



Thuật toán sinh kế tiếp

- Bài toán liệt kê
- Phương pháp sinh
- Một số ví dụ



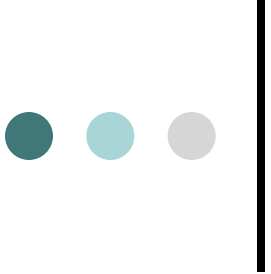
Bài toán liệt kê

- Lớp bài toán phổ biến trong tin học
 - Liệt kê các tập con của 1 tập hợp
 - Liệt kê các hoán vị ..
- Bài toán liệt kê đơn giản (khi số liệu ít hoặc xác định):
 - Thủ công
 - Dùng vòng lặp (trong lập trình)
- Phương pháp sinh giúp giải quyết các bài toán liệt kê tổ hợp phức tạp hơn



Điều kiện thực hiện

- Xác định được thứ tự trên tập các cấu hình liệt kê
 - Thứ tự từ điển
- Xác định được cấu hình đầu và cấu hình cuối
- Xây dựng được thuật toán, từ 1 cấu hình chưa phải cấu hình cuối, sinh ra cấu hình kế tiếp nó
 - Liệt kê tất cả: Liên tục sinh cho đến cấu hình cuối



Thứ tự từ điển

- So sánh kiểu số: Sử dụng giá trị
- So sánh ký tự: Sử dụng thứ tự từ điển
 - Thứ tự trong bảng chữ cái
- So sánh xâu ký tự hoặc dãy số: So sánh từng phần tử tính từ đầu xâu/dãy
 - Dãy $a[1..n] < b[1..n]$:
 $a[1] = b[1]$
...
 $a[k] = b[k]$
 $a[k+1] < b[k+1]$
 - Nếu độ dài khác nhau, bổ sung phần tử rỗng cho = nhau



Các bước thực hiện

1. Tạo cấu hình đầu tiên (thứ tự từ điển nhỏ nhất)
2. Thực hiện vòng lặp
 - 2.1 Đưa ra cấu hình hiện tại
 - 2.2 Kiểm tra cấu hình hiện tại có phải cấu hình cuối không.
Nếu là cấu hình cuối thoát khỏi vòng lặp.
 - 2.3 Từ cấu hình hiện tại, sinh ra cấu hình kế tiếp
 - 2.4 Gán cấu hình hiện tại = cấu hình mới sinh
3. Kết thúc



Mã giả thuật toán sinh

<Xây dựng cấu hình đầu tiên>

while True:

 <In ra cấu hình đang có>

 if <đây là cấu hình cuối cùng>:

 break

 <sinh cấu hình tiếp theo>



VD 1: Xâu nhị phân kế tiếp

1. Dãy nhị phân độ dài n : x_1, x_2, \dots, x_n (x_i thuộc $[0, 1]$)
2. Mỗi dãy là 1 số nguyên thuộc $[0, 2^n - 1]$
3. Cấu hình đầu tiên: 00...0
4. Cấu hình cuối cùng: 11...1
5. Quy tắc sinh cấu hình kế tiếp:
 1. Dãy kế tiếp dãy x_1, x_2, \dots, x_n có được bằng cách cộng 1 (cơ số 2) vào dãy hoặc số thập phân tiếp theo.
 2. Thuật toán sinh: Xét từ cuối dãy, gặp số 0 đầu tiên thì thay = 1 và cho tất cả các số phía sau = 0



VD 2: Tập con k phần tử

1. Tập n phần tử $(1, 2, \dots, n)$.
2. Tập con k phần tử bất kỳ lấy từ tập trên.
3. Cấu hình đầu tiên: $1, 2, \dots, k$
4. Cấu hình cuối cùng: $n-k+1, \dots, n$
5. Quy tắc sinh cấu hình kế tiếp:
 1. Tập con $x_1 < x_2 < \dots < x_k$: Giới hạn trên của x_k là n , của x_i là $n-k+i$. Giới hạn dưới của x_i là $x_{i-1} + 1$.
 2. Thuật toán sinh: Tìm từ cuối dãy, gặp 1 phần tử chưa gặp giới hạn trên \rightarrow tăng phần tử đó lên 1 và cho các phần tử phía sau = giới hạn dưới.



VD 3: Liệt kê hoán vị

1. Liệt kê các hoán vị của $(1, 2, \dots, n)$.
2. Cấu hình đầu tiên: $1, 2, \dots, n$
3. Cấu hình cuối cùng: $n, n-1, \dots, 1$
4. Quy tắc sinh cấu hình kế tiếp:
 1. Tìm đoạn cuối giảm dần dài nhất (duyệt từ cuối dãy gặp $a[j]$ đầu tiên $< a[j+1]$).
 2. Tìm $a[k]$ nhỏ nhất trong dãy đó mà $> a[j]$.
 3. Đổi chỗ $a[j], a[k]$.
 4. Đảo ngược đoạn $a[j+1], a[n]$



Bài tập

Viết chương trình bằng ngôn ngữ C/++ minh họa các giải thuật trên:

1. Liệt kê các xâu nhị phân kích thước n (n nhập từ bàn phím)
2. Liệt kê các tập con k phần tử của tập n phần tử (n , k nhập từ bàn phím, $n > k$)
3. Liệt kê các hoán vị của tập $1, 2, \dots, n$ (n nhập từ bàn phím)