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

Toán rời rạc 1

Bài toán liệt kê

Nguyễn Thị Trang

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

- Bài toán đưa ra danh sách tất cả cấu hình tổ hợp có thể có được gọi là bài toán liệt kê tổ hợp.
- Bài toán đếm là tìm kiếm một công thức cho lời giải, bài toán liệt kê lại cần xác định một thuật toán để theo đó có thể xây dựng được lần lượt tất cả những cấu hình quan tâm.
- **Cần đảm bảo hai nguyên tắc:**
  - Không được lặp lại một cấu hình.
  - Không được bỏ sót 1 cấu hình.

VD1

- Bài toán người bán hàng: Anh ta cần bán hàng tại 8 thành phố. Xuất phát 1 thành phố bất kì trong 8 thành phố. Phải đi qua 7 thành phố còn lại, và trở về thành phố xuất phát. Tìm cách đi với 1 con đường ngắn nhất.
- Có thể thấy rằng tổng số hành trình là n!, nhưng trong đó chỉ có thực sự (n-1)! Hành trình thực sự khác nhau(bởi vì có thể xuất phát từ 1 thành phố bất kì, nên có thể cố định 1 thành phố xuất phát).
- => 7! = 5040 cách
- => Liệt kê và đưa ra được con đường ngắn nhất.
- Khi có n=12 đã là hàng tỉ cách chọn (bùng nổ về tổ hợp)

VD2

- Cho tập hợp tất cả các số a1,a2,…an và số nguyên M.
- Hãy tìm tất cả các tập con k phần tử của dãy số {an} sao cho tổng các phần tử trong tập con đúng bằng M.
- Lời giải:

Số các tập con k phần tử của tập gồm n phần tử là C(n,k)

- Như vậy duyệt trong số C(n,k) của tập k phần tử để lấy ra những tập có tổng các phần tử đúng bằng M.
- Vì không thể xác định được có bao nhiêu tập k phần tử có tổng các phần tử đúng bằng M nên chúng ta chỉ có cách liệt kê các cấu hình thoả mãn điều kiện đã cho.

Nhận xét

- Khó khăn: Sự bùng nổ tổ hợp.
- Vì vậy phương pháp liệt kê là cách cuối cùng để có thể giải một số bài toán tổ hợp hiện nay.
- Ví dụ 1 tỉ cấu hình cần 31 năm mới giải quyết xong.

Ví dụ 3

Bài toán: Cho hình vuông gồm 25 hình vuông đơn vị. Hãy điền các số từ 0 đến 9 vào mỗi hình vuông đơn vị sao cho những điều kiện sau được thỏa mãn:

a) Đọc từ trái sang phải theo hàng ta nhận được 5 số nguyên tố có 5 chữ số;
b) Đọc từ trên xuống dưới theo cột ta nhận được 5 số nguyên tố có 5 chữ số;
c) Đọc theo hai đường chéo chính ta nhận được 2 số nguyên tố có 5 chữ số;
d) Tổng các chữ số trong mỗi số nguyên tố đều là S cho trước.

Ví dụ hình vuông dưới đây với $S = 11$.

3	5	1	1	1
5	0	0	3	3
1	0	3	4	3
1	3	4	2	1
1	3	3	1	3

Ví dụ 3

Bước 1: Tìm tập các số nguyên tố như sau

$X = \{\, x \in [10001, \dots, 99999] \mid x \text{ là nguyên tố và tổng các chữ số là } S\}$

Bước 2: Thực hiện chiến lược vét cạn như sau:

- Lấy $x \in X$ đặt vào hàng 1 (H1): ta điền được ô vuông 1, 2, 3, 4, 5.
- Lấy $x \in X$ có số đầu tiên trùng với ô số 1 đặt vào cột 1 (C1): ta điền được ô vuông 6, 7, 8, 9.
- Lấy $x \in X$ có số đầu tiên trùng với ô số 9, số cuối cùng trùng với ô số 5 đặt vào đường chéo chính 2 (D2): ta điền được ô vuông 10, 11, 12.
- Lấy $x \in X$ có số thứ nhất và số thứ 4 trùng với ô số 6 và 12 đặt vào hàng 2 (H2): ta điền được ô vuông 13, 14, 15.
- Lấy $x \in X$ có số thứ nhất, thứ hai, thứ 4 trùng với ô số 2, 13, 10 đặt vào cột 2 (C2): ta điền được ô vuông 16, 17.
- Làm tương tự như vậy ta, cho đến khi ta điền vào hàng 5 ô số 25.
- Cuối cùng ta chỉ cần kiểm tra $D1 \in X$ và $C5 \in X$?

Ví dụ 1

3	5	1	1	1
5	0	0	3	3
1	0	3	4	3
1	3	4	2	1
1	3	3	1	3

Thứ tự điền số

1	2	3	4	5
6	13	14	12	15
7	16	11	18	19
8	10	20	22	23
9	17	21	24	25

Nội dung

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

Thuật toán sinh (1/3)

- 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.

Ý tưởng thuật toán sinh (2/3)

- a. Xây dựng 1 cấu hình tổ hợp ban đầu thoả mãn điều kiện đã cho.
- b. Đưa ra cấu hình đã có.
- c. Từ thông tin cấu hình đã có, xây dung cấu hình mới cũng thoả mãn các điều kiện đã cho.
  - Nếu sinh được cấu hình mới ta tiếp tục lặp bước 2.
  - Nếu không sinh ra được ta dung lại.

Giả mã thuật toán(3/3)

- Void Generate(void)

```
{       <Xây dựng cấu hình ban đầu>;
        stop=false;
        while(not stop)
        {<Đưa ra cấu hình đang có>;
          Sinh_Kế_Tiếp;
        }
}
```

- Trong đó, **Sinh_Kế_Tiếp** thủ tục sinh cấu hình kế tiếp từ cấu hình ban đầu.
- Nếu đây là cấu hình cuối cùng, gán giá trị stop = True

VD2: Bài toán điển hình sinh xâu nhị phân.

- **Bài toán**: Liệt kê (duyệt) các xâu nhị phân có độ dài n.
- Xâu $X = (x_1 x_2 ... x_n)$: $x_i = 0, 1; i = 1, 2, ..., n$ được gọi là xâu nhị phân có độ dài n.
- **Dễ thấy**: Một dãy nhị phân x, độ dài n là biểu diễn nhị phân của một giá trị nguyên p(x) trong khoảng

[0,2^(n-1)] (2^n phần tử)

$\Rightarrow$ Xây dựng giải thuật liệt kê các dãy nhị phân theo thứ tự từ điển.

$\Rightarrow$ 0,1,2,..,2^(n-1)

VD2: Bài toán điển hình sinh xâu nhị phân.

- Ví dụ với $n = 4$, ta có 16 xâu nhị phân dưới đây:

STT	*X =(x₁...xₙ)*	*F(X)*	*STT*	*X =(x₁...xₙ)*	*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

VD2: Bài toán điển hình sinh xâu nhị phân.

- Cấu hình đầu: 00…000
- Cấu hình cuối: 11…111
- **Nhận xét**: Nếu dãy $X = x_1x_2x_3..x_n$ là dãy đang có và không phải dãy cuối cùng ➔ dãy kế tiếp nhận bằng cách + 1 (cơ số 2) vào dãy hiện tại.
- n = 8

$$\begin{array}{r} 10100100 \\ +\quad 1 \\ \hline 10100101 \end{array} \qquad\qquad \begin{array}{r} 10110011 \\ +\quad 1 \\ \hline 10110100 \end{array}$$

VD2: Bài toán điển hình sinh xâu nhị phân.

- \- Xét từ cuối dãy về đầu dãy (phải -> trái), gặp số 0 đầu tiên thay bằng số 1 và gán tất cả các phần tử sau vị trí đó bằng 0.
- Mã giả:
- Void Next_Bit_String(int *B, int n)

```
{         i = n;
          while(b_i==1 && i>1)
                {b_i=0;
                 i=i-1;
                }
          if (i==0) stop = true;
          else b_i=1


}
```

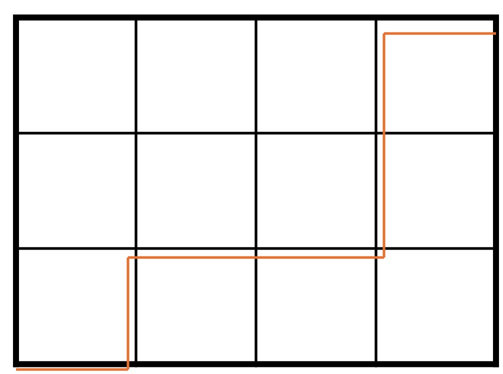
Ví dụ 2

- 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 cấu hình tiếp theo
  - Giả sử cấu hình hiện tại $x = x_1 x_2 \dots x_n$
  - 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 $x = x_1 x_2 \dots x_{k-1} 011 \dots 1$
  - Cấu hình tiếp theo $y = y_1 y_2 \dots y_n$ được tạo ra như sau
  - $y_i = x_i$ với $1 \le i \le k-1$, $y_i = 1 - x_i$ với $\mathrm{k} \le i \le n$
  - $y = x_1 x_2 \dots x_{k-1} 100 \dots 0$

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

Bài tập

- **Bài tập 1**. Cho một hình chữ nhật gồm $n{\times}m$ hình vuông đơn vị. Hãy liệt kê tất cả các đường đi từ đỉnh cuối của ô vuông cuối cùng phía bên trái đến đỉnh đầu của ô vuông trên cùng phía bên phải. Biết mỗi bước đi chỉ được phép dịch chuyển sang bên phải hoặc lên trên theo các cạnh của hình vuông đơn vị.



Bài tập

- **Bài tập 2.** Hãy liệt kê tất cả các xâu nhị phân có độ dài n sao cho mỗi xâu nhị phân có duy nhất một dãy k bít 1 liên tiếp.

- **Bài tập 3.** Hãy liệt kê tất cả các xâu nhị phân có độ dài n sao cho mỗi xâu nhị phân có duy nhất một dãy m bít 1 liên tiếp và duy nhất một dãy có k bít 0 liên tiếp.

Bài tập

- **Bài tập 4.** Chuỗi ký tự $X = (x_1, x_2, \ldots, xn)$ được gọi là chuỗi ký tự AB nếu $x_i = \text{‘}A\text{’}$ hoặc $x_i = \text{‘}B\text{’}$. Chuỗi X được gọi là chuỗi AB bậc k nếu X tồn tại duy nhất một dãy k kí tự A liên tiếp. Hãy liệt kê tất cả các chuỗi AB bậc k.

- **Bài tập 5.** Cho dãy số $A = (a_1, a_2, \ldots, a_n)$ gồm n số tự nhiên khác nhau và số tự nhiên k. Hãy liệt kê tất cả các dãy con của dãy số A sao cho tổng các phần tử của dãy con đó đúng bằng k.

Ví dụ 3

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

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

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 $X = (x_1, x_2, \dots, xk)$ là đứng trước tập con $Y = (y_1, y_2, \dots, yk)$ nếu tìm được chỉ số t sao cho $x_1 = y_1$, $x_2 = y_2, \dots, x_{t-1} = y_{t-1}, x_t < y_t$.

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

Ví dụ 3

- Thuật toán sinh cấu hình (tổ hợp) tiếp theo
- Giả sử cấu hình hiện tại là $X = (x_1, x_2, \dots, x_k)$
- Nếu $x_i = n - k + i$ với mọi $i = 1,2, \dots, k$ thì X là cấu hình cuối cùng. Thuật toán duyệt kết thúc
- Gọi t là chỉ số lớn nhất (x_t là số đầu tiên từ phải sang) sao cho $x_t < n - k + t$
- Cấu hình tiếp theo $Y = (y_1, y_2, \dots, y_k)$ được sinh ra như sau
  - $y_i = x_i$ với $i < t$,
  - $y_t = x_t + 1$,
  - $y_i = y_t + (i - t)$ với $i > t$

Ví dụ 3

- Mã giả:
  - Void next_combination(int *X){

```
## sinh k- tập con kế tiếp theo thứ tự từ điển.
## của tập con (x1,x2,..xk) khác {n-k+1,…,n}
i = k;
While (x[i] == n-k + i ){i = i – 1;}
X[i] = X[i] + 1;
for (int j= i+1; j < k; j ++){
     X[j] = X[i] + j - i ;}
return X
}
```

Bài tập

- **Bài tập 6.** Cho dãy số $A = (a_1, a_2, . . . , a_n)$ 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 $A = (a_1, a_2, \ldots, a_n)$. 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: Bài toán sinh các hoán vị

- Liệt kê (duyệt) các hoán vị của $1,2,\dots,n$.

Mỗi hoán vị của $1,2,\dots,n$ là một cách xếp có tính đến thứ tự của $1,2,\dots,n$. Số các hoán vị là $n!$. Ví dụ với $n = 3$ ta có 6 hoán vị dưới đây.

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

Ví dụ 4: Bài toán sinh hoán vị

- **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: Bài toán sinh hoán vị

- **Thuật toán**: Dựa trên việc xây dựng hoán vị kế tiếp theo thứ tự từ điển của hoán vị cho trước a = (a1,a2,..,an)
- **Phân tích:**
- Giả sử $a_{n-1} < a_n$ nếu đổi chỗ a_{n-1} cho a_n ta nhận được 1 hoán vị mới ngay sau hoán vị đã cho.
- Nếu $a_{n-1} > a_n$ thì không thể nhận được hoán vị lớn hơn bằng việc đổi chỗ.
- Ta phải xét 3 số hạng cuối cùng a_{n-2}, a_{n-1}, a_n
  - Nếu $a_{n-2} < a_{n-1}$ thì có thể sắp xếp lại 3 số cuối cùng, để có thể nhận được 1 hoán vị mới liền sau hoán vị xuất phát.
  - Trong 2 số a_{n-1} và a_n ta chọn số nhỏ hơn (mà vẫn > a_{n-2}) để đặt vào vị trí n-2, sau đó đặt số nguyên còn lại và số a_{n-2} vào 2 vị trí cuối theo thứ tự tăng dần.
  - Nếu $a_{n-2} > a_{n-1}$ (và $a_{n-1} > a_n$) xét thêm a_{n-3}

Ví dụ 4: Bài toán sinh hoán vị

- **Tổng quát:**
- **Bước 1**: Tìm cặp số nguyên liền kề đầu tiên tính từ phải sang trái mà số đầu nhỏ hơn số sau: tức là tìm các số nguyên a_j và a_{j+1} sao cho: $a_j < a_{j+1}$ và $a_{j+1} > a_{j+2} > \cdots > a_n$
- **Bước 2**: Đặt vào vị trí thứ j số nguyên nhỏ nhất trong các số > a_j của tập $(a_{j+1}, a_{j+2}, \dots, a_n)$
- **Bước 3**: Liệt kê theo thứ tự tăng dần các số còn lại: $a_{j+1}, a_{j+2}, \dots, a_n$ và a_j vào các vị trí j+1,...,n

Ví dụ 4: Bài toán sinh hoán vị

- **Vd**: Hoán vị hiện tại là x = (3,2,6,5,4,1)
- Tương ứng x = (x1,x2,x3,x4,x5,x6)
- Tìm hoán vị tiếp theo, đứng sau liền kề.
- **Bước 1**: Cặp số nguyên liền kề đầu tiên từ phải sang trái thoả mãn : $a_i < a_{i+1}$ là x2 và x3 , xét x2=2
- **Bước 2**: Số nguyên thuộc (x3,x4,x5) nhỏ nhất lớn hơn x2 là x5 = 4, đổi chỗ => x2=4, x5=2
- **Bước 3**: Liệt kê tăng dần các số còn lại (x3,x4,x6 và x2(ban đầu)) => (x6,x2,x4,x3) => (1,2,5,6)
- ➔ (3,4,1,2,5,6)

Ví dụ 4: Bài toán sinh hoán vị

- Nhận xét: Nếu đổi chỗ x5 và x2 thì sẽ được x5=2 và đoạn cuối vẫn được xếp giảm dần (6,5,2,1)
- Khi đó muốn biểu diễn nhỏ nhất cho các giá trị trong đoạn cuối thì ta chỉ cần đảo ngược đoạn cuối (sắp xếp tăng dần)

Ví dụ 4: Bài toán sinh hoán vị

- Giả mã thuật toán:
- Void Next_permutation(int *A, int n){

```
        int j,k,r,s,temp;
        j=n-1;
        while(A[j] > A[j+1] && j >= 2) {j = j-1;}
        if (j<2) {stop =true;}
        else{
                k=n;
                while (A[j] > A[k]) {
                        k = k – 1;
                        temp = A[j]; A[j] = A[k]; A[k] = temp;}
                r=j+1; s=n;
                while(r<s) { temp=A[r]; A[r] = A[s]; A[s]=temp}
}

}
```

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
  - $y_i = x_i$ với $i \leq t - 2$
  - y_{t-1} bằng phần tử nhỏ nhất trong tập $x_t,\ldots,x_n$ và lớn hơn x_{t-1} (ký hiệu là a)
  - $y_t,\ldots,y_n$ là dãy sắp xếp tăng dần gồm các số trong tập $\{x_{t-1}, x_t, \ldots, x_n\} \backslash \{a\}$

Bài tập

- **Bài tập 8.** Một dãy số tự nhiên bất kỳ $A_n = \{a_1, a_2, \ldots, a_n\}$ được gọi là một đường nguyên tố bậc k nếu tổng k phần tử liên tiếp bất kỳ của dãy số A_n là một số nguyên tố $(k \leq n)$. Ví dụ dãy số $A_n = \{3, 27, 7, 9, 15\}$ là một đường nguyên tố bậc 3. Cho dãy số A_n. Hãy liệt kê tất cả các đường nguyên tố bậc k có thể có được tạo ra bằng cách tráo đổi các phần tử khác nhau của dãy số A_n.

- Ví dụ với dãy $A = (3, 7, 9, 15, 27)$ ta sẽ thành lập được 4 dãy nguyên tố thuần nhất bậc 3 như dưới đây:

3	27	7	9	15
15	9	7	3	27
15	9	7	27	3
27	3	7	9	15

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

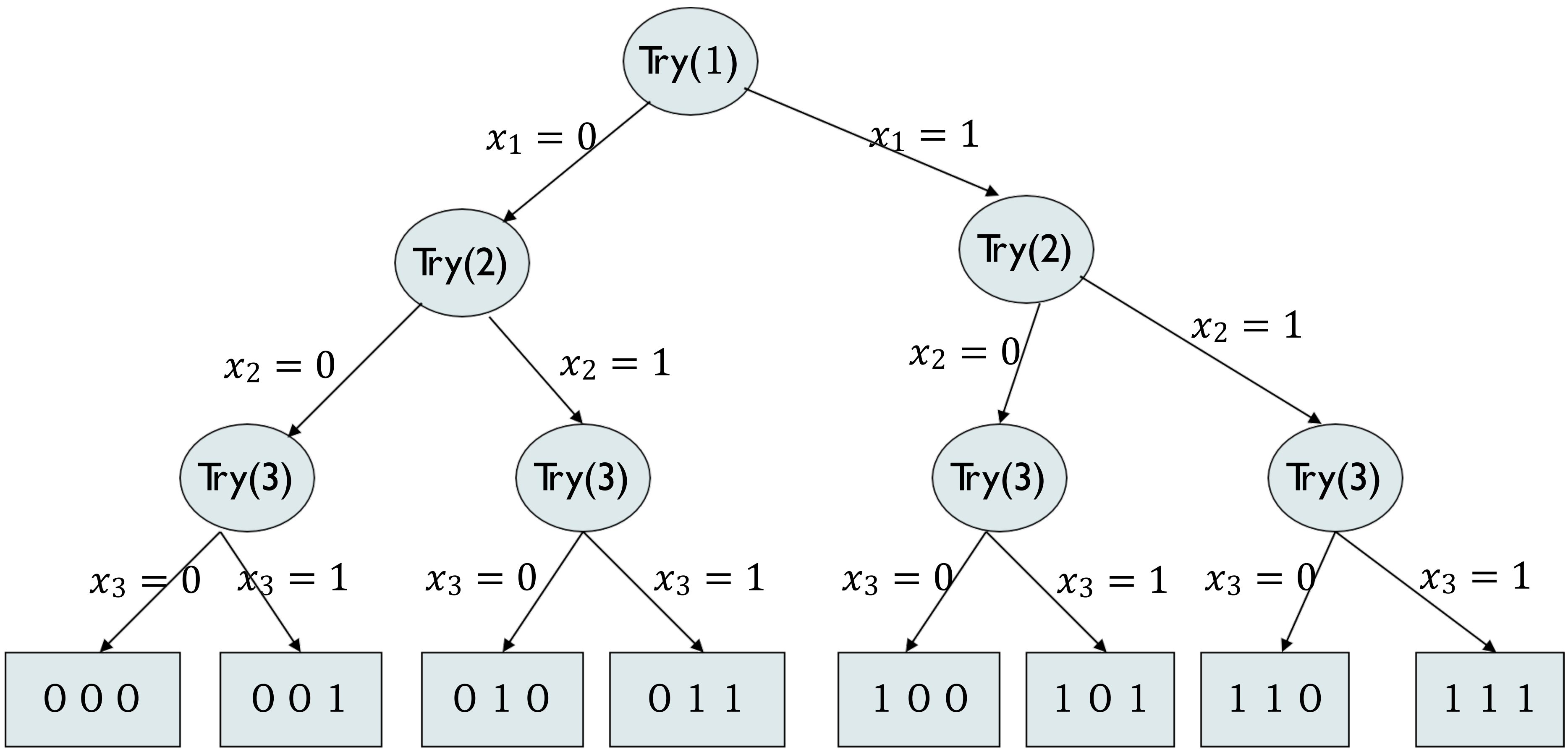
- Liệt kê (duyệt) các xâu nhị phân có độ dài n.

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

Bài tập

- **Bài tập 9**. Sử dụng thuật toán quay lui, hãy liệt kê tất cả các phần tử của tập:

$$D = \left\{ X = (x_1, x_2, .., x_n) : \sum_{i=1}^{n} a_i x_i \leq W \wedge \sum_{i=1}^{n} c_i x_i = K \right\}.$$

Trong đó, $x_i = 0, 1; c_i, a_i \in Z^{+};$

$n \leq 100, W \leq 32000;\ K \leq 32000.$

Ví dụ 6

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

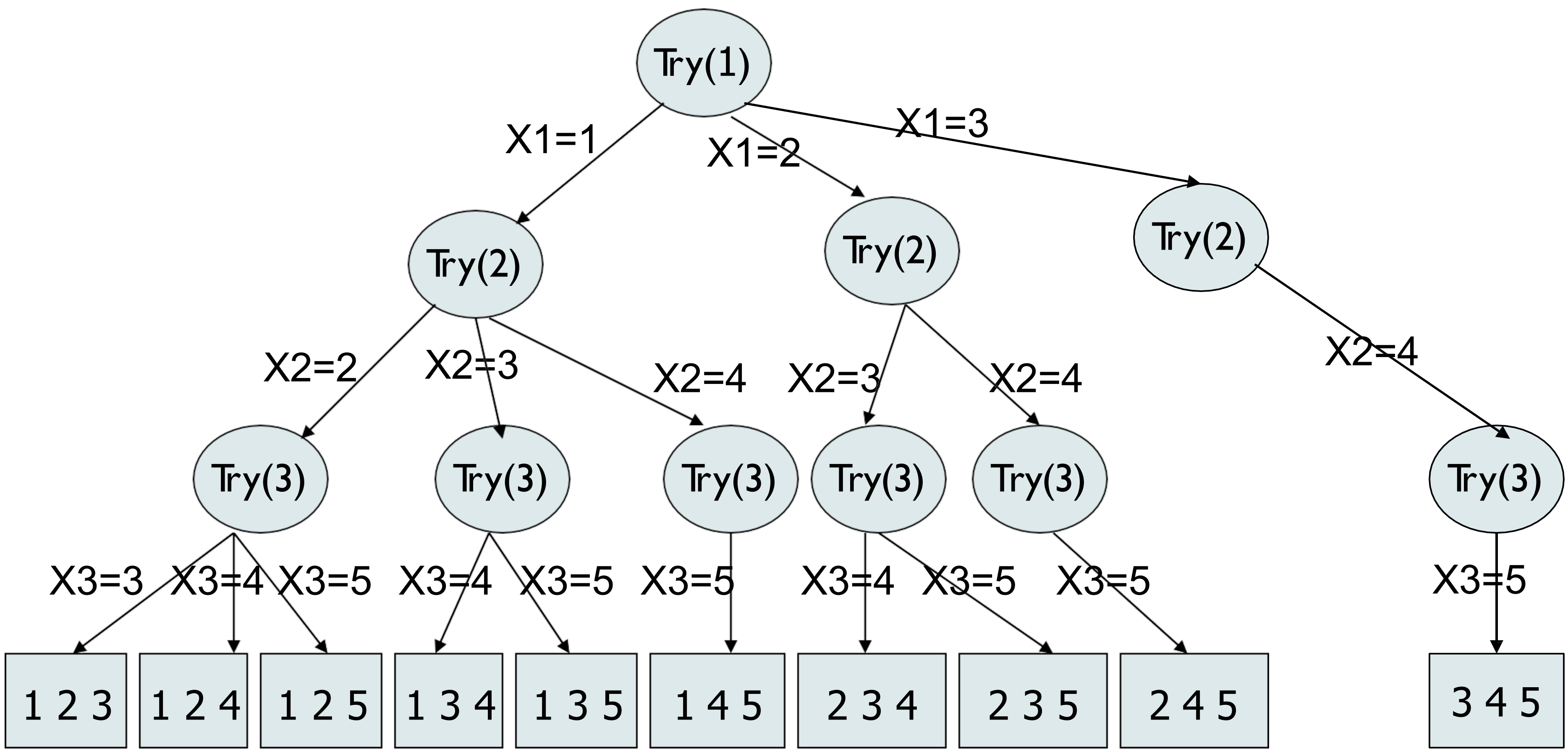
Mỗi tổ hợp chập k của $1, 2, \dots, n$ là một tập con k phần tử khác nhau của $1, 2, \dots, n$.

```
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 $X[1], \dots, X[k]$

Khi đó, để duyệt các tập con k phần tử của $1, 2, \dots, n$ 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

Bài tập

- **Bài tập 10**. Sử dụng thuật toán quay lui, hãy liệt kê tất cả các phần tử của tập:

$$D = \left\{ X = (x_1, x_2, .., x_n) : \sum_{i=1}^{n} x_i = K \wedge \sum_{i=1}^{n} a_i x_i = S \right\}.$$

Trong đó, $x_i = 0, 1$; $a_i \in Z^{+}$;

$n \leq 100, K \leq 100$; $S \leq 32000$.

Ví dụ 7

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

Mỗi hoán vị $X = (x_1, x_2, \ldots, xn)$ là bộ có tính đến thứ tự của $1, 2, \ldots, n$. 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.

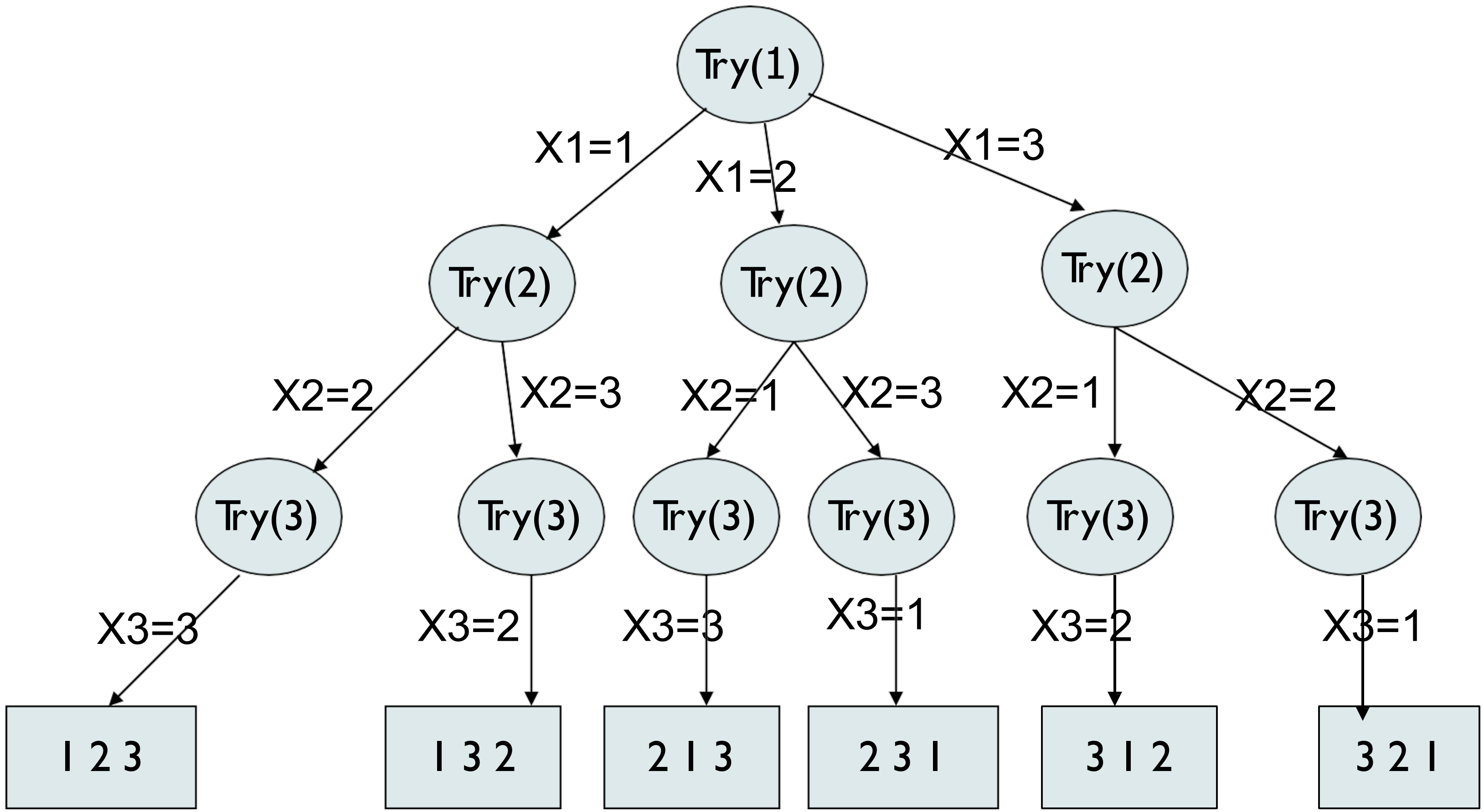
- Nếu $unused[i] = True$ điều đó có nghĩa giá trị i được chấp thuận
- $unused[i] = False$ tương ứng với giá trị i không được phép sử dụng

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 $1, 2, \dots, n$ 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

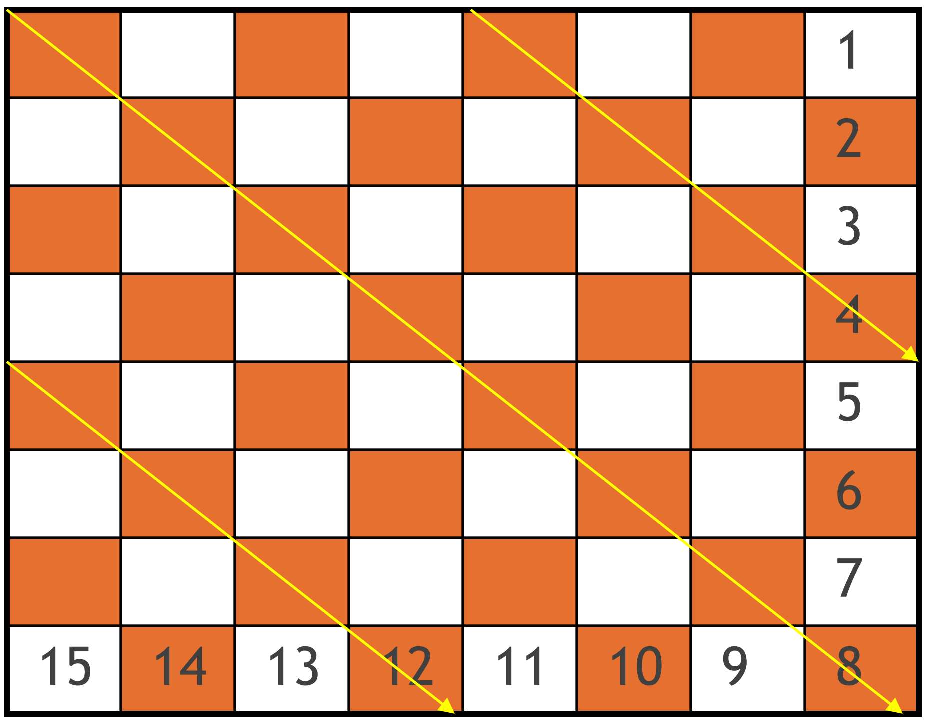
- **Bài toán n quân hậu**. Trên bàn cờ kích cỡ $n \times n$, hãy đặt n quân hậu mỗi quân trên 1 hàng sao cho tất cả các quân hậu đều không ăn được lẫn nhau.

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.

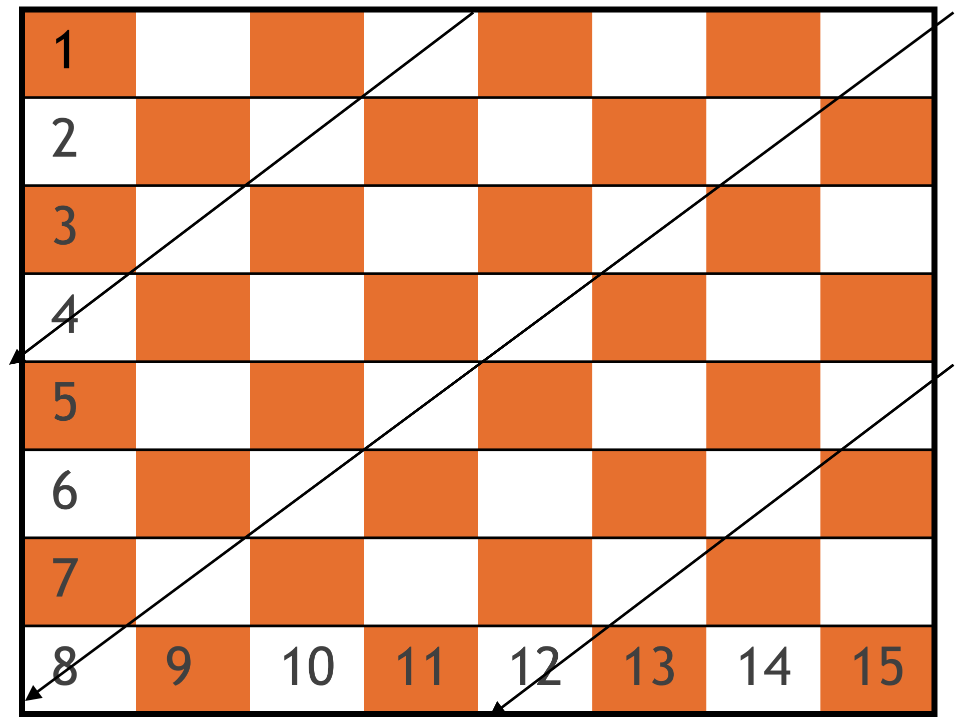
Ta có n cột $A = (a_1, \ldots, a_n)$, có $Xuoi[2 * n - 1]$ đường chéo xuôi, $Nguoc[2 * n - 1]$ đường chéo ngược.

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 $1, 2, \ldots, n$.
3. Liệt kê các hoán vị của $1, 2, \ldots, n$.
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 1liền tiếp.
6. Liệt kê các dãy con của dãy số A_n sao cho tổng các phần tử của dãy con đó đúng bằng k.

1.7. Liệt kê tất các các dãy con k phần tử của dãy số A_n 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 A_n bằng cách tráo đổi nội dung các phần tử của dãy số A_n.

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 $1, 2, \dots, n$.
3. Liệt kê các hoán vị của $1, 2, \dots, n$.
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 A_n sao cho tổng các phần tử của dãy con đó đúng bằng k.

2.7. Liệt kê tất các các dãy con k phần tử của dãy số A_n 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 A_n bằng cách tráo đổi nội dung các phần tử của dãy số A_n.

10. Giải bài toán n quân hậu.