Học viện Công nghệ Bưu chính Viễn thông
Khoa Công nghệ thông tin 1

Toán rời rạc 1

Bài toán liệt kê

Nội dung

- Giới thiệu bài toán
- Phương pháp sinh
- Phương pháp quay lui
- Bài tập

Giới thiệu bài toán liệt kê

- **Bài toán đếm:** Xây dựng công thức tính nghiệm của bài toán
- **Bài toán liệt kê:** Nghiệm của bài toán là gì?
- **Phương pháp chung để giải quyết bài toán liệt kê:** Sử dụng thuật toán vét cạn xem xét tất cả các khả năng xảy ra của các cấu hình tổ hợp để từ đó đưa ra từng nghiệm của bài toán
- Phương pháp liệt kê cần thỏa mãn 2 điều kiện:
  - Không được lặp lại bất kỳ cấu hình nào
  - Không được bỏ sót bất kỳ cấu hình nào
- Các bước tiến hành giải bài toán bằng máy tính:
  - Hiểu yêu cầu của bài toán
  - Chọn cấu trúc dữ liệu biểu diễn phương án cần duyệt
  - Chọn thuật toán phù hợp với cấu trúc dữ liệu
  - Cài đặt thuật toán & thử nghiệm chương trình
  - Tối ưu chương trình

Giới thiệu bài toán liệt kê

Ví dụ 1: Cho tập hợp các số $a_1, a_2, .., a_n$ và số M. Hãy tìm tất cả các tập con k phần tử của dãy số $\{a_n\}$ sao cho tổng số các phần tử trong tập con đó đúng bằng M.

Ví dụ 2: Một thương nhân đi bán hàng tại tám thành phố. Chị ta có thể bắt đầu hành trình của mình tại một thành phố nào đó nhưng phải qua 7 thành phố kia theo bất kỳ thứ tự nào mà chị ta muốn. Hãy chỉ ra lộ trình ngắn nhất mà chị ta có thể đi.

Nội dung

- Giới thiệu bài toán
- Phương pháp sinh
- Phương pháp quay lui
- Bài tập

Thuật toán sinh (1/2)

- 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, ta xây dựng được thuật toán sinh ra cấu hình đứng ngay sau nó theo thứ tự.

Thuật toán sinh (2/2)

Bước 1 (Khởi tạo):

<Thiết lập cấu hình đầu tiên>;

Bước 2 (Bước lặp):

```
while (<Lặp khi cấu hình chưa phái cuối cùng>)
{
        <Đưa ra cấu hình hiện tại>;
        <Sinh ra cấu hình kế tiếp>;
}
```

<Đưa ra cấu hình cuối cùng>;

Ví dụ 2

- **Bài toán**: Liệt kê (duyệt) các xâu nhị phân có độ dài n.

Ví dụ 2

- **Bài toán**: Liệt kê (duyệt) các xâu nhị phân có độ dài n.

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

- Ta nhận thấy cấu hình đầu tiên sẽ là (00…0) và cấu hình cuối cùng sẽ là (11…1).

Nhận xét rằng nếu cấu hình x[1..n] là cấu hình đang có và không phải cấu hình cuối cùng cần liệt kê thì dãy kế tiếp sẽ nhận được bằng cách cộng thêm 1 (theo cơ số 2 có nhớ) vào cấu hình hiện tại.

Ví dụ 2

- **Bài toán**: Liệt kê (duyệt) các xâu nhị phân có độ dài n.

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

STT	$X = (x_1 \ldots x_n)$	*F(X)*	*STT*	$X = (x_1 \ldots x_n)$	*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

Ví dụ 2

- Thứ tự trên tập cấu hình được sắp theo giá trị của số mà cấu hình (xâu nhị phân) biểu diễn
- Cấu hình đầu tiên là xâu gồm n chữ số 0
- Cấu hình cuối cùng là xâu gồm n chữ số 1
- **Thuật toán sinh xâu nhị phân kế tiếp**
  - Giả sử cấu hình hiện tại $x = x_1x_2 \ldots x_n$
  - Nếu $x_i = 1$ với mọi i, thì x là cấu hình cuối cùng, thuật toán liệt kê kết thúc
  - Gọi x_k là chữ số 0 đầu tiên tính từ bên phải của x, như vậy

$$x = x_1x_2 \ldots x_{k-1}011 \ldots 1$$

  - Cấu hình tiếp theo $y = y_1y_2 \ldots y_n$ được tạo ra như sau
  - $y_i = x_i$ với $1 \le i \le k-1$, $y_i = 1 - x_i$ với $k \le i \le n$
  - $y = x_1x_2 \ldots x_{k-1}100 \ldots 0$

$$\boxed{y = x + 1}$$

Ví dụ 3

- Liệt kê (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 $X = \{x_1, \ldots, x_k\}$	STT	Tập con $X = \{x_1, \ldots, x_k\}$
1	1 2 3	6	1 4 5
2	1 2 4	7	2 3 4
3	1 2 5	8	2 3 5
4	1 3 4	9	2 4 5
5	1 3 5	10	3 4 5

Ví dụ 3

- **Thứ tự tự nhiên duyệt các tổ hợp chập k**

Có thể xác định được nhiều trật tự khác nhau trên các tổ hợp. Tuy nhiên, thứ tự đơn giản nhất có thể được xác định như sau:

Ta gọi tập con $X = (x_1, x_2, \ldots, xk)$ là đứng trước tập con $Y = (y_1, y_2, \ldots, yk)$ nếu tìm được chỉ số t sao cho:

$x_1 = y_1, x_2 = y_2, \ldots, x_{t-1} = y_{t-1}, x_t < y_t$.

Ví dụ tập con $X = (1, 2, 3)$ đứng trước tập con $Y = (1, 2, 4)$ vì với $t = 3$ thì $x_1 = y_1, x_2 = y_2, x_3 < y_3$.

Tập con (cấu hình) đầu tiên là $X = (1, 2, \ldots, k)$, tập con (cấu hình) cuối cùng là $(n - k + 1, \ldots, n)$. Như vậy điều kiện 1 của thuật toán sinh được thỏa mãn.

Thuật toán sinh cấu hình (tổ hợp) tiếp theo

Giả sử cấu hình tại a= $(a_1, a_2, \ldots, a_k)$ là tập con hiện tại và chưa phải là cấu hình cuối cùng.
Tìm từ phải sang trái a= $(a_1, a_2, \ldots, a_k)$ phần tử:
$a_i \neq n - k + i$
Thay a_i $bởi$ $a_i + 1$,
Thay a_j bởi $a_i{+}j - i$, với j=i+1, i+2,....,k

Ví dụ minh họa

Ví dụ: Chẳng hạn với n=6, k=4. Giả sử ta đang có tập con (1,2,5,6), cần xây dựng tập con kế tiếp nó trong thứ tự từ điển.

11 . Cho tập A = { 1, 2, 3, 4, 5, 6, 7, 8, 9}. Sử dụng phương pháp sinh tổ hợp chập k của một tập hợp theo thứ tự từ điển, hãy tạo 4 tổ hợp chập 4 liền kề tiếp theo của tổ hợp 2,6,8,9.

14. Cho tập A = { 1, 2, 3, 4, 5, 6, 7, 8, 9}. Sử dụng phương pháp sinh tổ hợp chập k của một tập hợp theo thứ tự từ điển, hãy tạo 4 tổ hợp chập 4 liền kề tiếp theo của tổ hợp 1,5,6,8.

Bài tập

Bài tập 6. Cho dãy số $A = (a_1, a_2, \ldots, a_n)$ và số tự nhiên P. Hãy liệt kê tất cả các dãy con k phần tử của dãy số A sao cho tổng các phần tử của dãy con đó đúng bằng P.

Ví dụ. $A = (5, 10, 15, 20, 25, 30, 35)$, $n = 7$, $k = 3$, $P = 50$ ta sẽ có các dãy con sau :

(5, 10, 35),
(5, 20, 25),
(10, 15, 25),
…

Bài tập 7. Cho dãy số $A = (a1, a2, \ldots, an)$. Hãy liệt kê tất cả các dãy con k phần tử tăng dần tự nhiên của dãy số A.

Ví dụ. $A = (1, 3, 2, 4, 5)$, $n = 5$, $k = 3$ ta có các dãy con tăng dần tự nhiên như sau :

(1, 3, 4),
(1, 3, 5),
(1, 2, 4),
…

Ví dụ 4

- Liệt kê (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 $n!$. Ví dụ với $n = 3$ ta có 6 hoán vị dưới đây.

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

Ví dụ 4

- **Thứ tự tự nhiên duyệt hoán vị**

Có thể xác định được nhiều trật tự khác nhau trên các hoán vị. Tuy nhiên, thứ tự đơn giản nhất có thể được xác định như sau. Hoán vị $X = (x_1, x_2, \ldots, x_n)$ được gọi là đứng sau hoán vị $Y = (y_1, y_2, \ldots, y_n)$ nếu tồn tại chỉ số k sao cho

$x_1 = y_1, x_2 = y_2, \ldots, x_{k-1} = y_{k-1}, x_k < y_k$.

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

Cấu hình đầu tiên là $(1, 2, \ldots, n)$

Cấu hình cuối cùng là $(n, n-1, \ldots, 1)$

Ví dụ 4

- Thuật toán sinh cấu hình (hoán vị) tiếp theo
- Giả sử cấu hình hiện tại là $X = (x_1, x_2, \ldots, x_n)$
- Nếu $x_{i-1} > x_i$ với mọi i, thì X là cấu hình cuối cùng. Thuật toán sinh kết thúc.
- Gọi t là chỉ số lớn nhất (chỉ số đầu tiên từ bên phải) sao cho $x_{t-1} < x_t$.
- Cấu hình tiếp theo $Y = (y_1, y_2, \ldots, y_n)$ được sinh ra như sau
  o $y_i = x_i$ với $i \leq t - 2$
  - x_{t-1} gán bằng phần tử nhỏ nhất trong tập $x_t, \ldots, x_n$ và lớn hơn x_{t-1} (ký hiệu là a)
  - Lật ngược đoạn $\{x_t, \ldots, x_n\}$

Nội dung

- Giới thiệu bài toán
- Phương pháp sinh
- Phương pháp quay lui
- Bài tập

Thuật toán quay lui (1/2)

- Giả sử ta cần xác định bộ $X = (x_1, x_2, \ldots, x_n)$ thỏa mãn một số ràng buộc nào đó. Ứng với mỗi thành phần x_i ta có n_i khả năng cần lựa chọn.
- Ứng với mỗi khả năng j trong n_i dành cho thành phần x_i ta cần thực hiện:
  - Kiểm tra xem khả năng j có được chấp thuận cho thành phần x_i hay không?
    - Nếu khả năng j được chấp thuận thì nếu i là thành phần cuối cùng ($i = n$) ta ghi nhận nghiệm của bài toán. Nếu i chưa phải cuối cùng ta xác định thành phần thứ $i + 1$.
    - Nếu không có khả năng j nào được chấp thuận cho thành phần x_i thì ta quay lại bước trước đó ($i - 1$) để thử lại các khả năng khác.

Thuật toán quay lui (2/2)

```
Back_Track (int i ) {
    for ( j =<Khả năng 1>; j <=n_i; j++ ){
        if (<chấp thuận khả năng j>) {
            X[i] = <khả năng j>;
            if ( i ==n)
                Result();
            else
                Back_Track(i+1);
        }
    }
```

Ví dụ 5

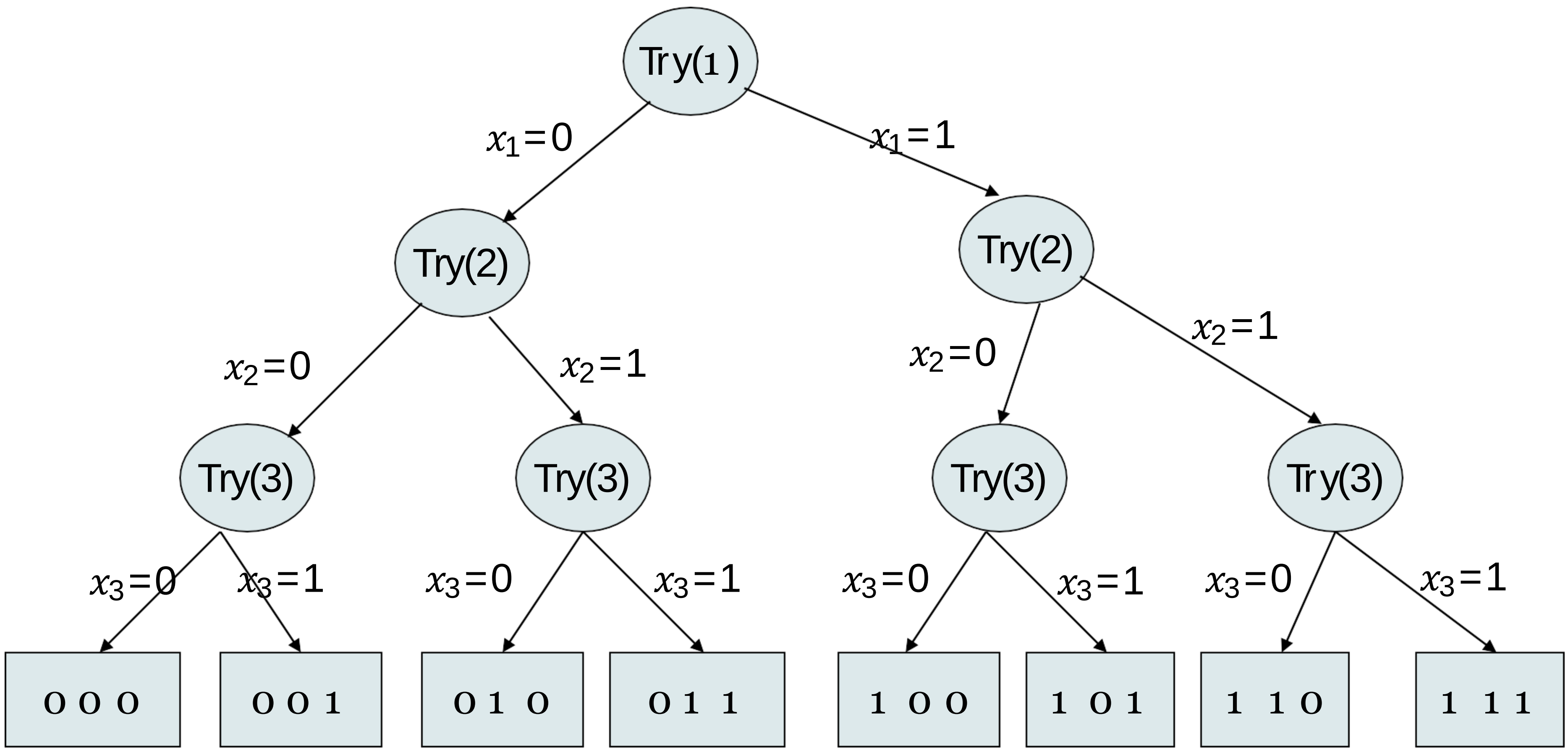
- Liệt kê (duyệt) các xâu nhị phân có độ dài n.

Xâu $X = (x_1x_2...x_n)$: $x_i = 0, 1$; $i = 1, 2, . . . , n$ được gọi là xâu nhị phân có độ dài n.

```
void Try (int i ){
        for (int j =0; j<=1; j++){
                X[i] = j;
                if (i ==n)
                        Result();
                else
                        Try (i+1);
        }
}
```

Khi đó, để duyệt các xâu nhị phân có độ dài n ta chỉ cần gọi đến thủ tục Try(1).

Ví dụ 5



Try(1)
$x_1=0$
$x_1=1$
Try(2)
Try(2)
$x_2=0$
$x_2=1$
$x_2=0$
$x_2=1$
Try(3)
Try(3)
Try(3)
Try(3)
$x_3=0$
$x_3=1$
$x_3=0$
$x_3=1$
$x_3=0$
$x_3=1$
$x_3=0$
$x_3=1$
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1

Ví dụ 6

- Liệt kê (duyệt) các tổ hợp chập k của 1, 2, … , n.

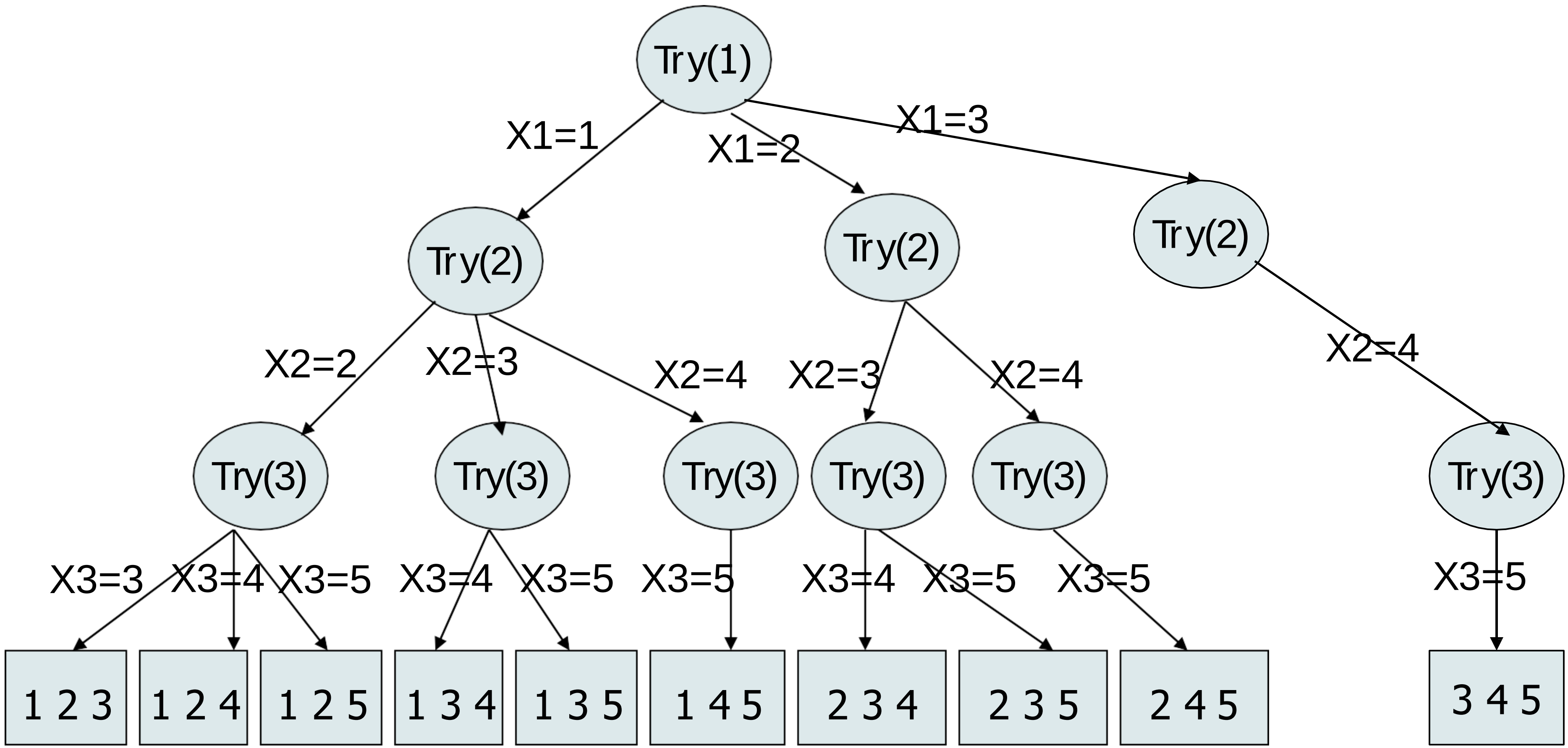
Mỗi tổ hợp chập k của 1, 2, … , n là một tập con k phần tử khác nhau của 1, 2, … , n.

```
void Try (int i ){
        for (int j =X[i-1]+1; j<=n-k+ i; j++){
            X[i] = j;
            if (i ==k)
                Result();
            else
                Try (i+1);
        }
}
```

Coi $X[0] = 0$.
Ta cần gán giá trị cho $X[1]$, … , $X[k]$

Khi đó, để duyệt các tập con k phần tử của 1, 2, … , n ta chỉ cần gọi đến thủ tục Try(1).

Ví dụ 6


Try(1)
X1=1
X1=2
X1=3
Try(2)
Try(2)
Try(2)
X2=2
X2=3
X2=4
X2=3
X2=4
X2=4
Try(3)
Try(3)
Try(3)
Try(3)
Try(3)
Try(3)
X3=3
X3=4
X3=5
X3=4
X3=5
X3=5
X3=4
X3=5
X3=5
X3=5
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

Ví dụ 7

- Sử dụng phương pháp quay lui liệt kê (duyệt) các hoán vị của 1,2, … , n.

Mỗi hoán vị $X = (x_1, x_2, \ldots, xn)$ là bộ có tính đến thứ tự của 1,2, … , n. Mỗi $x_i \in X$ có n lựa chọn. Khi $x_i = j$ được lựa chọn thì giá trị này sẽ không được chấp thuận cho các thành phần còn lại.

Để ghi nhận điều này, ta sử dụng mảng $unused[]$ gồm n phần tử.

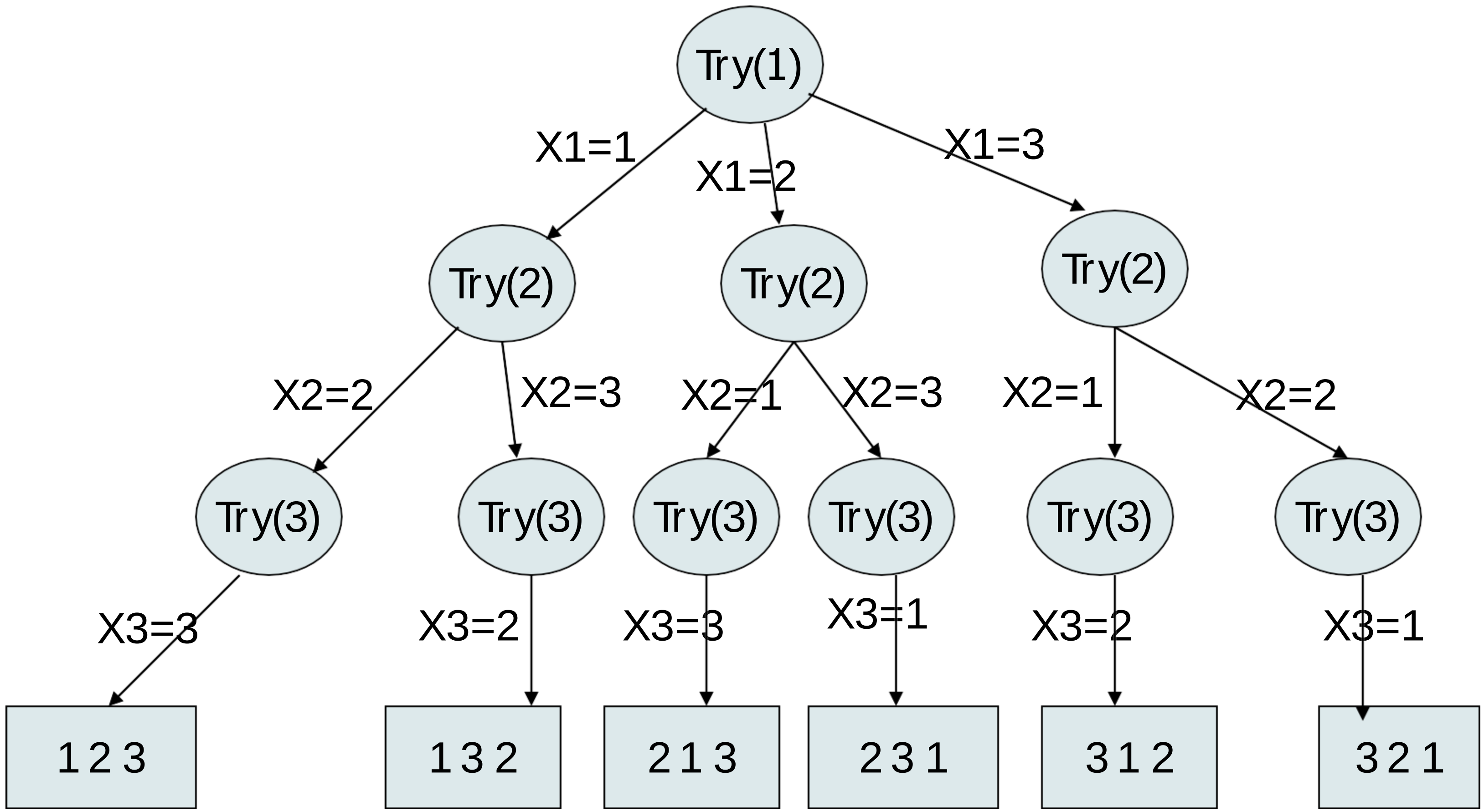
- Nếu $unused[i] = True$ điều đó có nghĩa giá trị i được chấp thuận
- $unused[i] = False$ tương ứng với giá trị i không được phép sử dụng

Ví dụ 7

```
void Try (int i ){
        for (int j =1; j<=n; j++){
                if (unused[j] ) {
                        X[i] =j;
                        unused[j] = false;
                        if (i ==n)
                                Result();
                        else
                                Try (i+1);
                        unused[j] = true;
                }
        }
}
```

Khi đó, để duyệt các hoán vị của 1, 2, … , n ta chỉ cần gọi đến thủ tục Try(1).

Ví dụ 7


Try(1)
X1=1
X1=2
X1=3
Try(2)
Try(2)
Try(2)
X2=2
X2=3
X2=1
X2=3
X2=1
X2=2
Try(3)
Try(3)
Try(3)
Try(3)
Try(3)
Try(3)
X3=3
X3=2
X3=3
X3=1
X3=2
X3=1
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1

Ví dụ 8

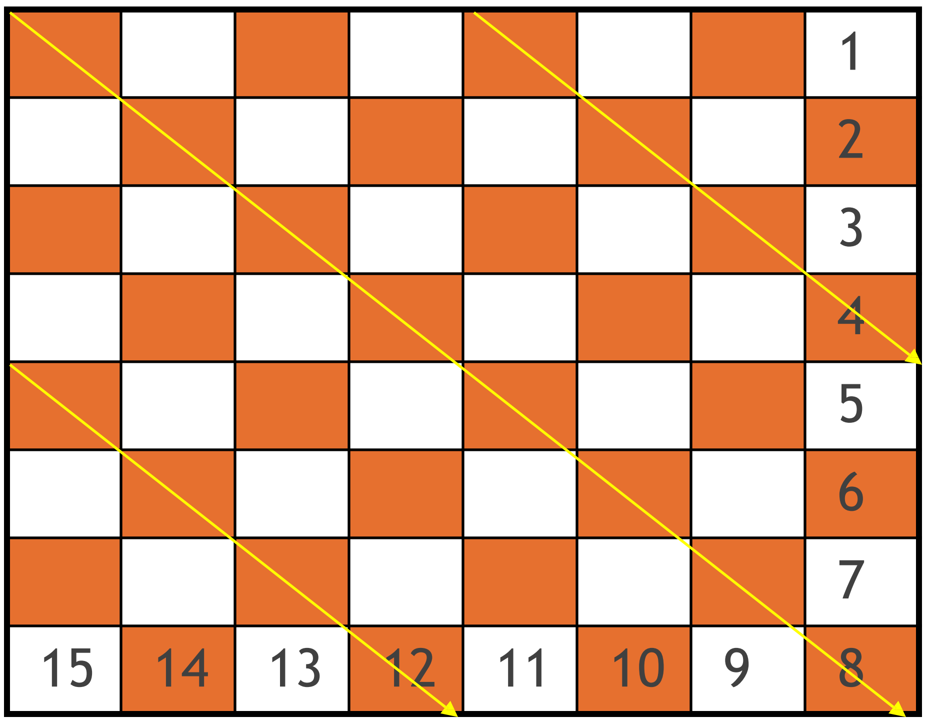
- Bài toán $\boldsymbol{n}$ quân hậu. Trên bàn cờ kích cỡ nxn, hãy đặt n quân hậu mỗi quân trên 1 hàng sao cho tất cả các quân hậu đều không ăn được lẫn nhau.

Gọi $X = (x_1, x_2, \ldots, xn)$ là một nghiệm của bài toán. Khi đó, $x_i = j$ được hiểu là quân hậu hàng thứ i đặt ở cột j. Để các quân hậu khác không thể ăn được, quân hậu thứ i cần không được lấy trùng với bất kỳ cột nào, không được cùng đường chéo xuôi, không được cùng trên đường chéo ngược.

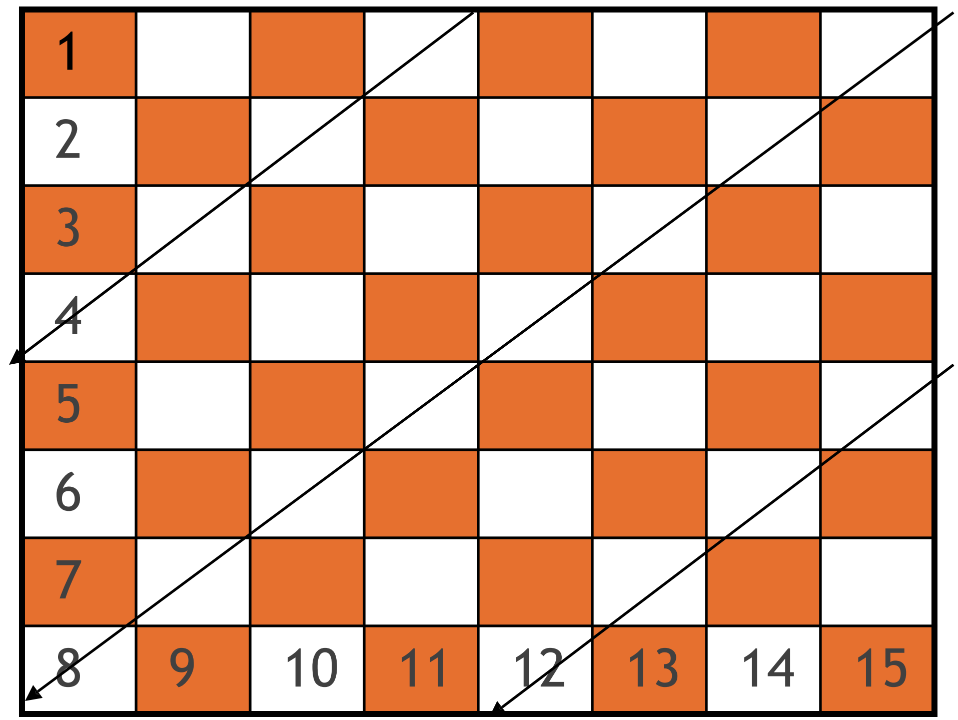
Ta có n cột $A = (a_1, \ldots, a_n)$, có $Xuoi[2 * n - 1]$ đường chéo xuôi, $Nguoc[2 * n - 1]$ đường chéo ngược.

Ví dụ 8

Đường chéo xuôi: $Xuoi\ [i-j+n]$



Đường chéo ngược: $Nguoc\ [i+j-1]$



Ví dụ 8

```
void Try (int  i){
        for(int j=1; j<=n; j++){
                if( A[j] && Xuoi[ i - j + n ]&& Nguoc[i + j -1]){
                        X[i] =j;
                        A[j]=false;
                        Xuoi[ i - j + n]=false;
                        Nguoc[ i + j - 1]=false;
                        if(i==n) Result();
                        else Try(i+1);
                        A[j] = true;
                        Xuoi[ i - j + n] = true;
                        Nguoc[ i + j - 1] = true;
                }
        }
}
```

Khi đó, để giải bài toán quân hậu ta chỉ cần gọi đến thủ tục Try(1).

Bài tập

I. Sử dụng thuật toán sinh, viết chương trình giải các bài tập dưới đây:

1.Liệt kê các xâu nhị phân có độ dài n.

2.Liệt kê các tập con k phần tử của 1, 2, … , n.

3.Liệt kê các hoán vị của 1, 2, . . . , n.

4. Liệt kê các cách chia số tự nhiên n thành tổng các số nhỏ hơn n.

5.Liệt kê tất cả các xâu nhị phân độ dài n có duy nhất một dãy k bít 0 liên tiếp và duy nhất một dãy m bít 1liên tiếp.

6.Liệt kê các dãy con của dãy số An sao cho tổng các phần tử của dãy con đó đúng bằng k.

7. Liệt kê tất các các dãy con k phần tử của dãy số An sao cho tổng các phần tử của dãy co đó đúng bằng B.

8. Liệt kê tất cả các cách chọn trên mỗi hàng, mỗi cột khác nhau các phần tử của ma trận vuông A cấp n sao cho tổng các phần tử đó đúng bằng K.

9. Liệt kê tất cả các dãy số nguyên tố thuần nhất bậc k của dãy số An bằng cách tráo đổi nội dung các phần tử của dãy số An.

10. Giải bài toán n quân hậu.

Bài tập

II. Sử dụng thuật toán quay lui, viết chương trình giải các bài tập dưới đây:

1.Liệt kê các xâu nhị phân có độ dài n.

2.Liệt kê các tập con k phần tử của 1, 2, … , n.

3.Liệt kê các hoán vị của 1, 2, . . . , n.

4. Liệt kê các cách chia số tự nhiên n thành tổng các số nhỏ hơn n.

5. Liệt kê tất cả các xâu nhị phân độ dài n có duy nhất một dãy k bít 0 liên tiếp và duy nhất một dãy m bít 1liên tiếp.

6.Liệt kê các dãy con của dãy số An sao cho tổng các phần tử của dãy con đó đúng bằng k.

7. Liệt kê tất các các dãy con k phần tử của dãy số An sao cho tổng các phần tử của dãy co đó đúng bằng B.

8. Liệt kê tất cả các cách chọn trên mỗi hàng, mỗi cột khác nhau các phần tử của ma trận vuông A cấp n sao cho tổng các phần tử đó đúng bằng K.

9. Liệt kê tất cả các dãy số nguyên tố thuần nhất bậc k của dãy số An bằng cách tráo đổi nội dung các phần tử của dãy số An.

10. Giải bài toán n quân hậu.