CÁC TẬP CON CỦA MỘT TẬP HỢP

Bài giảng chuyên đề "*Một số thuật toán tổ hợp*"

Lê Hồng Phương[1]

[1]Khoa Toán–Cơ–Tin học
Trường Đại học Khoa học Tự nhiên, ĐHQG Hà Nội
<*phuonglh@gmail.com*>

08/2012

Nội dung

Nội dung

Nội dung

1. Giới thiệu
2. Sinh các tập con
   - Thuật toán cộng một
   - Thuật toán đệ quy
   - Thuật toán mã Gray
3. Sinh các tập con k phần tử
   - Các hệ số nhị thức
   - Thuật toán sinh theo thứ tự từ điển
   - Một số thuật toán khác
4. Tóm lược

Nội dung

1. Giới thiệu
2. Sinh các tập con
   - Thuật toán cộng một
   - Thuật toán đệ quy
   - Thuật toán mã Gray
3. Sinh các tập con k phần tử
   - Các hệ số nhị thức
   - Thuật toán sinh theo thứ tự từ điển
   - Một số thuật toán khác
4. Tóm lược

Nội dung

Nội dung

Nội dung

Giới thiệu

- Cho $X = \{x_1, x_2, \ldots, x_n\}$ là một tập hợp gồm n phần tử.
- Mỗi tập con Y của tập X có thể được biểu diễn bằng một dãy nhị phân $\langle b_1, b_2, \ldots, b_n \rangle$ xác định như sau:

$$b_i = \begin{cases} 1, & \text{nếu } x_i \in Y, \\ 0, & \text{nếu } x_i \notin Y. \end{cases}$$

- Nói riêng, tập $Y \equiv \emptyset$ tương ứng với dãy $\langle 0, 0, \ldots, 0 \rangle$ và tập $Y \equiv X$ tương ứng với dãy $\langle 1, 1, \ldots, 1 \rangle$.
- Ta thấy $b_i, i = 1, 2, \ldots, n$ nhận giá trị nhị phân nên số tập con của tập X là 2^n.
- Nếu ta coi mỗi dãy nhị phân là biểu diễn nhị phân của một số nguyên không âm thì mỗi tập con của tập X ứng với một số nguyên trong đoạn $[0, 2^n - 1]$.

- Cho $X = \{x_1, x_2, \ldots, x_n\}$ là một tập hợp gồm n phần tử.
- Mỗi tập con Y của tập X có thể được biểu diễn bằng một dãy nhị phân $\langle b_1, b_2, \ldots, b_n\rangle$ xác định như sau:

$$b_i = \begin{cases} 1, & \text{nếu } x_i \in Y, \\ 0, & \text{nếu } x_i \notin Y. \end{cases}$$

- Nói riêng, tập $Y \equiv \emptyset$ tương ứng với dãy $\langle 0, 0, \ldots, 0\rangle$ và tập $Y \equiv X$ tương ứng với dãy $\langle 1, 1, \ldots, 1\rangle$.
- Ta thấy $b_i, i = 1, 2, \ldots, n$ nhận giá trị nhị phân nên số tập con của tập X là 2^n.
- Nếu ta coi mỗi dãy nhị phân là biểu diễn nhị phân của một số nguyên không âm thì mỗi tập con của tập X ứng với một số nguyên trong đoạn $[0, 2^n - 1]$.

Giới thiệu

Ta quan tâm tới hai bài toán:

1. Sinh mọi tập con của tập X;
2. Sinh mọi tập con gồm k phần tử của tập X với $k \leq n$.

Nội dung

Sinh các tập con

Bài toán

Hãy liệt kê mọi tập con của một tập hợp gồm n phần tử.

Ví dụ, các tập con của tập gồm 3 phần tử {1, 2, 3 } là:

$$\begin{aligned}
&\{\},\\
&\{1\}, \{2\}, \{3\},\\
&\{1, 2\}, \{1, 3\}, \{2, 3\},\\
&\{1, 2, 3\}.
\end{aligned}$$

Chú ý:

- Số tập con của một tập gồm n phần tử là 2^n, là rất lớn nếu n lớn.
- Vì vậy, bài toán này chỉ có thể giải được nếu n nhỏ ($n \leq 20$).

Nội dung

Thuật toán cộng một

Biểu diễn dãy nhị phân của các tập hợp gợi ý cho ta một phương pháp đơn giản để sinh mọi tập hợp của n phần tử như sau:

1. Xuất phát từ tập rỗng, ứng với số $k = 0$ hay dãy nhị phân $a^0 = \langle 0, 0, \dots, 0 \rangle$;
2. Trong mỗi bước, số k được cộng thêm 1 và tìm các biểu diễn nhị phân tương ứng của nó. Ví dụ, 5 dãy nhị phân tiếp theo là

$$\begin{aligned} a^1 &= \langle 0, 0, \dots, 0, 0, 1 \rangle \\ a^2 &= \langle 0, 0, \dots, 0, 1, 0 \rangle \\ a^3 &= \langle 0, 0, \dots, 0, 1, 1 \rangle \\ a^4 &= \langle 0, 0, \dots, 1, 0, 0 \rangle \\ a^5 &= \langle 0, 0, \dots, 1, 0, 1 \rangle \end{aligned}$$

3. Dừng thuật toán khi $k = 2^n - 1$ hay khi dãy nhị phân là $\langle 1, 1, \dots, 1, 1 \rangle$.

Thuật toán cộng một

Ta có thể tăng tốc độ của thuật toán dựa trên quan sát đơn giản: *dãy nhị phân đứng sau có thể được sinh từ dãy nhị phân đứng trước bằng cách quy nạp.*

Giả sử đã sinh được dãy nhị phân $a^i = \langle b_0, b_1, \ldots, b_n \rangle$, dãy a^{i+1} được tìm bằng cách:

1. Xét các bít b_j với j giảm dần, bắt đầu từ n.
2. Lặp, trong khi $j \geq 1$:
   - nếu $b_j = 1$ thì đặt $b_j = 0$ và tiếp tục xét b_{j-1};
   - nếu $b_j = 0$ thì đặt $b_j = 1$ và dừng vòng lặp.

Sinh tập con tiếp theo:

```
void nextSubset(int a[], int n) {
  int j = n - 1;
  while (j >= 0) {
    if (a[j] == 1)
      a[j] = 0;
    else {
      a[j] = 1;
      break;
    }
    j--;
  }
}
```

Thuật toán cộng một

Sinh mọi tập con bằng thuật toán cộng một:

```
void enumerate(int a[], int n) {
  int i, j;
  unsigned long max = exponential(2, n);

  for (j = 0; j < n; j++)
    a[j] = 0;

  for (i = 0; i < max; i++) {
    printSubset(a, n);
    nextSubset(a, n);
  }
}
```

Thuật toán cộng một

Với $n = 4$, các dãy nhị phân sinh bởi thuật toán là:

$$
\begin{array}{llll}
0 & \langle 0,0,0,0\rangle & 8 & \langle 1,0,0,0\rangle \\
1 & \langle 0,0,0,1\rangle & 9 & \langle 1,0,0,1\rangle \\
2 & \langle 0,0,1,0\rangle & 10 & \langle 1,0,1,0\rangle \\
3 & \langle 0,0,1,1\rangle & 11 & \langle 1,0,1,1\rangle \\
4 & \langle 0,1,0,0\rangle & 12 & \langle 1,1,0,0\rangle \\
5 & \langle 0,1,0,1\rangle & 13 & \langle 1,1,0,1\rangle \\
6 & \langle 0,1,1,0\rangle & 14 & \langle 1,1,1,0\rangle \\
7 & \langle 0,1,1,1\rangle & 15 & \langle 1,1,1,1\rangle
\end{array}
$$

Nội dung

1. Giới thiệu
2. Sinh các tập con
   - Thuật toán cộng một
   - Thuật toán đệ quy
   - Thuật toán mã Gray
3. Sinh các tập con k phần tử
   - Các hệ số nhị thức
   - Thuật toán sinh theo thứ tự từ điển
   - Một số thuật toán khác
4. Tóm lược

Thuật toán đệ quy

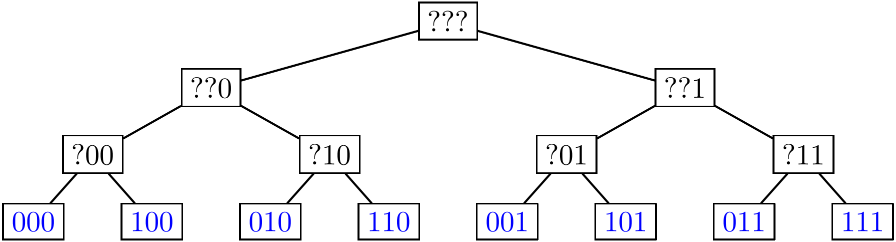
Ta có thể liệt kê mọi dãy nhị phân độ dài n bằng thuật toán đệ quy:

```
void enumerate(int a[], int n, int k) {
  if (k < 0) {
    printSubset(a, n);
  }
  else {
    a[k] = 0;
    enumerate(a, n, k - 1);
    a[k] = 1;
    enumerate(a, n, k - 1);
  }
}
```

Chương trình chính gọi hàm `enumerate(a, n, n - 1)`.

Thuật toán đệ quy

Ví dụ, với $n = 3$, cây đệ quy tìm các dãy nhị phân được minh họa trong hình sau:



Ta thấy thuật toán đệ quy thực hiện duyệt cây theo cách duyệt tiền thứ tự và xử lí các nút lá của cây.

Bài tập

Bài tập 1. Giải thích thuật toán đệ quy và vẽ sơ đồ cây đệ quy tìm các dãy nhị phân ứng với các tập con của tập gồm 4 phần tử.

Bài tập 2. Viết chương trình cài đặt các thuật toán cộng một và đệ quy sinh các tập con của một tập hợp. So sánh thời gian chạy của hai thuật toán này trên các tập có kích thước tương tự nhau.

Nội dung

Thuật toán mã Gray

- Mã Gray n-bít là một danh sách gồm 2^n dãy nhị phân độ dài n trong đó dãy tiếp theo chỉ khác dãy đứng trước ở một bít.
- Mã Gray được phát minh năm 1947 bởi Frank Gray, một nghiên cứu viên ở Bell Labs, sau đó được công bố năm 1953 trong [1].
- Ban đầu được sử dụng trong các hệ thống chuyển mạch cơ điện tử; ngày nay, mã Gray được sử dụng rộng rãi trong việc phát hiện lỗi của các hệ thống liên lạc điện tử.

[1] F. Gray, "Pulse code communication," 1953, U.S. Patent 2,632,058

Thuật toán mã Gray

- Mã Gray còn được gọi là "*mã nhị phân phản xạ*" vì mã Gray n bít được xây dựng đệ quy từ mã Gray $n-1$ bít bằng cách phản xạ mã này.
- Phản xạ: liệt kê các phần tử của danh sách dãy nhị phân theo thứ tự ngược lại.
- Các bước cụ thể để sinh mã Gray n bít như sau:

Thuật toán mã Gray

1. Xuất phát từ mã Gray $n-1$ bít là danh sách gồm $k = 2^{n-1}$ dãy nhị phân:
$$\alpha_1, \alpha_2, \dots, \alpha_{k-1}, \alpha_k$$
2. Phản xạ mã Gray này, tức liệt kê các dãy nhị phân của nó theo thứ tự ngược lại:
$$\alpha_k, \alpha_{k-1}, \dots, \alpha_2, \alpha_1$$
3. Đặt bít 0 lên trước các dãy trong danh sách ban đầu:
$$0\alpha_1, 0\alpha_2, \dots, 0\alpha_{k-1}, 0\alpha_k$$
4. Đặt bít 1 lên trước các dãy trong danh sách phản xạ:
$$1\alpha_k, 1\alpha_{k-1}, \dots, 1\alpha_2, 1\alpha_1$$
5. Ghép hai danh sách này lại sẽ thu được mã Gray n bít:
$$0\alpha_1, 0\alpha_2, \dots, 0\alpha_k, 1\alpha_k, 1\alpha_{k-1}, \dots, 1\alpha_2, 1\alpha_1.$$

Thuật toán mã Gray

Ví dụ, mã Gray với $n = 3$ được sinh từ mã Gray với $n = 2$ như sau:

Mã 2-bít	$00, 01, 11, 10$	
Mã phản xạ		$10, 11, 01, 00$
Đặt 0 lên trước mã ban đầu	$000, 001, 011, 010$	
Đặt 1 lên trước mã phản xạ		$110, 111, 101, 100$
Ghép hai mã	$000, 001, 011, 010,$	$110, 111, 101, 100$

Thuật toán mã Gray

Mã Gray 1 bít là $G_1 = \langle 0, 1 \rangle$. Mã này có thể được sinh từ mã Gray 0 bít $G_0 = \langle \epsilon \rangle$ chỉ gồm một dãy rỗng theo cách trên.

Trong quá trình sinh mã G_{n+1} từ G_n ta thấy một số tính chất sau:

- Nếu coi mỗi dãy nhị phân trong mã G_n là một số nguyên (trong cơ số 10) thì G_n là một hoán vị của dãy số $\langle 0, 1, \ldots, 2^n - 1 \rangle$.
- G_n được "nhúng" vào nửa đầu của G_{n+1}.
- Mỗi dãy của G_n chỉ khác với dãy đứng trước nó một bít.[2].
- Dãy cuối cùng của G_n chỉ khác dãy đầu tiên một bít.

[2] Nói cách khác, khoảng cách Hamming giữa hai dãy liên tiếp của G_n là 1.

Bài tập

Bài tập 3. Sinh mã Gray 4 bít bằng thuật toán đệ quy nêu trên.

Bài tập 4. Viết chương trình cài đặt thuật toán tìm mọi tập con của tập hợp bằng thuật toán sinh mã Gray n bít đệ quy nêu trên.

Thuật toán mã Gray

Ta có thể tìm mã G_{n+1} từ mã G_n bằng một thủ tục đệ quy.

Tuy nhiên, từ các tính chất của mã Gray ở trên, ta có thể tìm mã Gray bằng một thuật toán nhanh hơn dựa trên quan sát sau:

$\boxed{\text{Chuỗi thứ } i \text{ trong } G_n \text{ là biểu diễn nhị phân của số } (i/2) \oplus i}$,

trong đó $i/2$ là phép chia nguyên của i cho 2 và $\oplus$ là phép toán `xor`.

- *Nhắc lại:* phép `xor` giữa hai bít a và b cho giá trị 1 nếu chỉ a hoặc b là 1. Cụ thể:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Ta có thể tìm mã G_{n+1} từ mã G_n bằng một thủ tục đệ quy.

Tuy nhiên, từ các tính chất của mã Gray ở trên, ta có thể tìm mã Gray bằng một thuật toán nhanh hơn dựa trên quan sát sau:

$$\boxed{\text{Chuỗi thứ } i \text{ trong } G_n \text{ là biểu diễn nhị phân của số } (i/2) \oplus i},$$

trong đó $i/2$ là phép chia nguyên của i cho 2 và $\oplus$ là phép toán `xor`.

- *Nhắc lại:* phép `xor` giữa hai bít a và b cho giá trị 1 nếu chỉ a hoặc b là 1. Cụ thể:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Thuật toán mã Gray

Ví dụ, các chuỗi nhị phân thứ $i, \forall i = 0, 1, \ldots, 7$ trong mã G_3:

$i_{(10)}$	$i_{(2)}$	$(i/2)_{(2)}$	$(i/2) \oplus i$	G_3
0	000	000	$000 \oplus 000$	000
1	001	000	$000 \oplus 001$	001
2	010	001	$001 \oplus 010$	011
3	011	001	$001 \oplus 011$	010
4	100	010	$010 \oplus 100$	110
5	101	010	$010 \oplus 101$	111
6	110	011	$011 \oplus 110$	101
7	111	011	$011 \oplus 111$	100

Cột đầu tiên là số i trong cơ số 10; cột thứ hai là biểu diễn của nó trong cơ số 2; cột thứ ba là số $i/2$ trong cơ số 2; cột thứ tư là phép toán `xor` giữa hai số $i/2$ và i trong cơ số 2; cột cuối cùng chính là chuỗi nhị phân thứ i trong mã G_3.

Bài tập

Bài tập 5. Sinh mã Gray 4 bít theo thuật toán tính nhanh như bảng trên (lập bảng tính G_4).

Bài tập 6. Viết chương trình cài đặt thuật toán tìm mọi tập con của tập hợp bằng thuật toán sinh mã Gray n bít theo thuật toán tính nhanh như trên.

Thuật toán mã Gray

Ta cũng có thể tìm chuỗi mã Gray thứ $i+1$ từ chuỗi mã Gray thứ i dựa trên quan sát sau:

1. Nếu chuỗi thứ i có một số chẵn bít 1 thì ta đảo bít cuối cùng của nó sẽ có chuỗi thứ $i+1$;
2. Nếu chuỗi thứ i có một số lẻ bít 1 thì ta tìm bít 1 ở bên phải nhất của chuỗi và đảo bít ở bên trái bít đó.

Sử dụng quy tắc đảo bít này, ta sinh được mã Gray G_4 như sau:

$$000 \overset{(1)}{\Rightarrow} 001 \overset{(2)}{\Rightarrow} 011 \overset{(1)}{\Rightarrow} 010 \overset{(2)}{\Rightarrow} 110 \overset{(1)}{\Rightarrow} 111 \overset{(2)}{\Rightarrow} 101 \overset{(1)}{\Rightarrow} 100.$$

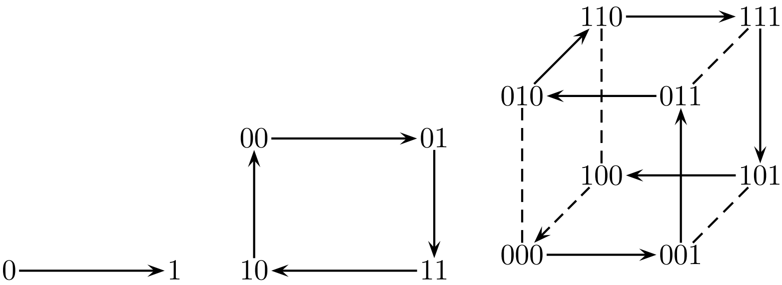
Bài tập

Bài tập 7. Viết chương trình cài đặt thuật toán tìm mọi tập con của tập hợp bằng thuật toán sinh mã Gray n bít theo thuật toán đảo bít.

Bài tập 8. Viết chương trình giải bài toán sau. Có n gói kẹo, trong mỗi gói kẹo chứa a_i cái kẹo, $i = 1, 2, \dots, n$. Hãy tìm cách chia n gói kẹo đó thành hai phần sao cho hiệu số kẹo của hai phần là nhỏ nhất.

Mô tả hình học

Nếu coi mỗi dãy nhị phân trong mã Gray n bít là một đỉnh của đồ thị; hai đỉnh kề nhau nếu hai dãy nhị phân chỉ khác nhau ở 1 bít thì mã Gray n bít tương ứng với một đường đi Hamilton trên đồ thị này.



$n = 1$ $n = 2$ $n = 3$

Nội dung

Nội dung

1. Giới thiệu
2. Sinh các tập con
   - Thuật toán cộng một
   - Thuật toán đệ quy
   - Thuật toán mã Gray
3. Sinh các tập con k phần tử
   - Các hệ số nhị thức
   - Thuật toán sinh theo thứ tự từ điển
   - Một số thuật toán khác
4. Tóm lược

Các k–tổ hợp

- Ta xét các tập con k phần tử của một tập hợp có n phần tử. Các tập này gọi là các *k–tổ hợp*.
- Ví dụ, các 3-tổ hợp của 5 phần tử $\{a, b, c, d, e\}$ là:

$$abc, \quad abd, \quad abe, \quad acd, \quad ace, \quad ade, \quad bcd, \quad bce, \quad bde, \quad cde.$$

- Ta đếm được 10 tổ hợp. Tổng quát, có bao nhiêu k–tổ hợp của n phần tử?

Các k–tổ hợp

- Từ mục đếm các hoán vị, ta thấy có $n(n-1)\cdots(n-k+1)$ cách chọn k phần tử đầu tiên của một hoán vị gồm n phần tử; và mỗi k–tổ hợp xuất hiện đúng $k!$ lần trong các xếp đặt này, vì mỗi tổ hợp xuất hiện trong mọi hoán vị của nó.
- Vì vậy số k–tổ hợp của n phần tử là:

$$\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k(k-1)\cdots(1)}.$$

- Ta dùng kí hiệu $\binom{n}{k}$, hoặc C_n^k để chỉ số tổ hợp. Đại lượng này còn được gọi là các *hệ số nhị thức*.

Các hệ số nhị thức

Các hệ số nhị thức ứng với n một số giá trị đầu tiên của k.

$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$	$\binom{n}{3}$	$\binom{n}{4}$	$\binom{n}{5}$	$\binom{n}{6}$	$\binom{n}{7}$	$\binom{n}{8}$
1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	2	1	0	0	0	0	0	0
1	3	3	1	0	0	0	0	0
1	4	6	4	1	0	0	0	0
1	5	10	10	5	1	0	0	0
1	6	15	20	15	6	1	0	0
1	7	21	35	35	21	7	1	0
1	8	28	56	70	56	28	8	1

Tam giác Pascal – Blaise Pascal, *Traité du triangle arithmétique*, 1653.

Các hệ số nhị thức – Biểu diễn bằng giai thừa

Một số kĩ thuật cơ bản để tính toán các hệ số nhị thức được liệt kê ngắn gọn như dưới đây.

1. Biểu diễn bằng giai thừa. Ta có công thức biểu diễn:

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

Hai phương pháp chứng minh công thức này:

- Chứng minh bằng tính toán
- Chứng minh bằng lập luận

2. Tính chất đối xứng.

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

(Chú ý: khi $k > n$ thì hệ số nhị thức bằng 0.)

- Có thể chứng minh tính chất này bằng công thức tính hệ số nhị thức.
- Tuy nhiên, tính chất này là dễ hiểu vì số cách chọn các tập k phần tử từ n phần tử cũng chính là số cách chọn các tập bù gồm $n-k$ phần tử từ n phần tử. Do đó số k–tổ hợp chính là số $n-k$ tổ hợp.
- Chú ý rằng tính chất đối xứng được thể hiện trong mỗi hàng của tam giác Pascal.

3. Công thức cộng. Ta có tính chất cơ bản

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

Tính chất này có thể được chứng minh bằng cách tự nhiên như sau. Để chọn k phần tử từ n phần tử $a_1, a_2, \ldots, a_n$, ta có thể chọn bằng một trong hai cách:

- chọn phần tử a_1, khi đó ta cần chọn $k-1$ phần tử nữa từ $n-1$ phần tử còn lại; số cách chọn này là $\binom{n-1}{k-1}$;
- không chọn phần tử a_1, khi đó ta cần chọn k phần tử từ $n-1$ phần tử còn lại; số cách chọn này là $\binom{n-1}{k}$.

Như vậy, số k–tổ hợp của n phần tử chính là tổng của hai đại lượng trên.

3. Công thức cộng. Ta có tính chất cơ bản

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

Tính chất này có thể được chứng minh bằng cách tự nhiên như sau. Để chọn k phần tử từ n phần tử $a_1, a_2, \dots, a_n$, ta có thể chọn bằng một trong hai cách:

- chọn phần tử a_1, khi đó ta cần chọn $k-1$ phần tử nữa từ $n-1$ phần tử còn lại; số cách chọn này là $\binom{n-1}{k-1}$;
- không chọn phần tử a_1, khi đó ta cần chọn k phần tử từ $n-1$ phần tử còn lại; số cách chọn này là $\binom{n-1}{k}$.

Như vậy, số k–tổ hợp của n phần tử chính là tổng của hai đại lượng trên.

Các hệ số nhị thức – Công thức cộng

Trong tam giác Pascal, mọi giá trị đều là tổng của hai giá trị ở hàng trước và bên trái nó. Ví dụ

5	10
	15

$(15 = 5 + 10)$

21	7
	28

$(21 = 7 + 28)$

$\binom{n}{0}$	$\binom{n}{1}$	$\binom{n}{2}$	$\binom{n}{3}$	$\binom{n}{4}$	$\binom{n}{5}$	$\binom{n}{6}$	$\binom{n}{7}$	$\binom{n}{8}$
1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	2	1	0	0	0	0	0	0
1	3	3	1	0	0	0	0	0
1	4	6	4	1	0	0	0	0
1	5	10	10	5	1	0	0	0
1	6	15	20	15	6	1	0	0
1	7	21	35	35	21	7	1	0
1	8	28	56	70	56	28	8	1

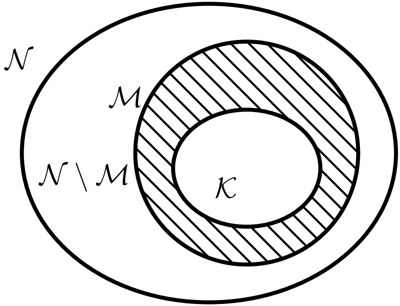
4. Công thức tích. Ta có công thức tích sau:

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

- Tương tự như trên, công thức này có thể được chứng minh dễ dàng bằng cách khai triển tổ hợp.
- Tuy nhiên, ta có thể chứng minh công thức này bằng cách suy luận.

Các hệ số nhị thức – Công thức tích

- Xét ba tập hợp $\mathcal{K}, \mathcal{M}, \mathcal{N}$. Giả sử $|\mathcal{K}| = k, |\mathcal{M}| = m, |\mathcal{N}| = n$.
- Ta thấy đại lượng bên trái của công thức chính là số các cặp tập hợp $(\mathcal{M}, \mathcal{K})$ thoả mãn điều kiện $\mathcal{K} \subset \mathcal{M} \subset \mathcal{N}$ vì tập $\mathcal{M}$ có thể chọn bằng $\binom{n}{m}$ cách và tập $\mathcal{K}$ có thể chọn bằng $\binom{m}{k}$ cách.



Các hệ số nhị thức – Công thức tích

Ta cũng có thể tính số cặp $(\mathcal{M}, \mathcal{K})$ này bằng cách khác như sau.

- Chọn tập $\mathcal{K}$ trước, có $\binom{n}{k}$ cách chọn tập $\mathcal{K}$.
- Với mỗi cách chọn $\mathcal{K}$, ta chọn tập $\mathcal{M}$ sao cho $\mathcal{K} \subset \mathcal{M} \subset \mathcal{N}$. Số cách chọn tập $\mathcal{M}$ bằng số cách chọn tập $\mathcal{M} \setminus K$ trong tập $\mathcal{N} \setminus \mathcal{K}$ và bằng $\binom{n-k}{m-k}$.
- Từ đó, số cách chọn các cặp $(\mathcal{M}, \mathcal{K})$ là $\binom{n}{k}\binom{n-k}{m-k}$.

Các hệ số nhị thức – Công thức tích

Công thức tích ở trên rất hữu dụng khi xử lí các công thức tổ hợp trong đó có một chỉ số (ở đây là m) xuất hiện đồng thời ở cả vị trí dưới và trên trong các tổ hợp; ta có thể chuyển chỉ số đó xuống vị trí dưới nhờ công thức tích.

Với $k = 1$, ta có

$$\binom{n}{m}\binom{m}{1} = \binom{n}{1}\binom{n-1}{m-1},$$

hay

$$\binom{n}{m} = \frac{n}{m}\binom{n-1}{m-1}.$$

Công thức này được gọi là công thức "di chuyển các phần tử ra khỏi ngoặc".

5. Công thức nhị thức.

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

Chính vì công thức này nên số k–tổ hợp được gọi là các hệ số nhị thức.

Chú ý rằng công thức này cho ta

$$0^0 = 1.$$

Trong trường hợp riêng khi $y = 1$, ta có

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

Và khi $x = 1$, ta có

$$2^n = \sum_{k=0}^{n} \binom{n}{k}.$$

5. Công thức nhị thức.

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

Chính vì công thức này nên số k–tổ hợp được gọi là các hệ số nhị thức.

Chú ý rằng công thức này cho ta

$$0^0 = 1.$$

Trong trường hợp riêng khi $y = 1$, ta có

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

Và khi $x = 1$, ta có

$$2^n = \sum_{k=0}^{n} \binom{n}{k}.$$

6. Tổng của tích.

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

Công thức này được chứng minh bằng lập luận đơn giản như sau.

- Ta có n quả cam và m quả táo và muốn lấy k quả, với $0 \leq k \leq n+m$.
- Trong mỗi lần lấy, ta có thể lấy s quả cam và $k-s$ quả táo. Có $\binom{n}{s}\binom{m}{k-s}$ cách lấy k quả trong đó có s quả cam.
- Giá trị của s có thể biến đổi từ 0 tới n nên ta có công thức tổng của tích nêu trên.

6. Tổng của tích.

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

Công thức này được chứng minh bằng lập luận đơn giản như sau.

- Ta có n quả cam và m quả táo và muốn lấy k quả, với $0 \leq k \leq n+m$.
- Trong mỗi lần lấy, ta có thể lấy s quả cam và $k-s$ quả táo. Có $\binom{n}{s}\binom{m}{k-s}$ cách lấy k quả trong đó có s quả cam.
- Giá trị của s có thể biến đổi từ 0 tới n nên ta có công thức tổng của tích nêu trên.

Nội dung

Sinh các k–tổ hợp

Giả sử tập n phần tử được kí hiệu là $\{1, 2, \ldots, n\}$.

Bài toán

Hãy sinh mọi tập con k phần tử của một tập có n phần tử.

Sinh các k–tổ hợp theo thứ tự từ điển

Ví dụ với $k = 3$ và $n = 5$, ta có 10 tập con gồm 3 phần tử của tập $\{0, 1, 2, 3, 4\}$ theo thứ tự như sau:

Thứ tự	**Tập**	$\vec{a}$
0	{0, 1, 2}	[0, 1, 2]
1	{0, 1, 3}	[0, 1, 3]
2	{0, 1, 4}	[0, 1, 4]
3	{0, 2, 3}	[0, 2, 3]
4	{0, 2, 4}	[0, 2, 4]
5	{0, 3, 4}	[0, 3, 4]
6	{1, 2, 3}	[1, 2, 3]
7	{1, 2, 4}	[1, 2, 4]
8	{1, 3, 4}	[1, 3, 4]
9	{2, 3, 4}	[2, 3, 4]

Sinh các k–tổ hợp theo thứ tự từ điển

- *Ý tưởng*: Sinh tập con tiếp theo có thứ tự đứng ngay sau tập con hiện tại.
- Giả sử tập con hiện tại là $\{\dots, \underline{5}, 8, 9, 10\}$, khi đó tập con ngay sau là $\{\dots, \underline{6}, 7, 8, 9\}$.
- Tổng quát, nếu tập con hiện tại là $\{a_1, \dots, a_{i-1}, \underline{a_i}, \dots, a_k\}$ thì tập con tiếp theo là

$$\{a_1, \dots, a_{i-1}, \underline{a_i + 1}, a_i + 2, \dots, a_i + k - i + 1\}.$$

Sinh các k–tổ hợp theo thứ tự từ điển

- Để tìm vị trí i (vị trí bắt đầu cập nhật giá trị), ta xuất phát từ vị trí cuối ($i = k - 1$), giảm i nếu $a_i = n - k + i$.
- Sau khi tìm được vị trí i thì ta có thể cập nhật các giá trị a_j như sau:

$$
\begin{aligned}
a_i &= a_i + 1 \\
a_j &= a_{j-1} + 1, \quad \forall j = i + 1, \dots, k - 1.
\end{aligned}
$$

- Từ đó, ta có thể liệt kê mọi k-tổ hợp của n phần tử bằng thuật toán lặp sinh các tập theo thứ tự từ điển.

Sinh các k–tổ hợp theo thứ tự từ điển

```
void enumerate(int a[], int n, int k) {
  int i, j;
  for (i = 0; i < MAX; i++)
    a[i] = i;
  while (a[k - 1] < n) {
    printSubset(a, k);
    i = k - 1;
    while (i > 0 && a[i] == n - k + i)
      i--;
    a[i]++;
    for (j = i + 1; j < k; j++)
      a[j] = a[j - 1] + 1;
  }
}
```

Sinh các k–tổ hợp theo thứ tự từ điển

Ví dụ, thực hiện thuật toán với $n = 6$ và $k = 3$ ta có 20 tập con sau:

Thứ tự	**Tập**	**Thứ tự**	**Tập**
0	$\{0, 1, 2\}$	10	$\{1, 2, 3\}$
1	$\{0, 1, 3\}$	11	$\{1, 2, 4\}$
2	$\{0, 1, 4\}$	12	$\{1, 2, 5\}$
3	$\{0, 1, 5\}$	13	$\{1, 3, 4\}$
4	$\{0, 2, 3\}$	14	$\{1, 3, 5\}$
5	$\{0, 2, 4\}$	15	$\{1, 4, 5\}$
6	$\{0, 2, 5\}$	16	$\{2, 3, 4\}$
7	$\{0, 3, 4\}$	17	$\{2, 3, 5\}$
8	$\{0, 3, 5\}$	18	$\{2, 4, 5\}$
9	$\{0, 4, 5\}$	19	$\{3, 4, 5\}$

Bài tập

Bài tập 9. Viết chương trình tìm mọi tập con gồm k phần tử của một tập n phần tử bằng phương pháp từ điển. Chạy thử chương trình với các dữ liệu khác nhau và kiểm tra kết quả.

Nội dung

Một số thuật toán khác

Tham khảo thêm một số thuật toán sinh các tập con khác trong:

- D. E. Knuth, *The Art of Computer Programming: Introduction to Combinatorial Algorithms and Boolean Functions*, 1st ed. Addison Wesley Publishing, 2008, vol. 4.
- F. Ruskey, “Adjacent interchange generation of combinations,” *Journal of Algorithms*, vol. 9, pp. 162–180, 1988.
- T. Hough and F. Ruskey, “An effcient implementation of the Eades, Hickey, Read adjacent interchange combination generation algorithm,” *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 4, pp. 79–86, 1988.
- S. G. Akl, “A comparison of combination generation methods,” *ACM Trans. Math. Softw.*, vol. 7, no. 1, pp. 42–45, 1981.

Một số thuật toán khác

Tham khảo thêm một số thuật toán sinh các tập con khác trong:

- D. E. Knuth, *The Art of Computer Programming: Introduction to Combinatorial Algorithms and Boolean Functions*, 1st ed. Addison Wesley Publishing, 2008, vol. 4.
- F. Ruskey, "Adjacent interchange generation of combinations," *Journal of Algorithms*, vol. 9, pp. 162–180, 1988.
- T. Hough and F. Ruskey, "An effcient implementation of the Eades, Hickey, Read adjacent interchange combination generation algorithm," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 4, pp. 79–86, 1988.
- S. G. Akl, "A comparison of combination generation methods," *ACM Trans. Math. Softw.*, vol. 7, no. 1, pp. 42–45, 1981.

Một số thuật toán khác

Tham khảo thêm một số thuật toán sinh các tập con khác trong:

- D. E. Knuth, *The Art of Computer Programming: Introduction to Combinatorial Algorithms and Boolean Functions*, 1st ed. Addison Wesley Publishing, 2008, vol. 4.
- F. Ruskey, "Adjacent interchange generation of combinations," *Journal of Algorithms*, vol. 9, pp. 162–180, 1988.
- T. Hough and F. Ruskey, "An effcient implementation of the Eades, Hickey, Read adjacent interchange combination generation algorithm," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 4, pp. 79–86, 1988.
- S. G. Akl, "A comparison of combination generation methods," *ACM Trans. Math. Softw.*, vol. 7, no. 1, pp. 42–45, 1981.

Tham khảo thêm một số thuật toán sinh các tập con khác trong:

- D. E. Knuth, *The Art of Computer Programming: Introduction to Combinatorial Algorithms and Boolean Functions*, 1st ed. Addison Wesley Publishing, 2008, vol. 4.
- F. Ruskey, "Adjacent interchange generation of combinations," *Journal of Algorithms*, vol. 9, pp. 162–180, 1988.
- T. Hough and F. Ruskey, "An effcient implementation of the Eades, Hickey, Read adjacent interchange combination generation algorithm," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 4, pp. 79–86, 1988.
- S. G. Akl, "A comparison of combination generation methods," *ACM Trans. Math. Softw.*, vol. 7, no. 1, pp. 42–45, 1981.

Nội dung

Tóm lược

Các nội dung chính của bài giảng:

- Biểu diễn các tập con của một tập hợp
- Ba thuật toán sinh các tập con của một tập hợp:
  - Thuật toán cộng một
  - Thuật toán đệ quy
  - Thuật toán mã Gray (phương pháp đệ quy, phương pháp tính nhanh bằng `xor`, phương pháp đảo bít)
- Các tập con k phần tử:
  - Các hệ số nhị thức
  - Thuật toán sinh các tập con theo thứ tự từ điển
- Các bài tập lập trình để củng cố kiến thức