Một số bài toán khác

Bởi:

Khoa CNTT ĐHSP KT Hưng Yên

Bài toán liệt kê các dãy nhị phân độ dài n

Bài toán:

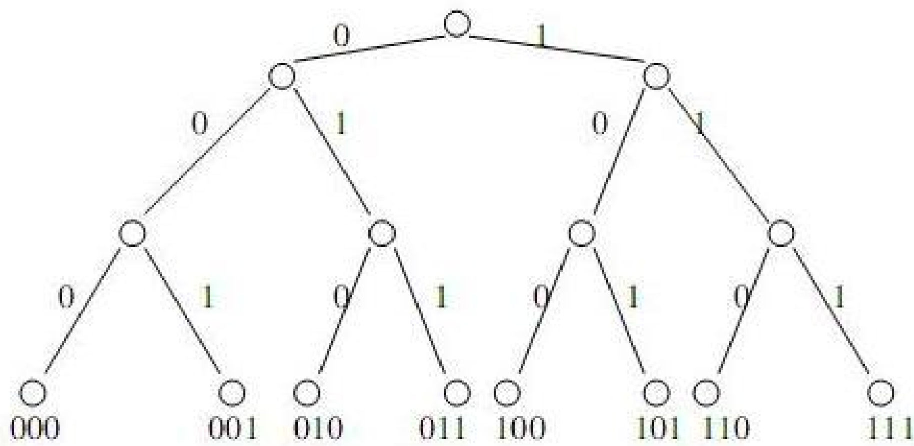
Liệt kê các dãy có chiều dài n dưới dạng $x_1x_2...x_n$, trong đó xi €{ 0,1 }.

Phân tích, thiết kế thuật toán:

Ta có thể sử dụng sơ đồ tìm tất cả các lời giải của bài toán. Hàm Try(i) xác định xi, trong đó xi chỉ có 1 trong 2 giá trị là 0 hay 1. Các giá trị này mặc nhiên được chấp nhận mà không cần phải thoả mãn điều kiện gì. Nên Hàm try(i) có thể viết như sau :

```
Try ( i) ≡
{
for (j = 0; j <= 1; j++)
{
x[i] = j;
if (i < n )
Try (i+1);
else
Xuất(x);
}
}
```

Cây không gian các trạng thái của bài toán có thể mô tả bởi



Bài toán liệt kê các hoán vị

Bài toán:

Liệt kê hoán vị của n số nguyên dương đầu tiên

Phân tích, thiết kế thuật toán:

Ta biểu diễn các hoán vị dưới dạng $a_1 \ldots a_n$; a_i €{1,..,n} và $a_i \neq a_j$ nếu $i \neq j$. Với mọi i, a_i chấp nhận giá trị j nếu j chưa được sử dụng, và vì vậy ta cần ghi nhớ j đã được sử dụng hay chưa khi quay lui. Để làm điều này ta dùng một dãy các biến logic b_j với quy ước :

$$\forall j = \overline{1,n} : b_j = \begin{cases} 1; \textit{nếu } j \textit{ chưa sử dụng} \\ 0; \textit{nếu ngược lại.} \end{cases}$$

Sau khi gán j cho ai , ta cần ghi nhớ cho bj (bj = 0) và phải trả lại trạng thái cũ cho bj (bj = True) khi thực hiện việc in xong một hoán vị. Ta chú ý rằng dãy các biến bj sẽ được khởi động bằng 1

Thuật toán có thể viết như sau :

Try(i)

{

for (j = 1; j <= n; j++)

```
if ( b[j])

{

a[i] = j;

b[j] = 0; // Ghi nhận trạng thái mới

if (i < n)

Try(i+1);

else

Xuất();

b[j] = True; // Trả lại trạng thái cũ

}

}
```

Bài toán liệt kế tổ hợp

Bài toán:

Liệt kê các tổ hợp chập k của n phần tử

Phân tích, thiết kế thuật toán:

Ta sẽ biểu diễn tổ hợp dưới dạng x1...xk ; Trong đó :

$1<= x_1, x_2, \ldots, x_n <= n$

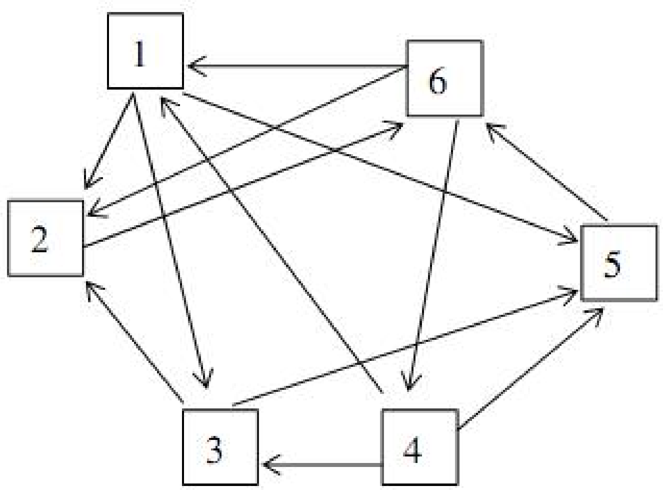
Ta nhận xét rằng với mọi j €{1,..n}: x_i chấp nhận j ≡ j €{ c_{i-1}+1, . ., n-k+i}.Các giá trị j thỏa điều kiện trên mặc nhiên được chấp nhận, nên ta không cần dùng các biến booole để ghi nhớ nữa.

Thuật toán có thể viết như sau :

```
Try( i)

for ( j = 1; j <= n ; j++)
```

```
if( x[i-1] + 1 <= j <= n - k + i )
{
x[i] = j;
if (i < k) Try(i+1);
else
Xuất(x);
}
```



Tìm đường đi từ đỉnh (1) đến đỉnh (4) : A(0) = {1};

A(1) = {2,3,5} A(2) = {6} A(3) = {4}

Đường đi ngắn nhất tìm được là 4 ? 6 ? 5 ? 1 , có chiều dài là 3.