



BÀI 2.2: THUẬT TOÁN SINH KẾ TIẾP

1. Bài toán duyệt
2. Thuật toán sinh kế tiếp
 1. Duyệt sâu nhị phân
 2. Duyệt tổ hợp
 3. Duyệt hoán vị
 4. Phân tích số



THUẬT TOÁN DUYỆT

- Bài toán Duyệt (vét cạn):
 - Mục tiêu: xem xét tất cả các khả năng có thể.
 - Áp dụng: Thường đưa về dạng bài toán tổ hợp
 - Ý tưởng: Sinh kế tiếp hoặc Quay lui
- Một thuật toán duyệt cần thỏa mãn hai điều kiện:
 - *Không được lặp lại bất kỳ khả năng nào.*
 - *Không được bỏ sót bất kỳ cấu hình nào*



THUẬT TOÁN SINH KẾ TIẾP

Thuật toán sinh được dùng để giải lớp các bài toán thỏa mãn hai điều kiện:

- **Xác định được một thứ tự trên tập các cấu hình cần liệt kê của bài toán. Biết được cấu hình đầu tiên, biết được cấu hình cuối cùng.**
- **Từ một cấu hình chưa phải cuối cùng, ta xây dựng được thuật toán sinh ra cấu hình đứng ngay sau nó theo thứ tự.**

Thuật toán:

Bước1 (Khởi tạo):

<Thiết lập cấu hình đầu tiên>;

Bước 2 (Bước lặp):

while (*<Lặp khi cấu hình chưa phải cuối cùng>*) **do**

<Đưa ra cấu hình hiện tại>;

<Sinh ra cấu hình kế tiếp>;

endwhile;

End.

BÀI TOÁN 1: DUYỆT CÁC XÂU NHỊ PHÂN ĐỘ DÀI N

Xâu $X = (x_1, x_2, \dots, x_n)$: $x_i = 0, 1$; $i = 1, 2, \dots, n$ được gọi là xâu nhị phân có độ dài n . Ví dụ với $n=4$, ta có 16 xâu nhị phân dưới đây:

<i>STT</i>	$X = (x_1, \dots, x_n)$	$F(X)$	<i>STT</i>	$X = (x_1, \dots, x_n)$	$F(X)$
1	0000	0	9	1000	8
2	0001	1	10	1001	9
3	0010	2	11	1010	10
4	0011	3	12	1011	11
5	0100	4	13	1100	12
6	0101	5	14	1101	13
7	0110	6	15	1110	14
8	0111	7	16	1111	15

THUẬT TOÁN SINH XÂU NHỊ PHÂN KẾ TIẾP

Input:

- + $X = (x_1, x_2, \dots, x_n)$ là biến toàn cục.
- + $OK = 1$ là biến toàn cục

Output: 2^n xâu nhị phân có độ dài n .

```
void Next_Bit_String(void) {  
    int i=n;  
    while (i>0 && X[i]!=0) { X[i] = 0; i--; }  
    if (i >0) X[i]=1;  
    else OK = 0;  
}
```

Actions:

```
X =(0,0,...,0); // Xâu nhị phân ban đầu.  
while (OK) { //Lặp khi xâu chưa phải cuối cùng  
    Result(); //Đưa ra xâu hiện tại>  
    Next_Bit_String(); //Sinh ra xâu kế tiếp  
}
```

Endactions

BÀI TOÁN 2: DUYỆT CÁC TỔ HỢP CHẬP K CỦA 1,2...N

Mỗi tổ hợp chập K của 1, 2, ..., N là một tập con K phần tử khác nhau của 1, 2,..., N. Ví dụ với N=5, K= 3 ta sẽ có $C(N,K)$ tập con dưới đây

STT	Tập con $X=(x_1,...,x_k)$
1	1 2 3
2	1 2 4
3	1 2 5
4	1 3 4
5	1 3 5
6	1 4 5
7	2 3 4
8	2 3 5
9	2 4 5
10	3 4 5

THUẬT TOÁN SINH KẾ TIẾP TỔ HỢP

```
void Next_Combination(void) {  
    int i = k; // Xuất phát từ phần tử cuối cùng của tổ hợp  
    while ( i>0 && X[i] == N - K + i) i --; //Tìm phần tử  $X[i] \neq N-K+i$   
    if (i>0) { //Nếu i chưa vượt quá phần tử cuối cùng  
        X[i] = X[i] + 1; //Thay  $X[i] = X[i] + 1$   
        for (int j = i+1; j<=k; j++) //Từ phần tử thứ j +1 đến k  
            X[j] = X[i] + j - i; // Thay thế  $X[j] = X[i] + j - i$   
    }  
    else OK = 0; //OK =0 nếu đã đến tập con cuối cùng  
}
```


BÀI TOÁN 3: DUYỆT CÁC HOÁN VỊ CỦA 1,2..N

Mỗi hoán vị của 1, 2, .., N là một cách xếp có tính đến thứ tự của 1, 2,...,N. Số các hoán vị là $N!$

Thứ tự tự nhiên. Thứ tự đơn giản nhất có thể được xác định như sau. Hoán vị $X = (x_1, x_2, \dots, x_n)$ được gọi là đứng trước hoán vị $Y = (y_1, y_2, \dots, y_n)$ nếu tồn tại chỉ số k sao cho $x_1 = y_1, x_2 = y_2, \dots, x_{k-1} = y_{k-1}, x_k < y_k$.

Ví dụ hoán vị $X = (1, 2, 3)$ được gọi là đứng trước hoán vị $Y = (1, 3, 2)$ vì tồn tại $k = 2$ để $x_1 = y_1$, và $x_2 < y_2$.

STT	Hoán vị $X = (x_1, \dots, x_N)$
1	1 2 3
2	1 3 2
3	2 1 3
4	2 3 1
5	3 1 2
6	3 2 1

```

void Next_Permutation(void) {
    int j = N-1; // Xuất phát từ phần tử N-1
    while ( j>0 && X[j]> X[j+1]) j --; //Tìm j sao cho X[j]>X[j+1]
    if (j>0) { //Nếu i chưa vượt quá phần tử cuối cùng
        int k =N; // Xuất phát từ k = N
        while ( X[j] > X[k] ) k --; // Tìm k sao cho X[j] <X[k]
        int t = X[j]; X[j] = X[k]; X[k] = t; //Đổi chỗ X[j] cho X[k]
        int r = j +1, s = N;
        while ( r <=s ) { //Lật ngược đoạn từ j +1 đến N
            t = r; r = s; s=t;
            r ++; s --;
        }
    }
    else OK =0; //Nếu đến hoán vị cuối cùng
}

```

BÀI TOÁN 4: PHÂN TÍCH SỐ

Cho số tự nhiên N ($N \leq 100$). Hãy liệt kê tất cả các cách chia số tự nhiên N thành tổng của các số tự nhiên nhỏ hơn N . Các cách chia là hoán vị của nhau chỉ được tính là một cách.

Ví dụ với $N = 5$ ta có 7 cách chia như sau:

5				
4	1			
3	2			
3	1	1		
2	2	1		
2	1	1	1	
1	1	1	1	1

THUẬT TOÁN SINH KẾ TIẾP - PHÂN TÍCH SỐ

X[]: Dãy phân tích hiện tại

k: độ dài dãy phân tích hiện tại

```
void Next_Division() {  
    int i = k, j, R, S, D;  
    while (i > 0 && X[i]==1) i--;  
    if (i>0) {  
        X[i] = X[i] - 1;    D = k - i + 1;  
        R = D / X[i];    S = D % X[i];  
        k = i;  
        if (R>0) {  
            for (j = i + 1; j <= i + R; j++)  
                X[j] = X[i];  
            k = k + R;  
        }  
        if (S>0) {  
            k = k + 1; X[k] = S;  
        }  
    }  
    else OK = 0;  
}
```