

Recursion, Backtracking, Branch and Bound (2)

Truong Ngoc Tuan

Nội dung

- ❑ Recursion
- ❑ Backtracking
- ❑ **Generating Method**
- ❑ Branch and Bound

Nội dung

□ Generating Method

- Lược đồ chung
- Bài toán chuỗi 3 ký tự
- Liệt kê tập con của tập N phần tử
- Bài toán tập con K phần tử
- Hóa vị tập N phần tử

Nội dung

□ Generating Method

- **Lược đồ chung**
- Bài toán chuỗi 3 ký tự
- Liệt kê tập con của tập N phần tử
- Bài toán tập con K phần tử
- Hóa vị tập N phần tử

Generating Method

□ Bài toán mở đầu

- Có 1 biến $x_1, x_2, x_3, \dots, x_n$
- Mỗi biến x_i có thể thuộc về 1 tập hợp P_i
- Miền của bài toán là tập tích

$$P_1 \times P_2 \times P_3 \times \dots \times P_n$$

- Phép gán (assignment): là một bộ giá trị $a_1, a_2, a_3, \dots, a_n$. Trong đó $a_1 \rightarrow a_i$ thuộc P_i
- Một lời giải của bài toán là 1 pháp gán trị
- Một phép gán trị được gọi là một cấu hình

Generating Method

□ Ví dụ 1

Có 3 nhân viên bảo vệ làm 3 ca sáng, chiều, tối. Trong 1 ca chỉ có 1 bảo vệ. Hỏi các cách bố trí các bảo vệ?

Mã hóa bài toán:

$\{x, y, z\}$ là tập biến có thứ tự mô tả cho 3 ca: sáng, chiều, tối theo thứ tự

Miền trị của 3 biến là $\{a, b, c\}$ mô tả cho 3 bảo vệ

Các phép gán

<u>x</u>	<u>y</u>	<u>z</u>
a	b	c
a	c	b
b	a	c
b	c	a
c	a	b
c	b	a

Generating Method

□ Ví dụ 2

Tìm số chuỗi có độ dài 3 ký tự xyz với

$$x \in \{a, b, c\}$$

$$y \in \{d, e\}$$

$$z \in \{m, n, t\}$$

Nhận xét:

3 biến có 3 miền giá trị khác nhau

<u>x</u>	<u>y</u>	<u>z</u>
a	d	m
a	d	n
a	d	t
a	e	m
a	e	n
a	e	t
<hr/>		
b	d	m
b	d	n
b	d	t
b	e	m
b	e	n
b	e	t
<hr/>		
c	d	m
c	d	n
c	d	t
c	e	m
c	e	n
c	e	t

Generating Method

□ Nhận xét

- Bài toán tổ hợp có độ phức tạp là $n!$ hoặc n^m
- Làm thế nào tạo ra các phép gán trị?
→ Phương pháp sinh

Generating Method

□ Lược đồ chung

- Phương pháp sinh: từ dữ liệu ban đầu, sinh ra dữ liệu tiếp theo cho đến khi kết thúc
- Dùng để giải quyết bài toán liệt kê của lý thuyết tổ hợp

Generating Method

□ Lược đồ chung

▪ Điều kiện của thuật toán sinh:

1. Có thể xác định 1 tập thứ tự các cấu hình của tổ hợp (thứ tự của các phép gán giá trị, thường dung thứ tự từ điển)
2. Có một cấu hình cuối (điều kiện kết thúc của giải thuật)
3. Có một cách để suy ra được cấu hình tiếp theo

Generating Method

□ Thứ tự từ điển

- $S1 = \text{"1234567"}$
- $S2 = \text{"1235567"}$
- $S1 < S2$ nếu có 1 vị trí i mà tại đó $S1[i] < S2[i]$
- Thứ tự từ điển ngược: ngược lại với thứ tự từ điển

Generating Method

```
void generate_method() {  
    c = initial_configure;  
    process(c);  
    if (c == last_configure)  
        stop = true;  
    else  
        stop = false  
    while(stop == false){  
        c = get_next_configure(c);  
        process(c);  
        if(c == last_configure)  
            stop = true;  
    }  
}
```

Nội dung

□ Generating Method

- Lược đồ chung
- Bài toán chuỗi 3 ký tự
- Liệt kê tập con của tập N phần tử
- Bài toán tập con K phần tử
- Hóa vị tập N phần tử

Generating Method

□ Bài toán

- Tìm số chuỗi có độ dài 3 ký tự xyz với

$$x \in \{a, b, c\}$$

$$y \in \{d, e\}$$

$$z \in \{m, n, t\}$$

Nội dung

□ Generating Method

- Lược đồ chung
- Bài toán chuỗi 3 ký tự
- **Liệt kê tập con của tập N phần tử**
- Bài toán tập con K phần tử
- Hóa vị tập N phần tử

Generating Method

□ Bài toán

- Mã hóa tập biến: Tập biến gồm n biến ký tự theo thứ tự các phần tử \rightarrow mảng n ký tự
- Miền trị của mỗi biến $\{ '0', '1' \}$. Với **'0'** mô tả cho tình huống phần tử này **không có** trong tập con. **'1'** mô tả cho tình huống phần tử này **có mặt** trong tập con

Generating Method

□ Bài toán

- Với tập cha là 4 phần tử $X = \{a, b, c, d\}$, có thể dung mảng “0111” mô tả cho tập con $\{b, c, d\}$
-> Mỗi tập con được biểu diễn là một xâu nhị phân
- Trạng thái khởi tạo “0000” mang ý nghĩa tập trống
- Trạng thái kết thúc “1111” mang ý nghĩa là tập cha

Generating Method

□ Ví dụ:

- Với tập cha gồm 4 phần tử, ta sẽ có 2^4 tập con

Generating Method

□ Cộng 1 đơn vị

000 0	000 1
00 0 1	00 1 0
0 0 11	0 1 00
0 111	1 000

- Gọi i là vị trí bit 0 đầu tiên từ bên phải
- Cho các bit 1 bên phải vị trí i thành 0
- Cho bit i mang giá trị 1

```
i = n-1;
while (i >= 0 && a[i] == '1')
    a[i--] = '0';
a[i] = '1'
```

Nội dung

□ Generating Method

- Lược đồ chung
- Bài toán chuỗi 3 ký tự
- Liệt kê tập con của tập N phần tử
- Bài toán tập con K phần tử
- Hóa vị tập N phần tử

Generating Method

□ Bài toán

- Liệt kê các tập con K phần tử của tập N phần tử.
- Ví dụ: các tập con 3 phần tử của tập $\{1, 2, 3, 4, 5\}$ là:
 $\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\},$
 $\{1, 3, 4\}, \{1, 3, 5\}, \{1, 4, 5\},$
 $\{2, 3, 4\}, \{2, 3, 5\}, \{2, 4, 5\},$
 $\{3, 4, 5\}$



Tập hợp N
chập K

Generating Method

□ Bài toán

- Ánh xạ tập hợp bất kỳ N phần tử vào tập $X = \{1, 2, \dots, n\}$
- Một tập con K phần tử của X là một bộ có thứ tự $a_1, a_2, a_3, \dots, a_k$ với:
 - $1 \leq a_1 < a_2 < a_3 \dots < a_k \leq a_n$

Generating Method

□ Ý tưởng

- Tập con đầu: $\{1, 2, 3, \dots, k\}$

Ví dụ: $\{1, 2, 3\}$ với $K = 3, N = 5$

- Tập con cuối: $\{n-k+1, n-k+2, \dots, n\}$

Ví dụ: $\{3, 4, 5\}$ với $K = 3, N = 5$

Generating Method

□ Sinh các tập con kế tiếp từ tập con đã có

1. Tìm vị trí đầu tiên từ bên phải 1 vị trí i sao cho

$a[i] \neq n-k+1$

$i=k$

while ($a[i] == n-k+i$)

$i--;$

Generating Method

□ Sinh các tập con kế tiếp từ tập con đã có

1.

2. Thay $a[i]$ bằng $a[i] + 1$

$a[i] = a[i] + 1;$

3. Thay các vị trí sau i ($a[j]$) bằng các giá trị $a[i] + j - i, \dots$

$\text{for}(j=i+1; j \leq k; j++)$

$a[j] = a[i] + j - i;$

$n=8, k=6$

$\{ 1, 2, 5, 6, 7, 8 \}$
 $(i=6, n-k+i=8)$

Tìm vị trí đầu tiên khác với nhóm trị ở cuối tập cha theo thứ tự

$\{ 1, 3, 5, 6, 7, 8 \}$

$\{ 1, 3, 4, 5, 6, 7 \}$

Nội dung

□ Generating Method

- Lược đồ chung
- Bài toán chuỗi 3 ký tự
- Liệt kê tập con của tập N phần tử
- Bài toán tập con K phần tử
- Hóa vị tập N phần tử

Generating Method

□ Bài toán

- Cho tập $X = \{1, 2, 3, \dots, n\}$. Hãy liệt kê tất cả các hoán vị của tập này
- Một hoán vị của X là một bộ $A = (a_1, a_2, \dots, a_n)$ với $a_i \neq a_j$ nếu $i \neq j$
- $A = (a_1, a_2, \dots, a_k, \dots, a_n)$ là hoán vị trước của $A' = (a'_1, a'_2, \dots, a'_k, \dots, a'_n)$ nếu tìm được vị trí k sao cho $a_k < a'_k$

Generating Method

□ Bài toán

- Ví dụ:

1234**5**67 là hoán vị trước của

1234**6**57

- Đây là thứ tự từ điển
- Độ phức tạp $n!$

Generating Method

□ Ý tưởng

- Trạng thái trước {1, 3, 4, 2}, trạng thái sau {1, 4, 2, 3}

- Giải thuật

1. Tìm chỉ số lớn nhất j mà $a_j < a_{j+1}$ từ phía phải vì đây là phần tử sẽ bị hoán vị

1 **3** 4 2 ($j = 2$)

$j = n-1;$

while ($a[j] > a[j+1]$)

$j--;$

Generating Method

□ Ý tưởng

- Giải thuật

2. Tìm vị trí đầu tiên k đi ngược từ cuối tập trị với $a[k] > a[j]$

1 3 4 2 ($k = 3$)

```
k = n;
```

```
while (a[j] > a[k])
```

```
    k--;
```

Generating Method

□ Ý tưởng

- Giải thuật

3. Hoán vị $a[j]$ với $a[k]$

1 4 3 2

4. Lật ngược đoạn a_{i+1}, \dots, a_n

1 4 2 3 -> trạng thái kế tiếp

Generating Method

- Bài tập

- SPOJ



