Chương 2. CÁC TẬP CON CỦA TẬP HỢP

Khi nghiên cứu một sự kiện ta thường phải xem xét tất cả các tình huống có thể xảy ra. Để xét hết mọi tình huống thì ta phải "vét cạn" các tình huống = tập con.

2.1. Các tập con của tập hợp và tập bội

Giả sử X là một tập hữu hạn nào đó.

Người ta thường ký hiệu tập các tập con của X là $P(X)$. Tập X có 2^n tập con. Do vậy, người ta còn ký hiệu tập các tập con của X là 2^X.

Trên tập $P(X)$ có quan hệ bao hàm thức $\subseteq$. Quan hệ này biến tập $P(X)$ thành một tập có thứ tự bộ phận.

Hai phép toán hợp và giao còn biến tập $P(X)$ thành một dàn với hai phép toán trên dàn:

$$\forall A, B \in P(X) \; : \; sup(A, B) = A \cup B \, , \, inf(A, B) = A \cap B$$

Cận trên (*sup*) của hai tập hợp là hợp của hai tập này; đó là tập con nhỏ nhất chứa hai tập trên. Cận dưới (*inf*) của hai tập hợp là giao của hai tập này, đó là tập con lớn nhất chứa trong hai tập trên.

Bài toán tập con

Cho tập X gồm n phần tử. Hãy tìm tất cả các tập con của tập X.

2.1.1. Thuật toán dãy nhị phân sinh các tập con

Ký hiệu $X = \{x_1, x_2, \ldots, x_n\}$. Mỗi tập con $A \subseteq X$ có thể biểu diễn bằng một dãy nhị phân $< b_1, b_2 \ldots b_n >$ độ dài n, với:

$$b_i = \begin{cases} 1 & \text{, nếu } x_i \in A \\ 0 & \text{, ngược lại} \end{cases}$$

Ngược lại, mỗi dãy nhị phân độ dài n xác định cho ta một tập con của tập X.

Vậy để tìm tất cả các tập con của tập X ta đi tìm tất cả các dãy nhị phân độ dài n.

Ví dụ 2.1:

Với $n = 3$ ta có 8 dãy nhị phân biểu diễn các tập con của tập 3 phần tử được sắp xếp trong bảng sau đây.

i	$B[1]$	$B[2]$	$B[3]$	v
1	0	0	<u>0</u>	0
2	0	<u>0</u>	1	1
3	0	1	<u>0</u>	2
4	<u>0</u>	1	1	3
5	1	0	<u>0</u>	4
6	1	<u>0</u>	1	5
7	1	1	<u>0</u>	6
8	1	1	1	7

Thuật toán 2.1

Đầu vào: Số nguyên dương n

Đầu ra: Dãy các dãy nhị phân độ dài n biểu diễn các tập con của tập n phần tử, được sắp tăng theo thứ tự từ điển

```
Begin
   input n ;
   B[1..n] ← 0 ;
   repeat
      print B[1..n] ;
      i ← n ;
      while B[i] = 1 do { B[i] ← 0 ; i ← i - 1 } ;
      if i ≥ 1 then B[i] ← 1 ;
   until i = 0 ;
End.
```

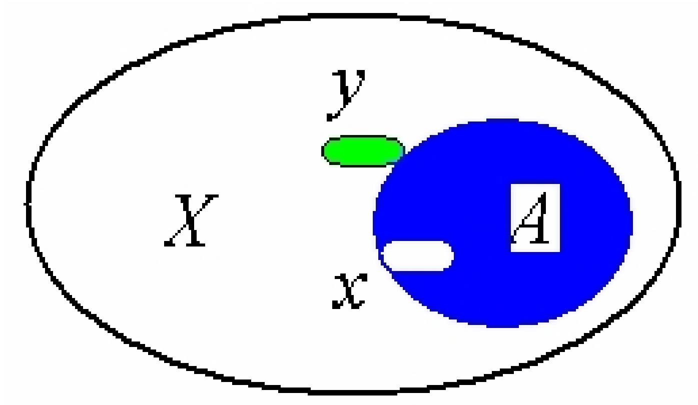
Độ phức tạp của Thuật toán 2.1:

Độ phức tạp của thuật toán là O($n.2^n$).

2.1.2. Thuật toán thêm/bớt sinh các tập con

Giả sử A là một tập con nào đó của X vừa tìm được.

- Nếu ta thêm phần tử y ngoài A vào A ta sẽ được tập con khác của X là $A' = A \cup \{y\}$. Hai dãy nhị phân biểu diễn hai tập con A và A' chỉ sai khác nhau một vị trí tương ứng với phần tử y: thay 0 bằng 1.
- Nếu ta bớt phần tử x trong A ta sẽ được tập con khác của X là $A'' = A \setminus \{x\}$. Khi đó, hai dãy nhị phân biểu diễn hai tập con A và A'' cũng chỉ sai khác nhau một vị trí tương ứng với phần tử x: thay 1 bằng 0.



Từ quan sát trên ta xây dựng thuật toán như sau.

a) *Thuật toán đệ quy*:

1) Các tập con của tập một phần tử được biểu diễn bởi hai dãy nhị phân độ dài một là < 0 > và < 1 >.

2) Để tìm các tập con của tập n phần tử, ta tìm các dãy nhị phân độ dài n-1 biểu diễn các tập con của tập n-1 phần tử là $c_1, c_2, \ldots, c_q$, với $q = 2^{n-1}$. Xây dựng dãy các dãy nhị phân độ dài n như sau: $c_1 0$, $c_2 0$,

…, c_q0, c_q1, $c_{q-1}1$, …, c_21, c_11. Đó chính là các dãy nhị phân biểu diễn các tập con cần tìm.

Ví dụ 2.2: Các dãy nhị phân biểu diễn các tập con của tập 3 phần tử được sinh ra bằng phương pháp thêm/bớt một phần tử như sau.

i	p	$B[1]$	$B[2]$	$B[3]$
0		0	0	0
1	1	**1**	0	0
2	2	1	**1**	0
3	1	**0**	1	0
4	3	0	1	**1**
5	1	**1**	1	1
6	2	1	**0**	1
7	1	**0**	0	1

b) *Thuật toán lặp*:

- Xuất phát từ dãy nhị phân đầu tiên gồm *n* chữ số 0, thuật toán vào vòng lặp để tìm các dãy nhị phân độ dài *n* theo nguyên lý kế thừa: dãy sau được tạo bởi từ dãy trước đó bằng cách thay đổi chỉ tại một vị trí, với 0 thay bằng 1 và ngược lại (thêm vào hoặc bớt đi).

- Vị trí thay thế: $p = 1 +$ số lần chia hết cho 2 của số thứ tự i

- Thuật toán kết thúc khi: $p > n$

Dãy các dãy nhị phân tìm được có tính chất đặc biệt: hai dãy nhị phân kế tiếp nhau chỉ khác nhau đúng một vị trí. Hay nói một cách khác, dãy sau đã kế thừa n-1 vị trí của dãy trước đó.

Do vậy dãy các dãy nhị phân như trên thường được dùng trong mật mã học với tên gọi là *mã nhị phân Gray bậc n*.

Thuật toán 2.2

Đầu vào: Số nguyên dương n

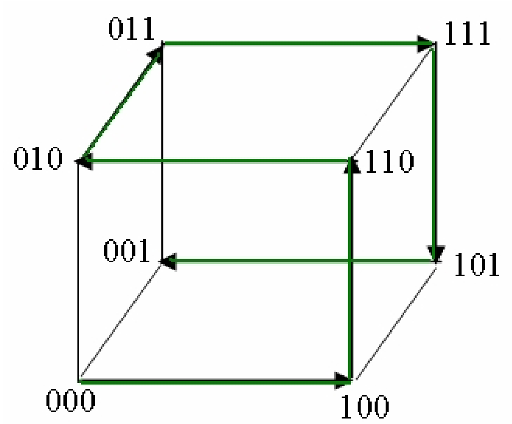
Đầu ra: Dãy các dãy nhị phân độ dài n biểu diễn các tập con của tập n

```
Begin
    input n ;
    B[1..n] ← 0 ;
    i ← 0 ;
    repeat
        print B[1..n] ;
        i ← i + 1 ;  p ← 1 ;  j ← i ;
        while j mod 2 = 0 do
            { j ← j div 2 ;
              p ← p + 1 } ;
        if p ≤ n then B[p] ← 1 - B[p] ;
```

```
    until p > n
End.
```

Ý nghĩa đồ thị của thuật toán

Xây dựng đồ thị mà các đỉnh là các dãy nhị phân độ dài n. Hai đỉnh được nối với nhau bằng một cạnh ⇔ hai dãy nhị phân tương ứng chỉ khác nhau đúng một vị trí. Đồ thị được gọi là khối nhị phân n chiều. Dãy các tập con được sinh tương ứng với một đường đi Hamilton.



2.1.3. Tập bội

1) Định nghĩa của tập bội (*multi-set*)

Định nghĩa 2.3: *Tập bội* là tập hợp, mà nó cho phép các phần tử xuất hiện nhiều hơn một lần.

Chẳng hạn, tập giá trị của các biến trong một chương trình, bộ đánh dấu trong một hệ mạng... là những ví dụ điển hình về tập bội.

Tập bội được viết dưới dạng tổng quát như sau:

$$X = (k_1 * x_1, k_2 * x_2, ..., k_n * x_n) ,$$

trong đó: $k_i \geq 0$ với $i = 1, 2, ..., n$.

Các phần tử $x_1, x_2, \ldots x_n$ được gọi là *cơ sở* của tập bội, còn $k_1, k_2, \ldots, k_n$ là *bội* của các phần tử tương ứng.

Lực lượng của tập bội này được tính như sau:

$$|X| = \sum_{i=1}^{n} k_i$$

2) Tập con bội của tập bội

Quan hệ bao hàm thức trên các tập bội được định nghĩa như sau. Giả sử $A = (t_1 * x_1, t_2 * x_2, \ldots, t_n * x_n)$ và $B = (q_1 * x_1, q_2 * x_2, \ldots, q_n * x_n)$ là hai tập bội nào đó.

Định nghĩa 2.4: Ta nói rằng tập bội A là tập con bội của tập bội B khi và chỉ khi bội của mỗi phần tử cơ sở trong A không vượt quá bội của phần tử này trong B.

$$A \subseteq B \quad \Leftrightarrow \quad \forall i = 1, 2, \ldots, n: \quad t_i \leq q_i \qquad (2.1)$$

Với tập bội $X = (k_1 * x_1, k_2 * x_2, ..., k_n * x_n)$ thì mỗi tập con bội $A = (t_1 * x_1, t_2 * x_2, ..., t_n * x_n)$ của tập bội X phải thỏa mãn:

$$0 \leq t_i \leq k_i\ , \forall i = 1, 2, \ldots, n. \qquad (2.2)$$

3) Bài toán tập con của tập bội

Cho tập bội $X = (k_1 * x_1, k_2 * x_2, ..., k_n * x_n)$. Hãy tìm tất cả các tập con bội của tập này.

Ta thường biểu diễn cho tập bội A bằng một dãy số nguyên $< t_1\, t_2 \ldots t_n >$ độ dài n các bội của phần tử cơ sở trong tập này.

Chẳng hạn, tập bội $A = (3 * x_1, 4 * x_2, 1 * x_3, 5 * x_4)$ có thể biểu diễn bởi dãy số $< 3\ 4\ 1\ 5 >$.

Bài toán tập con của tập bội có thể đưa về bài toán dãy bị chặn với các dãy biên là $s = < 0\ 0 \ldots 0 >$ và $g = < k_1\ k_2 \ldots k_n >$

Số tập con bội của tập bội X là: $\prod_{i=1}^{n}(k_i + 1)$

Việc tìm tất cả các tập con bội A của tập bội X có thể đưa về việc tìm tất cả các dãy số $< t_1\ t_2 \ldots t_n >$ thỏa mãn (2.2) biểu diễn các tập con bội cần tìm.

4) Sinh các tập con bội

Thuật toán 2.3

Đầu vào: Các số nguyên dương $n, k_1, k_2, \ldots k_n$

Đầu ra: Dãy các tập con bội của tập bội $X = (k_1 * x_1, k_2 * x_2, \ldots, k_n * x_n)$, mà các từ biểu diễn $t_1\ t_2 \ldots t_n$ của chúng được sắp tăng theo thứ tự từ điển

```
Begin
   input n ;
   for i ← 1 to n do { K[i] ← k_i ;  T[i] ← 0 } ;
   repeat
      print T[1..n] ;
      i ← n ;
      while T[i] = K[i] do { T[i] ← 0 ; i ← i - 1 } ;
      if i ≥ 1 then T[i] ← T[i] + 1 ;
```

```
until i = 0 ;
End.
```

Độ phức tạp của thuật toán

Độ phức tạp tổng thể của thuật toán là $O\left(n.\prod_{i=1}^{n}(k_i+1)\right)$.

Độ phức tạp của thuật toán xấp xỉ bằng $O(n.l^n)$,

với $l = \max\ \{k_1, k_2, ..., k_n\}$.

Ví dụ ứng dụng - *Bài toán xếp ba lô*:

Giả sử ta có n loại đồ vật quý và một ba lô có dung tích c.
Loại đồ vật thứ i có số lượng là k_i và mỗi đồ vật loại này có giá trị là p_i và dung lượng là s_i ($i = 1,2, \ldots, n$).

Hãy chọn và xếp các đồ vật vào ba lô sao cho tổng giá trị các đồ vật xếp trong ba lô là lớn nhất có thể.

Ký hiệu đồ vật loại thứ nhất là x_1, loại thứ hai là x_2, … và loại thứ n là x_n. Tập các đồ vật trên có thể biểu diễn bằng tập bội:

$$X = (k_1 * x_1, k_2 * x_2, \ldots k_n * x_n)$$

Mỗi phương án chọn các đồ vật để xếp vào ba lô chính là một tập con bội A của tập bội trên, sao cho:

$$A = (t_1 * x_1, t_2 * x_2, ..., t_n * x_n) \ , \ A \subseteq X \ \text{ và } \ \sum_{i=1}^{n} t_i.s_i \le c$$

Phương án tối ưu là phương án mà $\sum_{i=1}^{n} t_i.p_i$ đạt giá trị lớn nhất.

Kết hợp Thuật toán 2.3 sinh các tập con bội và thuật toán tìm số lớn nhất ta sẽ nhận được nghiệm tối ưu của bài toán một cách nhanh chóng.

Xét bài toán trên với dữ liệu được cho trong bảng dưới đây.

Loại đồ vật	*a*	*b*	*c*	*d*	*e*
Số lượng	3	5	2	4	6
Giá trị	25	28	20	12	15
Dung lượng	10	11	8	5	6

và ba lô có dung lượng là 100

Tính toán theo các thuật toán trên, ta nhận được phương án xếp ba lô tối ưu là: $A = (0 * a, 5 * b, 2 * c, 1 * d, 4 * e)$ với tổng dung lượng là 100 (đầy ba lô) và tổng giá trị lớn nhất là 252

Bài toán xếp ba lô được ứng dụng nhiều trong các hệ thống truyền tin và mạng máy tính.

2.2. Các tập con *k*-phần tử

Cho tập X gồm n phần tử và k là một số nguyên không âm.

Hỏi có bao nhiêu tập con k-phần tử của tập X?

Ký hiệu: $\binom{n}{k}$ là số các tập con k-phần tử của tập n phần tử.

Hiển nhiên: $\binom{n}{k} = 0$ nếu $k > n$.

Định lý 2.5: Số các tập con k-phần tử của tập hợp n phần tử là:

$$\binom{n}{k} = \frac{[n]_k}{k!} = \frac{n!}{k!.(n-k)!}$$

Chứng minh:

- Theo bài toán chọn có thứ tự ta có: $[n]_k$ là số các cách chọn có thứ tự k phần tử trong tập n phần tử.

- Mỗi cách chọn này cho ta một tập con k-phần tử của tập n phần tử.

- Bằng cách hoán vị các phần tử thì mỗi tập con k-phần tử của tập n phần tử lại cho ta $k!$ cách chọn có thứ tự k phần tử trong tập n phần tử.

Từ đó suy ra điều phải chứng minh.

Từ Định lý 2.5 đại lượng $\binom{n}{k}$ còn được gọi là *tổ hợp chập k của n*.

Trước đây người ta thường ký hiệu đại lượng này là C_k^n.

Ý nghĩa khác của đại lượng tổ hợp

Luỹ thừa bậc n của nhị thức $(x + y)$ là một đa thức bậc n của x và y. Nghĩa là:

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

Đại lượng a_k được gọi là hệ số nhị thức. Ta viết lại luỹ thừa trên như sau:

$$(x+y)^n = \underbrace{(x+y)(x+y)...(x+y)}_{n}$$

Để có $x^k y^{n-k}$ ta phải chọn k thừa số trong số n thừa số ở trên, trong các thừa số đó ta chọn x để nhân và trong các thừa số còn lại thì chọn y để nhân.

Hệ số a_k chính là số các cách chọn này. Vậy thì: $a_k = \binom{n}{k}$.

Từ đó ta có hằng đẳng thức: $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$

Thay $y = 1$ ta có hằng đẳng thức sau đây:

$$(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} x^k$$

Hằng đẳng thức này quen thuộc với tên gọi là *nhị thức Newton* và đại lượng $\binom{n}{k}$ còn được gọi là *hệ số nhị thức*.

Định lý 2.6

1) $\displaystyle\sum_{k=0}^{n}\binom{n}{k} = 2^{n}$

2) $\displaystyle\sum_{k=0}^{n}\binom{n}{k}k = n.2^{n-1}$

Chứng minh:
1) Các đại lượng ở hai vế - số tất cả các tập con của một tập n phần tử. Nên chúng bằng nhau.

2) Xây dựng danh sách L liệt kê tất cả các phần tử trong tất cả các tập con của một tập n phần tử X.

Với $X = \{a, b, c\}$ thì $L = < a, b, c, a, b, a, c, b, c, a, b, c >$.

Hiển nhiên, đại lượng ở vế trái là độ dài của danh sách L. Ta cần chỉ ra rằng đại lượng ở vế phải cũng là độ dài của danh sách L.

Số lần xuất hiện của một phần tử $x \in X$ trong danh sách L bằng số các tập con của tập X chứa x. Số các tập con này lại bằng chính số các tập con của tập $X \setminus \{x\}$ có n-1 phần tử. Một phần tử tuỳ ý xuất hiện trong danh sách L là 2^{n-1} lần. Vậy độ dài danh sách L chính là đại lượng ở vế phải.

Định lý 2.7

1) $$\binom{n}{k} = \binom{n}{n-k}$$

2) $$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \qquad (2.3)$$

Chứng minh:

1) Mỗi tập con B gồm k phần tử của tập X tương ứng 1 - 1 với tập con $X \setminus B$ gồm n-k phần tử - tính đối xứng của tổ hợp

2) Lấy một phần tử nào đó $x \in X$ rồi phân tập các tập con k phần tử của X thành hai lớp rời nhau:

- Lớp thứ nhất bao gồm các tập con không chứa x và

- Lớp thứ hai bao gồm các tập con có chứa x.

Khi đó, lực lượng của lớp thứ nhất là $\binom{n-1}{k}$ và lực lượng của lớp thứ hai là số các tập con k-1 phần tử của tập $X \setminus \{x\}$ và bằng $\binom{n-1}{k-1}$.

Đẳng thức (2.3) liên quan chặt chẽ với tam giác sau đây do B. Pascal đưa ra mà người ta quen gọi là *tam giác Pascal*.

Tam giác Pascal gồm nhiều hàng đánh số từ 0, 1, 2, 3, ... Trên hàng thứ n là các hệ số nhị thức:

$$\binom{n}{0}\binom{n}{1}\binom{n}{2}\cdots\binom{n}{n-1}\binom{n}{n}$$

$$\begin{array}{lccccccccccc}
0 & & & & & & 1 & & & & & \\
1 & & & & & 1 & & 1 & & & & \\
2 & & & & 1 & & 2 & & 1 & & & \\
3 & & & 1 & & 3 & & 3 & & 1 & & \\
4 & & 1 & & 4 & & 6 & & 4 & & 1 & \\
5 & 1 & & 5 & & 10 & & 10 & & 5 & & 1 \\
\ldots & & & & & & \ldots\ldots\ldots & & & & &
\end{array}$$

Tam giác Pascal vô hạn về phía dưới, có hai cạnh bên là các số 1. Các số ở giữa là tổng hai số tương ứng thẳng ở hàng trên.

Định lý 2.8 (*Đẳng thức Cauchy*)

$$\binom{m+n}{k} = \sum_{s=0}^{k} \binom{m}{s}\binom{m}{k-s}$$

Chứng minh:

Giả sử có m nam và n nữ. Ta muốn chọn k người trong số những người này. Hiển nhiên đại lượng ở vế trái là số các cách chọn.

Phân tập các cách chọn trên thành các lớp rời nhau theo số nam s trong mỗi cách chọn. Ta có k+1 lớp ứng với s = 0,1,2,..., k

Để có một cách chọn đầy đủ thì ta phải chọn s nam trong số m nam trước, sau đó chọn tiếp k-s nữ trong số n nữ.

Vậy lực lượng của lớp thứ s là: $\binom{m}{s}\binom{m}{k-s}$

Từ đó suy ra đẳng thức Cauchy.

Đẳng thức Cauchy giúp cho việc tính toán tổ hợp trên các tham số lớn.

Trường hợp đặc biệt $m = k = n$ ta có đẳng thức sau đây:

$$\binom{2n}{n} = \sum_{s=0}^{n} \binom{n}{s}^2$$

Định lý 2.9

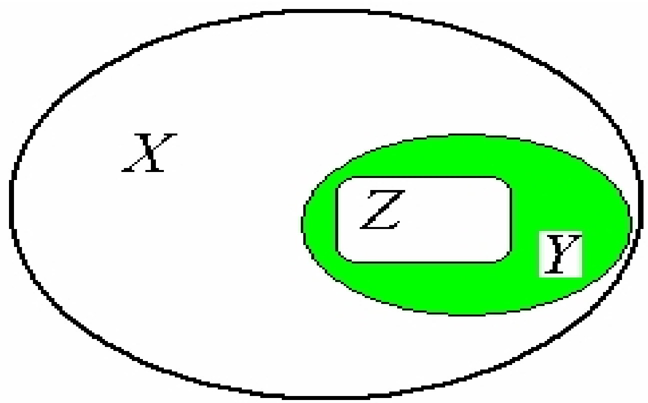
Với mọi bộ số nguyên $0 \le k \le n \le m$:

$$\binom{m}{n} . \binom{n}{k} = \binom{m}{k} . \binom{m-k}{m-n}$$

Chứng minh:

Giả sử ta có tập m phần tử X. Xét các cặp (Y, Z) các tập con của X mà: $|Z| = k$, $|Y| = n$ và $Z \subseteq Y \subseteq X$. Hiển nhiên, đại lượng ở vế trái là số các cặp này.

- Để xây dựng nên các cặp (Y, Z) ta chọn Z trước. Khi đó, ta có $\binom{m}{k}$ cách chọn tập Z.
- Chọn tập Y có n phần tử và chứa tập Z đã có k phần tử. Số cách chọn tập Y bằng số cách chọn tập $Y \setminus Z$ trong tập $X \setminus Z$ và bằng $\binom{m-k}{n-k}$.



- Sử dụng tính đối xứng của tổ hợp ta có đẳng thức cần chứng minh.

Bài toán tập con bội k-phần tử

Cho tập bội $X = (k_1 * x_1, k_2 * x_2, ..., k_n * x_n)$ và số nguyên $k \geq 0$.
Hỏi có bao nhiêu tập con bội k-phần tử của tập bội X?

Ký hiệu c_k là số các tập con bội k-phần tử của tập bội X.

Giả sử tập bội $A = (t_1 * x_1, t_2 * x_2, ... t_n * x_n)$ là một tập con bội k-phần tử của tập bội X.

Các bội của tập A phải thoả mãn hệ n+1 bất phương trình sau đây:

$$\begin{cases} 0 \leq t_i \leq k_i, i = 1,2,...,n \\ t_1 + t_2 + ... + t_n = k \end{cases} \qquad (2.4)$$

Số các nghiệm của hệ bất phương trình (2.4) sẽ là số các tập con bội k-phần tử của tập bội X.

Ta xét bài toán tập con bội k-phần tử trong một số trường hợp đặc biệt, sau đó sẽ giải quyết cho trường hợp tổng quát.

1) *Trường hợp 1*: Các bội và số k bằng nhau, $k_1 = k_2 = ... = k_n = k$

Định lý 2.10:

Số các tập con bội k phần tử của tập bội $\overline{X} = (k * x_1, k * x_2, ..., k * x_n)$ là

$$c_k = \binom{n+k-1}{k}$$

Chứng minh:

Mỗi tập con bội k phần tử của tập $\overline{X}$ có thể biểu diễn bằng dãy các số nguyên $< t_1\ t_2\ \ldots\ t_n >$ thoả mãn hệ bất phương trình sau:

$$\begin{cases} 0 \le t_i \le k, \ \ i = 1,2,\ldots,n \\ t_1 + t_2 + \ldots + t_n = k \end{cases}$$

hệ trên có thể thay bằng hệ:

$$\begin{cases} 0 \le t_i, \ \ i = 1,2,\ldots,n \\ t_1 + t_2 + \ldots + t_n = k \end{cases} \qquad (2.5)$$

Song việc giải hệ (2.5) vẫn còn khó khăn.

Đồng nhất phần tử $x_1 \equiv 1, x_2 \equiv 2, ..., x_n \equiv n$

Từ tập con $A = (t_1 * x_1, t_2 * x_2, ... t_n * x_n)$ ta thay $t_1 * x_1$ bằng t_1 chữ số 1, thay $t_2 * x_2$ bằng t_2 chữ số 2, ... , $t_n * x_n$ bằng t_n chữ số n.

Mỗi tập con k phần tử của tập $\overline{X}$ được biểu diễn bằng một dãy các số nguyên không giảm độ dài k là $<\underbrace{11...1}_{t_1}\ \underbrace{22...2}_{t_2}\ ...\ \underbrace{nn...n}_{t_n}>$ có n nhóm số.

Ta chèn n-1 số 0 vào giữa các nhóm số thì được dãy số độ dài n+k-1 là $<\underbrace{11...1}_{t_1}0\underbrace{22...2}_{t_2}0\ ...\ 0\underbrace{nn...n}_{t_n}>$

Ngược lại, nếu có một dãy số độ dài n+k-1 trong đó có n-1 số 0 là $<\underbrace{**...*}_{t_1}0\underbrace{**...*}_{t_2}0\ ...\ 0\underbrace{**...*}_{t_n}>$ ta sẽ xác định được một tập con bội k phần tử của tập bội $\overline{X}$ với:

- t_1 bằng số vị trí từ đầu tiên đến trước số 0 thứ nhất,
- t_2 bằng số vị trí từ sau số 0 thứ nhất đến trước số 0 thứ hai, ...
- t_n bằng số vị trí từ sau số 0 cuối cùng đến vị trí cuối cùng.

Như vậy, số tập con bội k phần tử của tập bội $\overline{X}$ bằng số các dãy số độ dài n+k-1 trong đó có n-1 số 0 và bằng $\binom{n+k-1}{n-1} = \binom{n+k-1}{k}$

2) *Trường hợp 2*: Các bội đều lớn hơn hoặc bằng số k, nghĩa là k_1, k_2, ... , $k_n \geq k$

Việc tìm các tập con bội k-phần tử đưa về việc giải hệ bất phương trình (2.4). Vì các bội $k_i \geq k$ nên hệ này có thể rút gọn thành hệ:

$$\begin{cases} 0 \leq t_i, \ \ i = 1,2,...,n \\ t_1 + t_2 + ... + t_n = k \end{cases}$$

và trùng với hệ (2.5) của trường hợp 1.

Vậy số tập con của trường hợp 2 trùng với số tập con của trường hợp 1 và ta có: $c_k = \begin{pmatrix} n+k-1 \\ k \end{pmatrix}$

Về mặt thực tiễn, ta thấy bội t_i (i = 1,2,...,n) của mỗi tập con k-phần tử trong trường hợp 2 luôn không vượt quá k.

Nghĩa là, các phần tử x_i từ thứ k+1 đến k_i không xuất hiện trong các tập con. Việc tăng các bội không làm tăng số tập con.

3) *Trường hợp tổng quát*

Tập bội $X = (k_1 * x_1, k_2 * x_2, ..., k_n * x_n)$ và số nguyên k.

Bội k_i có thể lớn hơn hoặc nhỏ hơn k. Như lý luận ở trường hợp 2, nếu có bội k_i nào đó lớn hơn k thì ta có thể thay bội đó bằng k mà số tập con vẫn không đổi. Vậy không mất tính tổng quát, ta có thể giả thiết:

$$k_i \le k \text{ với } i = 1, 2, ..., n$$

2.3. Sinh các tập con *k*-phần tử

Để sinh ra tất cả các tập con *k*-phần tử với số nguyên *k* cho trước, ta có thể sử dụng một trong các thuật toán sinh tất cả các tập con đã trình bày.

Công việc cần thêm vào là kiểm tra lực lượng của tập con vừa sinh ra có đúng bằng *k* hay không. Nếu đúng thì nhặt tập này ra, ngược lại thì bỏ qua tập này.

Việc kiểm tra có thể thực hiện đơn giản thông qua đẳng thức:

$$\sum_{i=1}^{n} B[i] = k$$

Song vì phải sinh tất cả các tập con nên mất nhiều thời gian tính toán.

Đồng nhất tập $X \equiv \{1, 2, \ldots, n\text{-}1, n\}$

Khi đó mỗi tập con k-phần tử của X được viết dưới dạng

$$A = \{i_1, i_2, \ldots, i_{k-1}, i_k\} \text{ với } i_j \in X \text{ và } i_r \neq i_s \text{ nếu } 1 \leq r \neq s \leq n$$

Để đảm bảo tính duy nhất của biểu diễn ta quy ước viết các phần tử theo thứ tự tăng dần:

$$1 \leq i_1 < i_2 < \ldots < i_{k-1} < i_k \leq n \qquad (2.6)$$

Vậy mỗi tập con k-phần tử của tập X tương ứng 1 - 1 với dãy các số nguyên dương tăng dần độ dài k là $< i_1\ i_2\ \ldots\ i_{k-1}\ i_k >$ mà các thành phần của nó thỏa mãn (2.6).

2.3.1. Thuật toán từ điển sinh các tập con k-phần tử

Mỗi tập con *k*-phần tử của tập *X* tương ứng 1 - 1 với dãy các số nguyên dương đơn điệu tăng độ dài *k*. Mỗi dãy như thế có thể xem như là một từ trên bảng chữ cái *X*.

Ta sắp xếp các từ trên theo thứ tự từ điển.

- từ đầu tiên sẽ là $< 1\ 2\ \ldots\ k\text{-}1\ k >$
- từ cuối cùng là $< n\text{-}k\text{+}1\ \ n\text{-}k\text{+}2\ \ldots\ n\text{-}1\ \ n >$

Xuất phát từ từ đầu tiên, thuật toán vào vòng lặp để sinh các từ còn lại cho đến khi sinh ra từ cuối cùng.

Theo thứ tự từ điển, từ đứng sau kế thừa phần bên trái nhiều nhất có thể của từ đứng trước nó.

Ta thay đổi phần bên phải bằng cách xác định vị trí thay đổi và tiến hành thay đổi để nhận được từ đứng sau từ vừa tìm được.

Thuật toán được trình bày sơ lược như sau:

1) Nhập số nguyên n và k
2) Chọn tập con đầu tiên là dãy $< 1\ 2\ \ldots\ k\text{-}1\ k >$
3) Vòng lặp:
   - Tìm vị trí thay đổi
   - Thay đổi để nhận được tập con mới
4) Lặp lại 3) cho đến khi sinh hết các tập con.

Ta chỉ cần chi tiết hóa bước 3) và 4).

Giả sử thuật toán vừa sinh được dãy $< a_1\ a_2\ \ldots\ a_{k-1}\ a_k >$. Ta cần tìm dãy kế tiếp sau nó là dãy $< b_1\ b_2\ \ldots\ b_{k-1}\ b_k >$.

Theo thứ tự từ điển, vị trí thay đổi *p* được xác định như sau:

$$p = \begin{cases} k & \text{, nếu } a_k < n \\ \max\{\ j \mid a_{j+1} - a_j \geq 2\ \} & \text{, nếu } a_k = n \end{cases}$$

Khi đó, phần bên trái vị trí *p* là phần kế thừa được giữ nguyên, có nghĩa là:

$$b_i = a_i \text{ với } i = 1, 2, \ldots, p\text{-}1.$$

Sự thay đổi được thực hiện như sau:

$$\begin{aligned} b_p &= a_p + 1 \qquad (2.7) \\ b_i &= b_p + i - p \text{ với } i = p+1, p+2, \ldots, k \end{aligned}$$

Khi xác định vị trí p trên tập con cuối cùng tương ứng với dãy số $< n\text{-}k+1\ n\text{-}k+2\ \ldots\ n\text{-}1\ n >$ ta nhận được

$$p = 0$$

Đây là điều kiện kết thúc của thuật toán.

Ví dụ 2.5: Dãy các tập con 4 phần tử của tập hợp 6 phần tử

p	*A*[1]	*A*[2]	*A*[3]	*A*[4]
4	1	2	3	4
4	1	2	3	5
3	1	2	3	6
4	1	2	4	5
3	1	2	4	6
2	1	2	5	6
4	1	3	4	5
3	1	3	4	6
2	1	3	5	6
1	1	4	5	6
4	2	3	4	5
3	2	3	4	6
2	2	3	5	6
1	2	4	5	6
0	3	4	5	6

Chú ý:

1) Giá trị đầu tiên của p là k nên $A[k]$ sẽ được tăng dần đến n. Khi đó, giá trị kế tiếp của p giảm đi 1.

Vậy ta có công thức tính p đơn giản hơn như sau:

$$p = \begin{cases} k & \text{, nếu } a_k < n \\ p - 1 & \text{, nếu } a_k = n \end{cases}$$

2) Việc thay thế theo công thức (2.7) có thể thực hiện gọn lại bằng cách thay thế từ phải sang trái theo công thức:

$$b_i = a_p + i - p + 1 \quad \text{với } i = k, k\text{-}1, \dots, p+1, p$$

Thuật toán 2.11

Đầu vào: Hai số nguyên n và k

Đầu ra: Dãy tất cả các tập con k phần tử của tập hợp $\{1, 2, \ldots, n\text{-}1, n\}$ được sắp theo thứ tự từ điển

```
Begin
  Input n, k ;
  for i ← 1 to k do A[i] ← i ;
  p ← k ;
  while p ≥ 1 do
     { print A[1..k] ;
       if A[k] < n then p ← k else p ← p - 1 ;
       if p ≥ 1 then
            for i ← k downto p do A[i] ← A[p] + i - p +1 } ;
End .
```

Độ phức tạp của thuật toán

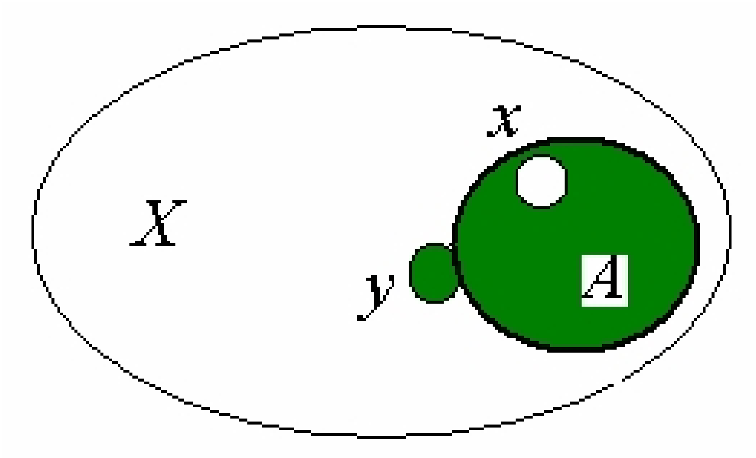
Để sinh ra một tập con k phần tử, chu trình 5-9 có độ phức tạp O(k).

Vậy độ phức tạp tổng thể của Thuật toán 2.11 là $O\left(\binom{n}{k}.k\right)$

2.3.2. Sinh các tập con k-phần tử bằng cách thay thế

Giả sử X là một tập n phần tử. Giả sử A là một tập con k phần tử của X. Ta bỏ đi phần tử x ở trong A và thêm vào phần tử y thuộc X nhưng ngoài A, ta nhận được tập con mới $A' = A \setminus \{x\} \cup \{y\}$ cũng có k phần tử.

Việc làm này tạo nên một tập con mới có k phần tử từ một tập con k phần tử bằng cách thay thế chỉ một phần tử.



Ta có thể sắp xếp tất cả các tập con *k*-phần tử của một tập *n* phần tử thành một dãy theo thứ tự:

Tập con sau tạo bởi từ tập con trước đó nhờ việc thay thế một phần tử.

Thuật toán tìm dãy tập con

Ký hiệu: $G(n,k)$ là danh sách liệt kê tất cả các tập con k-phần tử của tập $\{1,2,\ldots,n-1,n\}$ thỏa mãn ba tính chất sau đây:

1) Tập con đầu tiên là $\{1, 2,\ldots, k-1, k\}$
2) Tập con cuối cùng là $\{1, 2,\ldots, k-1, n\}$
3) Tập con sau tạo bởi từ tập con trước đó nhờ việc thay thế một phần tử.

Định lý 2.12: Danh sách $G(n,k)$ thỏa mãn công thức đệ quy dưới đây:

$$G(n,k) = G(n-1,k)\ ,\ G^{*}(n-1,k-1) \cup \{n\} \qquad (2.8)$$

trong đó, $G^{*}(n-1,k-1)\cup\{n\}$ là danh sách tạo nên từ việc đảo ngược thứ tự của $G(n-1,k-1)$ và thêm phần tử n vào từng tập con.

Chứng minh:

Để chứng minh công thức (2.8) ta chỉ việc kiểm tra xem danh sách ở vế phải có thỏa mãn ba tính chất nêu ở trên không.

- Tập con đầu tiên của nó là tập con đầu tiên của danh sách $G(n\text{-}1,k)$ và chính là tập con $\{1, 2,\ldots, k\text{-}1, k\}$.

- Tập con cuối cùng của danh sách ở vế phải (2.8) là tập con cuối cùng của danh sách $G^{*}(n\text{-}1,k\text{-}1) \cup\{n\}$. Vậy tập con này phải là tập con đầu tiên của danh sách $G(n\text{-}1,k\text{-}1) \cup\{n\}$ trước khi đảo ngược, và nó là tập con $\{1, 2,\ldots, k\text{-}1, n\}$.

- Ta còn phải kiểm tra tính chất bất biến của danh sách thể hiện ở thứ tự của nó. Hiển nhiên, tính chất đó thỏa mãn trong từng danh sách con.

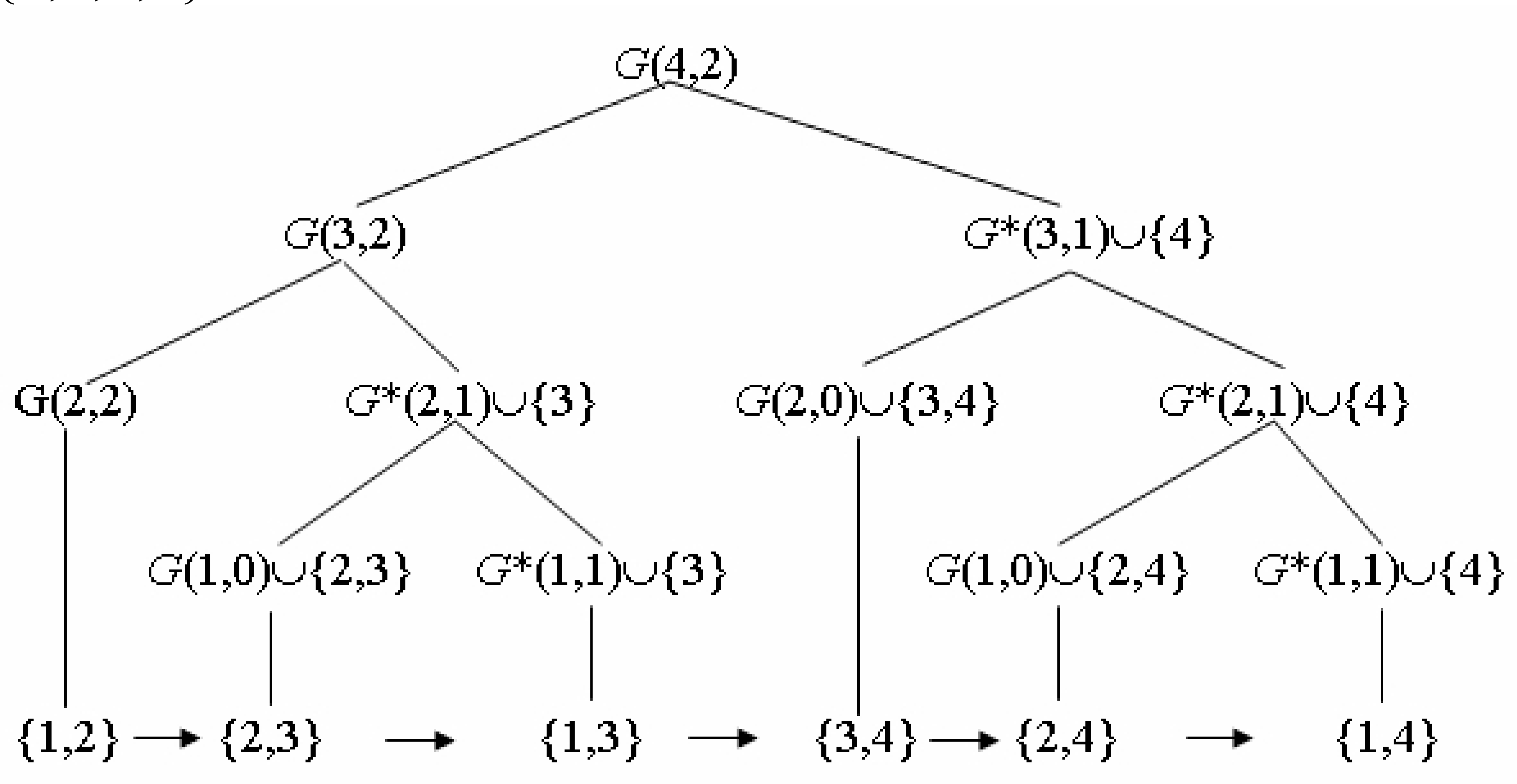
Ta xem tại "điểm nối" của chúng tính chất bất biến có thỏa mãn hay không:

$$\begin{array}{ll} G(n\text{-}1,k) & \uparrow\{1, 2,\ldots, k\text{-}2, k\text{-}1, n\text{-}1\} \\ \hline G^{*}(n\text{-}1,k\text{-}1) \cup \{n\} & \downarrow\{1, 2,\ldots, k\text{-}2, n\text{-}1, n\} \end{array}$$

Tại điểm nối tập con dưới tạo bởi tập con trên bằng cách bỏ đi phần tử k-1 và thêm vào phần tử n. Tính chất 3) thỏa mãn. Công thức (2.8) là đúng đắn.

Công thức (2.8) cho ta một thuật toán đệ quy đơn giản để sinh tất cả các tập con k phần tử của tập n phần tử.

Ví dụ 2.6: Xây dựng danh sách $G(4,2)$ liệt kê các tập con 2 phần tử của tập $\{1,2,3,4\}$.

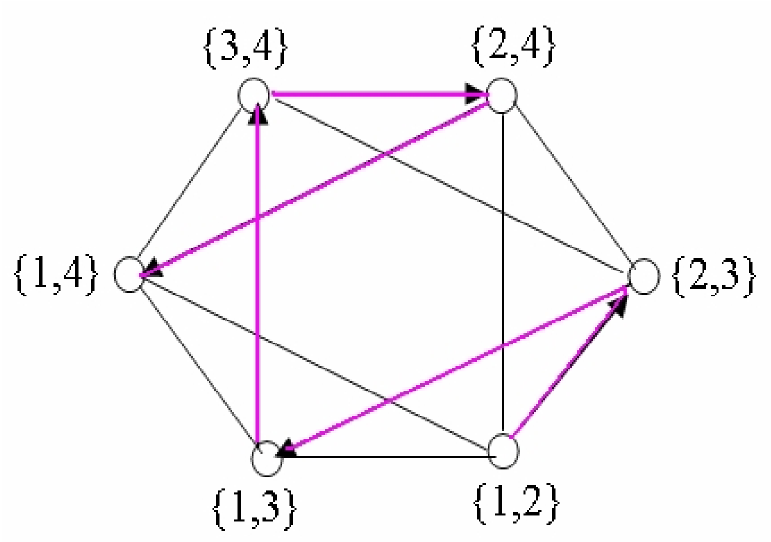


Ý nghĩa đồ thị của thuật toán

Xây dựng đồ thị vô hướng với:

- tập đỉnh là các tập con *k* phần tử.
- hai đỉnh *A* và *B* có cạnh nối với nhau khi và chỉ khi $|A \cap B| = k\text{-}1$.

Danh sách $G(n,k)$ sẽ cho ta một đường đi Hamilton trên đồ thị này.



Thuật toán 2.13

Đầu vào: Hai số nguyên *n* và *k*

Đầu ra: Dãy tất cả các tập con *k*-phần tử của tập hợp {1, 2,…, *n*-1, *n*} mà tập con sau tạo bởi tập con trước đó bằng cách thay một phần tử

```
procedure DSG(nt, kt, Gsao)
begin
  if (kt = 0) or (kt = nt) then print A[1..n] ;
  if (kt > 0) and (kt < nt) then
        if not Gsao
            then  { DSG(nt-1, kt, false) ; A[nt] ← 1 ;
                    DSG(nt-1, kt-1, true) ; A[nt] ← 0 }
            else  { A[nt] ← 1 ; DSG(nt-1, kt-1, false) ;
                    A[nt] ← 0 ; DSG(nt-1, kt, true) }
end ;
```

```
Begin
    input  n, k ;
    A[1..n] ← 0 ;
    DSG(n, k, false) ;
End .
```

Việc xây dựng phương án lặp cho thuật toán trình bày ở trên xin dành cho đọc giả.

BÀI TẬP CHƯƠNG 2

2.1. Một lớp Tin học có 7 sinh viên nam và 5 sinh viên nữ. Hỏi có bao nhiêu cách chọn 6 sinh viên trong lớp trong đó có ít nhất có một sinh viên nữ theo học chuyên đề về mạng?

2.2. Chứng minh rằng tích của k số tự nhiên liên tiếp tùy ý đều chia hết cho $k!$.

2.3. Chứng minh các đồng nhất thức dưới đây:

$$1)\ \binom{n+1}{k} = \binom{n}{k} + \binom{n-1}{k-1} + \binom{n-2}{k-2} + \ldots + \binom{n-k}{0}$$

$$2)\ \binom{n+1}{k+1} = \binom{n}{k} + \binom{n-1}{k} + \binom{n-2}{k} + \ldots + \binom{k}{k}$$

2.4. Chứng minh đồng nhất thức:

$$n.\binom{n-1}{k-1} = k.\binom{n}{k}$$

Gợi ý: *Tính số các cặp (a, K) với a* ∈ *K và K là tập con k phần tử của tập n phần tử.*

2.5. Chứng minh rằng:

$$(x_1 + x_2 + \ldots + x_q)^n = \sum_{n_1+n_2+\ldots+n_q=n} \binom{n}{n_1\, n_2\, \ldots n_q} x_1^{n_1} x_2^{n_2} \ldots x_q^{n_q}$$

với $\binom{n}{n_1\ n_2\ \ldots\ n_q} = \dfrac{n!}{n_1!n_2!\ldots n_q!}$, đại lượng này được gọi là tổ hợp chập n_1, n_2, …, n_q của n.

2.6. Cho m+1 vectơ tọa độ nguyên n chiều v_1, v_2, …, v_m, t. Lập thuật toán kiểm tra xem có tồn tại một tập con $I \subseteq \{1, 2, \ldots, m\}$, sao cho:

$$t = \sum_{i \in I} v_i$$

2.7. Chứng minh rằng, với các số tự nhiên *p, q, n* tùy ý thì:

1) $[p+q]_n = \sum_{k=0}^{n} \binom{n}{k} [p]_k [q]_{n-k}$

2) $[p+q]^n = \sum_{k=0}^{n} \binom{n}{k} [p]^k [q]^{n-k}$

Gợi ý: - *Trường hợp a*) *tính số tất cả các đơn ánh* $f : X \to Y$, *với* $|X| = n$, $|P| = p$, $|Q| = q$ *và* $P \cap Q = \varnothing$.

- *Trương hợp b*) *tìm thể hiện tương tự của việc xếp đặt có thứ tự n* "*đối tượng*" *vào p*+*q* "*hộp*".

2.8. Mỗi dãy nhị phân độ dài *n* biểu diễn một tập con của tập *n* phần tử tương ứng với một số nguyên hệ thập phân. Chứng minh rằng, dãy số nguyên tương ứng với các dãy nhị phân sinh bởi Thuật toán 2.13 sinh các tập con bằng phương pháp thêm/bớt trùng với dãy số nguyên sau đây:

$$i \text{ XOR } (i \text{ SHR } 1) \text{, với } i = 0, 1, 2, \ldots, 2^n\text{-}1.$$

2.9. Giả sử có *n* tệp văn bản với dung lượng v_1, v_2, …, v_n. Xây dựng thuật toán phân chia các tệp này cho hai máy in đồng thời sao cho thời gian in là ngắn nhất có thể.

2.10. Cho n điểm trên mặt phẳng có tọa độ là (x_i, y_i) , $i = 1, 2, \ldots, n$ được tô bằng một trong 3 màu: xanh, đỏ, vàng. Xây dựng thuật toán tìm đa giác k đỉnh cùng màu ($k \geq 3$) có chu vi lớn nhất.

2.11. Chứng minh phụ thuộc đệ quy sau đây:

$$G(n,k) = G(n\text{-}1,k),\ G(n\text{-}2,k\text{-}2) \cup \{n\text{-}1, n\},\ G^*(n\text{-}2,k\text{-}1) \cup \{n\}$$