từ một đỉnh tùy ý của đồ thị $s$.

- $Đầu \text{ } tiên \text{ } ta \text{ } nối \text{ } s \text{ } với \text{ } đỉnh \text{ } lân \text{ } cận \text{ } gần \text{ } nó \text{ } nhất$, chẳng hạn là đỉnh $y$
- Tìm cạnh có độ dài nhỏ nhất trong số các cạnh kề với hai đỉnh $s, y$
- Quá trình này tiếp tục cho đến khi được cây $n \text{ } đỉnh$ và $n - 1 \text{ } cạnh$

$$C = \begin{vmatrix} 0 & 33 & 17 & \infty & \infty & \infty \\ 33 & 0 & 17 & \infty & \infty & \infty \\ 17 & 18 & 0 & 16 & 4 & \infty \\ \infty & 20 & 16 & 0 & 9 & 8 \\ \infty & \infty & 4 & 9 & 0 & 14 \\ \infty & \infty & \infty & 8 & 14 & 0 \end{vmatrix}$$

procedure Prim;
$T = \{e\text{: } cạnh \text{ } có \text{ } trọng \text{ } số \text{ } nhỏ \text{ } nhất \text{ } trong \text{ } E\}$;
$E = E - \{e\}$;
for $i$=1 to $n$-2 do
begin
    Chọn $e$ là cạnh có trọng số nhỏ nhất trong $E$ và liên thuộc với $T$;
    if $T \cup \{e\} không \text{ } chứa \text{ } chu \text{ } trình \text{ } then \quad = T \cup \{e\}$;
    $E = E \setminus \{e\}$;
end

## BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT TRÊN ĐỒ THỊ

Bài toán 1.

Cho một đồ thị đơn vô hướng liên thông. Tìm đường đi ngắn nhất theo số cạnh từ đỉnh $a$ *đến đỉnh* $b$ *của* đồ *thị*.

Đồ thị của bài toán này không có trọng số hay trọng số của các cạnh là bằng nhau, và đường đi ngắn nhất giữa hai đỉnh trên đồ thị được hiểu là số cạnh của đường đi này là bé nhất.

Có thể giải bài toán này bằng thuật toán BFS.

Bài toán 2. (Thuật toán tìm đường đi ngắn nhất trong đồ thị có trọng số cạnh không âm – Thuật toán Dijkstra)

Cho đồ thị đơn vô hướng liên thông $G$ có trọng số dương. Tìm đường từ đỉnh $s$ đến $t$ của $G$ sao cho tổng trọng số của đường đi này là nhỏ nhất.

- Đặc điểm của phương pháp này là cực tiểu hóa nhãn cho các đỉnh

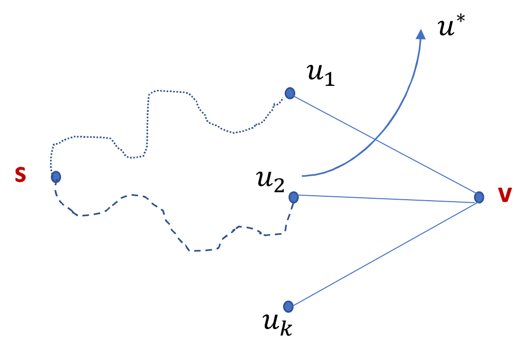
Ký hiệu nhãn của đỉnh $v$ *là* $d[v]$ *là độ dài ngắn nhất (tổng các trọng số)của đường đi từ* $s$ *đến* $v$.

Ký hiệu $a[u,v]$ là trọng số của cạnh $(u,v)$. Giả sử $n$ *là số đỉnh của* đồ *thị*.

## BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT TRÊN ĐỒ THỊ

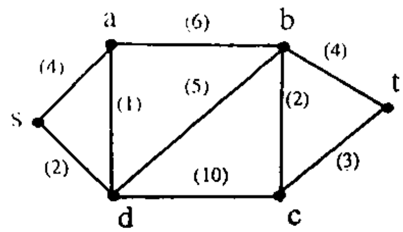
- *Input* Đồ *thị* $G = (V, E)$*có trọng số,* $s$ *là đỉnh xuất phát*
- *Output*: *Khoảng cách ngắn nhất từ* $s$ *đến các đỉnh còn lại*

```
procedure Dijkstra;
   for v ∈ V do
   begin
        d[v] = a[s, v];
        truoc[v]=s;
    end
    d[s] = 0; H= V\{s};
    while H ≠ ∅ do
    begin
            d[u]=min{d[z]:z∈ H};
            H=H\{u};
            for v ∈ V do
            if d[v] > d[u] + a[u,v] then
            begin
                    d[v]=d[u] + a[u,v];
                    truoc[v]=u;
            end
    end
end
```




d[$v$]=$min$ {d[$v$],d[$u_k$]+a[$u_k$,$v$]}

## BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT TRÊN ĐỒ THỊ

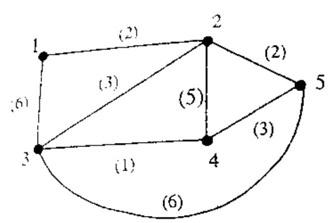



| Bước lặp | a | b | c | d | e | f | Đỉnh |
|---|---|---|---|---|---|---|---|
| *Khởi tạo* | [a, 0] | **[a, 1]*** | [a, ∞] | [a, ∞] | [a, ∞] | [a, ∞] | a |
| $1^0$ | | - | [b, 6] | **[b, 3]*** | [a, ∞] | [b, 8] | b |
| $2^0$ | | | **[d, 4]*** | - | [d, 7] | [b, 8] | d |
| $3^0$ | | | - | | [d, 7] | **[c, 5]*** | c |
| $4^0$ | | | | | **[f, 6]*** | - | f |
| $5^0$ | | | | | - | | e |

## BÀI TOÁN TÌM ĐƯỜNG ĐI NGẮN NHẤT TRÊN ĐỒ THỊ

### Thuật toán FLOYD

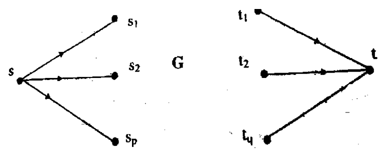
```
procedure Floyd;
begin
        for i=1 to n do
            for j=1 to n do
             begin
                   d[i,j]=a[i,j];
                   p[i, j]=i;
             end
        for k=1 to n do
            for i=1 to n do
               for j= 1 to n do
                   if   d[i, j] > d[i ,k] + d[k, j] then
              begin
                    d[i, j] = d[i, k] + d[k, j];
                    p[i, j] = p[k, j];
              end
end
```




$$A = \begin{bmatrix} 0 & 2 & 6 & \infty & \infty \\ 2 & 0 & 3 & 5 & 2 \\ 6 & 3 & 0 & 1 & 6 \\ \infty & 5 & 1 & 0 & 3 \\ \infty & 2 & 6 & 3 & 0 \end{bmatrix}$$

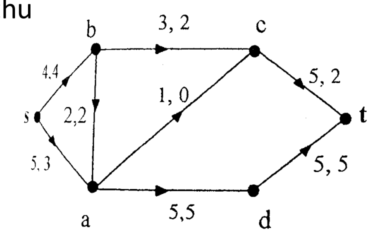
$$D = \begin{bmatrix} 0 & 2 & 5 & 6 & 4 \\ 2 & 0 & 3 & 4 & 2 \\ 5 & 3 & 0 & 1 & 4 \\ 6 & 4 & 1 & 0 & 3 \\ 4 & 2 & 4 & 3 & 0 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & 1 & 2 & 3 & 2 \\ 2 & 2 & 2 & 3 & 2 \\ 2 & 3 & 3 & 3 & 4 \\ 2 & 3 & 4 & 4 & 4 \\ 2 & 5 & 4 & 5 & 5 \end{bmatrix}$$

## BÀI TOÁN LUỒNG CỰC ĐẠI

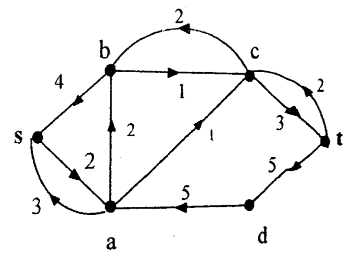



$$f(u,v) = \begin{cases} f(u,v) + \delta \ \text{nếu} \ (u,v) \in P \text{ là cung thuận} \\ f(u,v) - \delta \ \text{nếu} \ (u,v) \in P \text{ là cung nghịch} \\ f(u,v) \qquad\qquad \text{nếu} \ (u,v) \notin P \end{cases}$$

Luồng cực đại, đỉnh phát và đỉnh thu




Đồ thị trạng thái của luồng




Đồ thị tăng luồng

Algorithm

*Repeat*

*1: Xây dựng đồ thị tăng luồng*

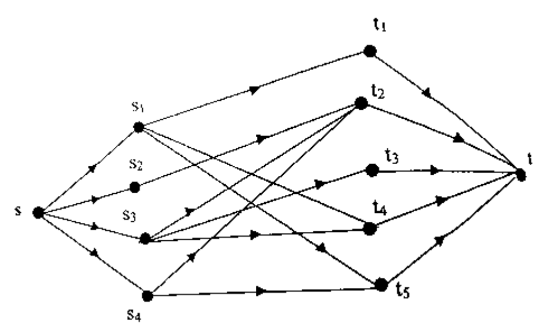
*2: Chọn đường tăng luồng với giá trị $\delta$*

*3: Tiến hành tăng luồng trên đường đã xét.*

Until *mạng đạt trạng thái cực đại của luồng;*

## BÀI TOÁN LUỒNG CỰC ĐẠI

| Chàng trai | Các cô gái mà chàng trai ưng ý |
|---|---|
| $s_1$ | $t_1, t_4, t_5$ |
| $s_2$ | $t_2$ |
| $s_3$ | $t_2, t_3, t_4$ |




Bài toán lập lịch cho hội nghị

Một hội nghị có $m$ tiểu ban và $n$ phòng họp, mỗi tiểu ban cần sinh hoạt trong một ngày tại phòng họp phù hợp với tiểu ban đó. Cho biết $A = a_{ij}$ $là\ ma\ trận,$ trong đó $a_{ij}$ = 1 (hoặc 0) cho biết tiểu ban $i\ có\ phù\ hợp\ (\ hoặc\ không\ phù\ hợp) với\ phòng\ họp\ j\ (i = 1, \dots, m, j = 1, \dots, n)$. Hãy bố trí các phòng họp sao cho hội nghị kết thúc sớm nhất.

**ĐẠI SỐ BOOLE VÀ ỨNG DỤNG**

Định nghĩa đại số Boole

Định nghĩa 1. Cho tập $B\ gồm\ hai\ phần\ tử\ 0\ và\ 1. Trên\ B\ xác\ định\ hai\ phép\ toán\ hai\ ngôi\ \ '+'$ cộng '.' nhân và một phép toán một ngôi '‾' (bù) thỏa mãn các tính chất sau:

1. Tính kết hợp: ∀ $a$, b, c ∈ $\{0,1\}$

$(a+b)+c=a+(b+c), (a.b).c=a.(b.c)$

2. Tính giao hoán: ∀ $a$, b ∈ $\{0,1\}$

$a+b=b+a$ a.b=b.a

3. Tính phân phối: $\forall a, b, c\ \in \{0,1\}$

$(a+b).c=a.c+b.c, \quad a.(b+c)=a.b+a.c$

4. Phần tử trung hòa $\forall a\ \in \{0,1\}$

$a+0=a, a.1=a$ (0 là $phần\ tử\ trung\ hòa\ với\ phép+, \quad$ 1 là $phần\ tử\ trung\ hòa$ đối $với\ phép\ .)$

5. Phần tử bù: ∀ $a\in\{0,1\}, tồn\ tại\ \bar{a}\ \ \in$ {0,1} sao cho

$a+\ \bar{a}=1,\ a.\bar{a}$=0

Bộ sáu(B, +,., '‾',0, 1) thỏa 5 tính chất trên gọi là một đại số Boole. Thay vì viết a.b, người ta viết ab.

**ĐẠI SỐ BOOLE VÀ ỨNG DỤNG**

a. Xét tập B gồm các mệnh đề toán học có giá trị false, true với các phép toán logic or, and, not. Khi đó bộ sáu ({true, false}, or, and, not, false, true) là một đại số boole.
b. Giả sử $E\ là\ một\ tập\ hợp. Ký\ hiệu$ '‾' là phép lấy bù của một tập hợp con của $E$. Ta có $(P(E)$,∪,∩,'‾' ',$\emptyset, E)\ là\ một\ đại\ số\ boole, trong$ đó $P(E)$là $tập\ hợp\ bao\ gồm\ các\ tập\ con\ của\ E.$
c. Bộ sáu {B={0,1}, +, ., '‾', 0, 1 }, trong đó phép + được hiểu theo phép + (mod 2) trong số học, phép '.' là phép nhân thông thường trong số học, phép bù '‾' thỏa mãn $\bar{1}=0, \bar{0}=1$. Là một đại số boole.

Các tính chất cơ bản

Từ định nghĩa trên có thể chứng minh các tính chất sau đây của một đại số Boole B={0,1}, $\forall a, b\ \in B=\{0,1\}. Ta\ có$

1. Luật đồng sức: $a+a=a,\ aa=a$

   $a+1=1, a0=0$

2. Luật nuốt: $a+ab=a,\ a(a+b)=a$

3. Luật bù kép: $\bar{\bar{a}}=a$

4. Luật De Morgan: $\overline{(a+b)}$=$\bar{a}\,.\,\bar{b}$ $\quad \overline{ab}$=$\bar{a}.\bar{b}$

Biểu thức Boole và hàm Boole

Với tập hợp $B = \{0, 1\}$, *một biến* $x$ *chỉ nhận các giá trị trong* $B$ *gọi* là *biến boole hay biến logic*.

*Một biểu thức boole trên* $B$ *được xây dựng từ việc* hợp thành bởi các biến boole với các phép toán cộng, nhân và bù, các dấu ngoặc (,).

Một cách đệ quy:

Nếu $P$ *và* $Q$ *là hai biểu thức boole*, thì $\bar{P}, \bar{Q}, P + Q, P.Q, \ldots$ *cũng* là biểu thức boole.

Định nghĩa 2

Ký hiệu $B^n$ là tập các phần tử $x = (x_1, x_2, \ldots, x_n)$ *trong* đó $x_i \in \mathrm{B}$, $i = \overline{1, n}$ . Một hàm boole n biến là một ánh xạ $f\colon B^n \to B$

Hàm boole $f(x_1, x_2, \ldots., x_n)$ nhận một trong hai giá trị 0 hoặc 1 ứng với mỗi bộ biến $(x_1, x_2, \ldots, x_n) \in B^n$. Vì $|B| = 2$ *và* $|B^n| = 2^n$.

Nên số hàm boole của n biến là $2^{2^n}$ . Với n=3, số hàm Boole 3 biến khác nhau là 256.

Ví dụ hàm boole ba biến $f(x, y, z) = xy + x\bar{y}z$

| x | y | z | f |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Một môn thi trắc nghiệm gồm 4 câu hỏi với số điểm lần lượt 2, 3, 5, 4. Nếu trả lời đúng mỗi câu sinh viên sẽ được điểm tối đa, trả lời sai chỉ được không điểm. Sinh viên thi đạt kết quả từ 10 điểm trở lên. Xác định hàm boole cho biết sinh viên thi đạt (=1) hay không đạt (=0).

| x | y | z | t | f |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Từ bảng chân trị này ta xác định biểu thức của $f$ *ở dạng tổng chuẩn tắc*.

$$f(x, y, z, t) = \bar{x}yzt + x\bar{y}zt + x\bar{y}zt + xyz\bar{t} + xyzt$$

**CÁC DẠNG KHÁC NHAU CỦA MỘT HÀM BOOLE**

1. Tích cơ bản

Để tiện cho việc trình bày, ta đưa vào quy tắc sau đây: giả sử biến $x \in B = \{0, 1\}, \sigma \in \{0, 1\}, kí\ hiệu$

$$x^{\sigma} = \begin{cases} x \quad khi\ \ \sigma = 1 \\ \bar{x} khi\ \ \sigma = 0 \end{cases}$$

Giả sử cho n biến $x_1, x_2, \dots, x_n$ $.Tích\ cơ\ bản\ của\ k\ biến$ là $tích\ có\ dạng\ x_{i_1}^{\sigma_1} x_{i_2}^{\sigma_2} \dots x_{i_k}^{\sigma_k}$ với $\sigma_1\sigma_2 \dots . \sigma_k \in \{0, 1\}$

$k \le n, 1 \le i_1, i_2, \dots, i_k \le$ n và $i_t \ne i_s$ nếu $t \ne s$.

Định lý 1

Mọi tích có thể đưa về 0 hoặc về một tích căn bản

Xét trên $B^4$ gồm các biến x, y, z, t. Ta có tích cơ bản của 3 biến: $xyt, xy\bar{t}$ , yzt và các tích cơ bản của 3 biến: $xy\bar{z}$y, $xyz\bar{z}$

Cho $P_1$ $và$ $P_2$ là $hai\ tích\ cơ\ bản, ta\ nói\ P_1\ chứa\ trong\ P_2$ nếu mọi biến và bù của biến $P_1$ đều có mặt trong $P_2$.

Từ định nghĩa trên dễ dàng chứng minh được rằng, nếu $P_1$ chứa trong $P_2$ thì

$P_1 + P_2 = P_1$ (luật nuốt)

Chẳng hạn, với $P_1$ =x$\bar{z}$ là tích cơ bản của hai biến x, z và $P_2$ =x$\bar{y}\bar{z}$ là $tích\ cơ\ bản\ của\ 3\ biến\ x, y, z. R$õ $ràngP_1$ chứa trong $P_2$ và $P_1 + P_2 = x\bar{z} + x\bar{y}\bar{z}$=x$\bar{z} = P_1$

Dạng tổng chuẩn tắc của hàm Boolean

Một hàm Boolean

$f\ có\ dạng\ tổng\ chuẩn\ tắc, nếu\ f$ là $một\ tích\ cơ\ bản\ hoặc$ là $tổng\ của\ nhiều\ tích\ cơ\ bản\ trong$ đó $không\ có\ tích$ nào $chứa\ trong\ một$ tích nào khác.

$f = x\bar{z}$ +xyz+$\bar{x}y\bar{z}$ là $dạng\ tổng\ chuẩn\ tắc$

$g = x\bar{z} + xyz + x\bar{y}\bar{z}$

Mọi hàm Boole (khác 0) đều có thể đưa về dạng tổng chuẩn tắc:

1. Dùng luật De Morgan và bù kép đưa biểu thức về dạng trong đó phép bù chỉ áp dụng trên các biến.
2. Luật phân phối đưa biểu thức về dạng tổng của các tích.
3. Dùng luật giao hoán, luật đồng sức và luật bù để đưa mỗi tích trong biểu thức về 0 hoặc là tích cơ bản.
4. Dùng luật nuốt để đưa về dạng tổng chuẩn tắc.

Ví dụ. Biểu diễn tổng chuẩn tắc của hàm Boole $f(x, y, z) = \overline{\overline{xy}z}\ \overline{(\bar{x} + z)(\bar{y} + \bar{z})}$

Giải

$f(x, y, z) = \overline{\overline{xy}z}\overline{(\bar{x} + z)(\bar{y} + \bar{z})}$

=(xy + $\bar{z}$)($x\bar{z}$+yz) (Luật De Morgan)

= $xy\bar{z} + xyz + x\bar{z}$ +0 (luật phân phối + luật bù)

= $x\bar{z} + xyz$

DẠNG TỔNG CHUẨN TẮC CỦA HÀM BOOLE

Định nghĩa 3. Một hàm boole

$f(x_1, x_2, \dots, x_n)$có dạng tổng chuẩn tắc hoàn toàn nếu $f$ có dạng tổng chuẩn tắc và trong đó mỗi số hạng đều là tích cơ bản có đủ $n$ biến $x_1, x_2, \dots, x_n$.

Ví dụ

Hàm $f(x,y,z) = xyz + xy\bar{z}+xy\bar{z} + x\bar{y}\bar{z}+\bar{x}yz + \bar{x}y\bar{z}$ là dạng tổng chuẩn tắc hoàn toàn.

Hàm $g(x,y,z) = x\bar{z} + xyz$ là một tổng chuẩn tắc nhưng chưa là tổng chuẩn tắc hoàn toàn.

Định lý 2. Một hàm boole $f(x_1, x_2, x_3,\dots, x_n)$ khác 0 đều có một dạng tổng chuẩn tắc hoàn toàn và duy nhất

Thuật toán

1. Biến đổi $f$ về dạng tổng chuẩn tắc.
2. Nhân tích cơ bản nào vắng mặt biến $x_i$ với ($x_i$+$\overline{x_i}$)

3. Lặp lại bước 2 cho đến khi trong $f$ mọi tích cơ bản đều là tích cơ bản của đủ $n$ biến $x_1$, $x_2$,…, $x_n$.

Ví dụ Biểu diễn hàm boole sau đây ở dạng tổng chuẩn tắc hoàn toàn

$$f(x,y,z) = x\bar{z} + \bar{x}y+xyz$$

Ta có biểu thức đã cho ở dạng tổng chuẩn tắc, tiếp tục với bước 2 và 3

$$f(x,y,z) = x\bar{z} + \bar{x}y + xyz = x\bar{z}(y+\bar{y}) + \bar{x}y(z+\bar{z}) + xyz = xy\bar{z}+x\bar{y}\bar{z}+\bar{x}yz + \bar{x}y\bar{z}+xyz$$

Dạng thu gọn và dạng tối thiểu của hàm Boole

Consensus của hai tích cơ bản

Định nghĩa 4. Cho $P_1, P_2$ là các tích cơ bản ( của ít nhất hai biến) trong đó $P_1, P_2$ có chứa $x_i$ hoặc $\overline{x_i}$. Ta gọi consensus $Q$ của $P_1$ , $P_2$ là tích của $P_1$, $P_2$ sau khi đã xóa $x_i$ và $\bar{x}_i$ và xóa các biến theo luật đồng sức $a.a = a$

Ví dụ: với $P_1 = xyzt$ và $P_2$=$x\bar{y}$, ta có $Q = xzt$. Với $P_1 = xy$, $P_2 = \bar{x}\bar{y}$, ta có $Q = 0$.

Định lý 3. Nếu $Q$ là consensus của $P_1, P_2$. Thì $P_1$+ $P_2 + Q = P_1$+ $P_2$

Chứng minh. Do $P_1, P_2$ có consensus nên giả sử $P_1$=$x^{\sigma_1}A$, $P_2$=$x^{\sigma_2}B$ với $\sigma_1, \sigma_2 \in \{0,1\}$.

Xét hai trường hợp: Nếu $\sigma_1 \neq \sigma_2$ , thì $Q = A.B$, do đó

$P_1 + P_2 + Q$= $x^{\sigma_1}A + x^{\sigma_2}B + AB\, x^{\sigma_1}$+ $ABx^{\sigma_2}$= $x^{\sigma_1}$A+ $x^{\sigma_2}B$= $P_1$+ $P_2$

Nếu $\sigma_1 = \sigma_2$= $\sigma_1 = \sigma$, thì $P_1 = x^{\sigma}A, P_2 = x^{\sigma}B$, nên $Q = x^{\sigma}AB$ Do đó

$P_1$+$P_2 + Q = x^{\sigma}A + x^{\sigma}B + x^{\sigma}$AB= $x^{\sigma}A + x^{\sigma}B$= $P_1 + P_2$ (Đpcm).

Với $P_1$=xyzt, $P_2 = x\bar{y}$ và $Q = xzt$ ta có

$P_1 + P_1$+Q=$xyzt + x\bar{y} + xzt = xyzt + x\bar{y} + xzt(y+\bar{y}) = xyzt + x\bar{y} = P_1 + P_2$

Nguyên nhân nguyên tố của hàm boole.

Tích cơ bản $P$ gọi là một nguyên (implicant) của hàm $f$ nếu: $P + f = f$

$P_1 = x\bar{z}$ là nguyên nhân, $P_2 = x$ không là nguyên nhân, $P_3 = xy\bar{z}$ là nguyên nhân của $f = x\bar{y} + xy\bar{z} + \bar{x}y\bar{z}$

Thật vây, viết $P_1$, $P_2, f$ ở dạng tổng chuẩn tắc hoàn toàn:

$P_1 = x\bar{z} = x\,\bar{z}\,(y+\bar{y}) = xy\bar{z} + x\bar{y}\bar{z}$

$P_2 = x = x(y+\bar{y})(z+\bar{z}) = xyz + xy\bar{z} + x\bar{y}z + x\bar{y}\bar{z}$

$f = x\bar{y}z + x\bar{y}\bar{z} + xy\bar{z} + \bar{x}y\bar{z}$

Dễ thấy : $P_1 + f = f$, $P_2 + f \neq f, P_3 + f = f$

Định nghĩa 5

Giả sử $P$ là một nguyên nhân của $f$, nếu xóa đi một biến nào đó trong $P$, thì $P$ không còn là nguyên nhân của $f$.

Khi đó $P$ được gọi là nguyen nhân nguyên tố của f.

Dễ dàng kiểm tra $P_1 = x\bar{z}$ là nguyên nhân nguyen tố $P_3 = xy\bar{z}$ là nguyên nhân nhưng không nguyên tố của

$f = x\bar{y} + xy\bar{z} + \bar{x}y\bar{z}$

Dạng thu gọn và tối thiểu của hàm Boole

Định nghĩa 6. Ta gọi độ phức tạp của một dạng tổng chuẩn tắc là số lần các biến xuất hiện trong nó.

Chẳng hạn, $\bar{x}\bar{y} + xy + x\bar{y}$ có độ phức tạp là 6, $x + \bar{y}$ có độ phức tạp là 2.

Định nghĩa 7. Dạng tổng chuẩn tắc của hàm $f$ có độ phức tạp bé nhất được gọi là dạng tối thiểu của $f$.

Hàm $f(x, y, z, t) = xyzt + x\bar{y} + xzt$ có dạng tương đương

$f_1 = xzt + x\bar{y}$ và $f_2 = xyzt + x\bar{y}$ rõ ràng dạng $f_1$ tối thiểu hơn $f_2$ của $f$

Định nghĩa 8.

Hàm Boole $f$ được gọi là dạng tổng chuẩn tắc thu gọn, nếu các số hạng của nó là các nguyên nhân nguyên tố của $f$.

Hàm $f = xyz + xy\bar{z} + \bar{x}yz + x\bar{y}z + \bar{x}\bar{y}z$ có dạng tổng chuẩn tắc thu gọn là $f = xy + xz + yz + \bar{x}\bar{y}\bar{z}$

Định nghĩa 9.

Một dạng tổng chuẩn tắc của hàm $f$ gọi là dạng tối thiểu, nếu các số hạng của nó là các nguyen nhân nguyên tố cốt yếu của $f$.

Lập bảng cho dạng thu gọn của $f = xy + xz + yz + \bar{x}\bar{y}\bar{z}$. Từ bảng này, mọi nguyên nhân nguyên tố của $f$ đều là cốt yếu. Vì vậy hàm $f$ đã cho có dạng tổng chuẩn tắc thu gọn đồng thời cũng là dạng tối thiểu.

| | xyz | $\bar{x}$yz | x$\bar{y}$z | xy$\bar{z}$ | $\bar{x}\bar{y}\bar{z}$ |
|---|---|---|---|---|---|
| xy | + | | | + | |
| xz | + | | + | | |
| yz | + | + | | | |
| $\bar{x}\bar{y}\bar{z}$ | | | | | + |

## Tối thiểu hóa hàm Boole

1. Phương pháp biến đổi đại số

Đưa $f$ về dạng tổng chuẩn tắc hoàn toàn. Tìm dạng tổng chuẩn tắc thu gọn gồm các số hạng là các nguyên nhân nguyên tố .

Xuất phát từ dạng thu gọn của $f$, chọn các nguyên nhân nguyên tố cốt yếu, ta được dạng tối thiểu cần tìm.

Tối thiểu hóa hàm boole $f = xyz + \bar{x}\bar{z} + xy\bar{z} + \bar{x}\bar{y}z + \bar{x}y\bar{z}$

Giải

Bước 1: $f = xyz + \bar{x}\bar{z} + xy\bar{z} + \bar{x}\bar{y}z + \bar{x}y\bar{z}$ ( luật nuốt của số hạng thứ 2 và thứ 5)

$= xyz + \bar{x}\bar{z} + xy\bar{z} + \bar{x}\bar{y}z$ (nhóm số hạng thứ 1 và thứ 3)

$= \bar{x}\,\bar{z} + \bar{x}\,\bar{y}z + xy$ (Cộng them Q=$\bar{x}\bar{y}$ của số hạng thứ nhất và thứ 2 )

$= \bar{x}\bar{z} + \bar{x}\bar{y}z + xy + \bar{x}\bar{y}$ (luật nuốt của số hạng thứ 2 và thứ 4)

$= \bar{x}\bar{z} + xy + \bar{x}\bar{y}$

Đến đây ta thu được dạng tối thiểu của $f$ đã cho là $\bar{x}\bar{z} + xy + \bar{x}\bar{y}$

Tuy nhiên nếu cộng thêm $Q = y\bar{z}$ của số hạng thứ 1 và thứ 2 vào dạng tối thiểu thu được của $f$:

$$f = \bar{x}\bar{z} + xy + \bar{x}\bar{y} + y\bar{z}$$

Bước 2: Trong hai biểu thức ở trên của $f$, có thể xóa số hạng $\bar{x}\bar{z}$, bởi vì $\bar{x}, \bar{z}$ có mặt trong $\bar{x}\bar{y}$ và $y\bar{z}$ (rõ hơn $\bar{x}\bar{z} = \bar{x}\bar{z}(y + \bar{y}) = \bar{x}y\bar{z} + \bar{x}\bar{y}\bar{z}$ sẽ bị $\bar{x}\bar{y}$ và $y\bar{z}$ nuốt). Nghĩa là ta thu được một dạng tối thiểu khác của $f$ là

$xy + \bar{x}\bar{y} + y\bar{z}$

## Phương pháp Karnaugh

Trường hợp hai biến $f(x, y)$: $B^2 \to B$

Mỗi tích cơ bản $xy, x\bar{y}, \bar{x}y, \bar{x}\bar{y}$ ứng với một ô vuông trong hình vuông lớn như

| | y | $\bar{y}$ |
|---|---|---|
| x | xy | x$\bar{y}$ |
| $\bar{x}$ | $\bar{x}$y | $\bar{x}\bar{y}$ |

$f_2 = \bar{x}+y$

Xác định tối thiểu của hàm sau đây

$f_1 = xy + x\bar{y}, f_2 = \bar{x}\bar{y} + xy + \bar{x}y, \quad f_3 = xy + \bar{x}\bar{y}$

| | y | $\bar{y}$ |
|---|---|---|
| x | x | x |
| $\bar{x}$ | | |

$f_1$

| | y | $\bar{y}$ |
|---|---|---|
| x | x | |
| $\bar{x}$ | x | x |

$f_2$

| | y | $\bar{y}$ |
|---|---|---|
| x | x | |
| $\bar{x}$ | | x |

$f_3$

$f_3 = xy + \bar{x}\bar{y}$

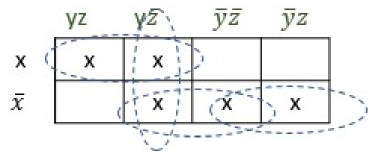
Trường hợp 3 biến $f(x, y, z)$: $B^3 \rightarrow \mathrm{B}$

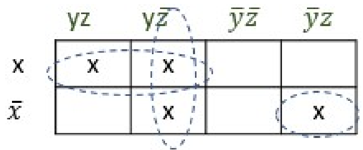
Tìm dạng tối thiểu của hàm

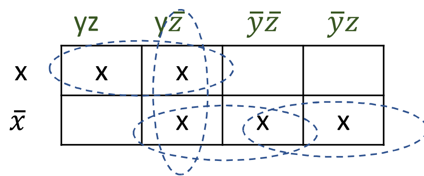
$f_1 = xyz + xy\bar{z} + \bar{x}y\bar{z} + \bar{x}\bar{y}z$

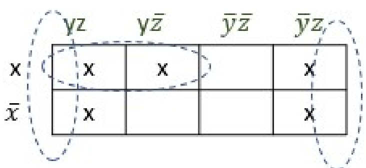
$f_2 = xyz + xy\bar{z} + x\bar{y}z + \bar{x}yz + \bar{x}\bar{y}z$

$f_3 = xyz + xy\bar{z} + \bar{x}y\bar{z} + \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z$




$f_1 = xy + y\bar{z} + \bar{x}\bar{y}z$

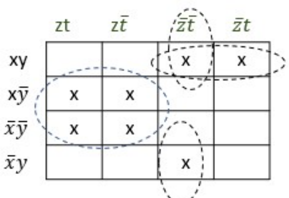
$f_2 = xy + z$

$f_3^1 = xy + y\bar{z} + \bar{x}\bar{y}$

$f_3^2 = xy + \bar{x}\bar{z} + \bar{x}\bar{y}$

Trường hợp bốn biến $f(x, y, z, t)$: $B^4 \rightarrow B$

$f = \bar{y}z + xy\bar{z} + y\bar{z}\bar{t}$




Phương pháp Quine-Mc.Cluskey

$f = \bar{x}\bar{y}z\bar{t} + \bar{x}\bar{y}zt + \bar{x}y \ \bar{t} + \bar{x}yzt + xy\bar{z}\bar{t}$+$xy\bar{z}$t+$xyz\bar{t} + xyzt$

- Bước 1: Đưa về dạng chuẩn tắc thu gọn $x_i \rightarrow 0, \bar{x_i} \rightarrow 1$
- Bước 2: Sắp xếp tổ hợp theo số lượng số 1 thành nhóm

| Bảng a | | |
|---|---|---|
| Số hạng | Nhị phân | Thập phân |
| $\bar{x}\bar{y}z\bar{t}$ | 0010 | 2 |
| $\bar{x}\bar{y}z$ | 0011 | 3 |
| $\bar{x}y \ \bar{t}$ | 0110 | 6 |
| $\bar{x}yzt$ | 0111 | 7 |
| $xy\bar{z}\bar{t}$ | 1100 | 12 |
| $xy\bar{z}t$ | 1101 | 13 |
| $xyz\bar{t}$ | 1110 | 14 |
| $xyzt$ | 1111 | 15 |

| Bảng b | | |
|---|---|---|
| 2 | 0010 | * |
| 3<br>6<br>12 | 0011<br>0110<br>1100 | *<br>*<br>* |
| 7<br>13<br>14 | 0111<br>1101<br>1110 | *<br>*<br>* |
| 15 | 1111 | * |

- Bước 3: So sánh mỗi tổ hợp nhóm $i\ với\ nhóm\ i+1\ và\ gộp\ nhóm\ nếu$ khác nhau một vị trí. Thay vị trí khác nhau bằng dấu '-'.
- Bước 4: Gạch bỏ tổ hợp trùng lặp chỉ để lại một và tiếp tục thực hiện bước 3 cho đến khi không thể thực hiện được nữa.

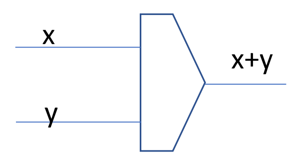
| Bảng c | | |
|---|---|---|
| 2,3 | 001- | * |
| 2,6 | 0-10 | * |
| 3,7 | 0_11 | * |
| 6,7 | 011- | * |
| 6, 14 | -110 | * |
| 12, 13 | 110- | * |
| 12, 14 | 11-0 | * |
| 7, 15 | -111 | * |
| 13, 15 | 11-1 | * |
| 14, 15 | 111- | * |

| Bảng d | | |
|---|---|---|
| 2,3, 6,7 | 0-1- | $\bar{x}z$ |
| 2, 6, 3, 7 | 0-1- | Loại |
| 6,7,14,15 | -11- | $yz$ |
| 6,14,7,15 | -11- | Loại |
| 12, 13, 14,15 | 11-- | Loại |
| 12, 14, 13, 15 | 11-- | xy |

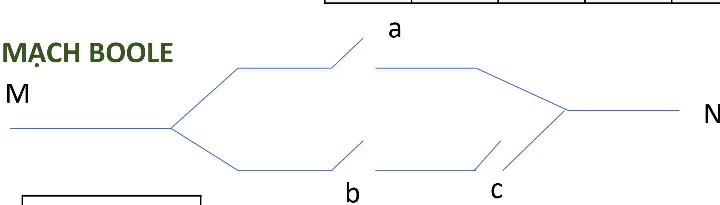
| Bảng e | | |
|---|---|---|
| 2,3, 6,7 | 0-1- | $\bar{x}z$ |
| 6,7,14,15 | -11- | $yz$ |
| 12,14,13,15 | 11-- | xy |

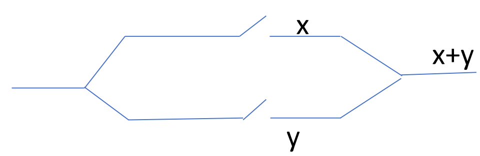
$$f=\bar{x}z+yz+xy$$

| $f$ | $\bar{x}\bar{y}z\bar{t}$ | $\bar{x}\bar{y}z$ | $\bar{x}yz\bar{t}$ | $\bar{x}yzt$ | $\bar{z}\bar{t}$ | $x\ \bar{z}t$ | $xy\ \bar{t}$ | $xyz$ |
|---|---|---|---|---|---|---|---|---|
| $\bar{x}z$ | x | x | x | X | | | | |
| $yz$ | | | x | x | | | x | x |
| $xy$ | | | | | x | x | x | X |




**MẠCH BOOLE**




| a | b | c | f |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

$f(a,b,c)=\bar{a}bc$+a$\bar{b}c$ $+ab\bar{c}$+abc
=a+bc