Toán rời rạc 1

Discrete mathematics 1

Bài 4: Bài toán liệt kê

8/20/2020

Nội dung Bài 4

1. Giới thiệu bài toán
   - Vét cạn: brute force, exhaustive search
2. Phương pháp sinh
3. Phương pháp quay lui
4. Bài tập

Tổng số khả năng mã hóa?

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

Ví dụ 1 Mở đầu

	6	7	8	9	10
1	3	5	1	1	1
2	5	0	0	3	3
3	1	0	3	4	3
4	1	3	4	2	1
5	1	3	3	1	3

11 12

Tổng 5 chữ số = S

- Đặt 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 ở trên thỏa mãn điều kiện với *S* = 11.

Ví dụ 1 Mở đầu

H: Hàng

C: Cột

D: Đường chéo

x_1	x_2	x_3	x_4	x_5
y_2	a_2	a_3	z_4	a_5
y_3	b_3	z_3	…	…
y_4	z_2	…	…	…
y_5	b_5	…	…	…

- Phương pháp:
  - Bước 1: Tìm tập X = { $x \in$ [10001, …, 99999] | x: *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 $y \in X$ có số đầu tiên trùng với ô số 1 đặt vào cột 1 (C1): ta điền được ô vuông 6, 7, 8, 9.
    - Lấy $z \in X$ 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 $a \in X$ 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 $b \in X$ 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 cho đến khi ta điền vào hàng 5 ô số 25.
    - Cuối cùng ta cần kiểm tra $D1 \in X$, C4 $\in X$ và C5 $\in X$?

Ví dụ 1 Mở đầu

H: Hàng

C: Cột

D: Đường chéo

x_1	x_2	x_3	x_4	x_5
y_2	a_2	a_3	z_4	a_5
y_3	b_3	z_3	...	...
y_4	z_2	...	...	...
y_5	b_5	...	...	...

Một nghiệm

D1

3	5	1	1	1	H1
5	0	0	3	3	H2
1	0	3	4	3	H3
1	3	4	2	1	H4
1	3	3	1	3	H5
C1	C2	C3	C4	C5	

D2

Thứ tự điền số

D1

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
C1	C2	C3	C4	C5

D2

Ví dụ 1 Mở đầu

Vét cạn = brute force, exhaustive search

```
Cac nghiem cua bai toan:

 1  1  3  5  1            2  5  1  2  1
 1  4  0  3  3            1  0  2  7  1
 3  0  3  2  3            5  4  1  0  1
 5  3  2  0  1            2  1  6  1  1
 1  3  3  1  3            1  1  1  1  7

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

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

Bài tập 1 Mở đầu

- Viết chương trình (C, C++) giải 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.

  Nhập S từ bàn phím.

Nội dung Bài 4

1. Giới thiệu bài toán
   - Vét cạn: brute force, exhaustive search
2. Phương pháp sinh
3. Phương pháp quay lui
4. Bài tập

Thuật toán sinh (1/2)

- Thuật toán sinh được dùng để giải lớp các bài toán thỏa mãn hai điều kiện:
  1. 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.
  2. 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)

Bước1 (Khởi tạo):

<Thiết lập cấu hình đầu tiên>;

Bước 2 (Bước lặp):

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

Ví dụ 2 Thuật toán sinh

- Bài toán: Liệt kê (duyệt) các xâu nhị phân có độ dài n.
- Xâu $X = x_1x_2 \ldots x_n$ với $x_i = 0, 1; \quad i = 1, 2, \ldots, n$ 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 Thuật toán sinh

Theo hệ thập phân
$y = x + 1$

- Thứ tự trên tập cấu hình được sắp theo giá trị thập phân của số nhị phân mà cấu hình biểu diễn
  - Cấu hình đầu tiên là xâu nhị phân gồm n chữ số 0
  - Cấu hình cuối cùng là xâu nhị phân 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, dừng thuật toán
    - 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 $k \le i \le n$
    - Hay: $y = x_1 x_2 \dots x_{k-1} 100 \dots 0$

Ví dụ 2 Thuật toán sinh

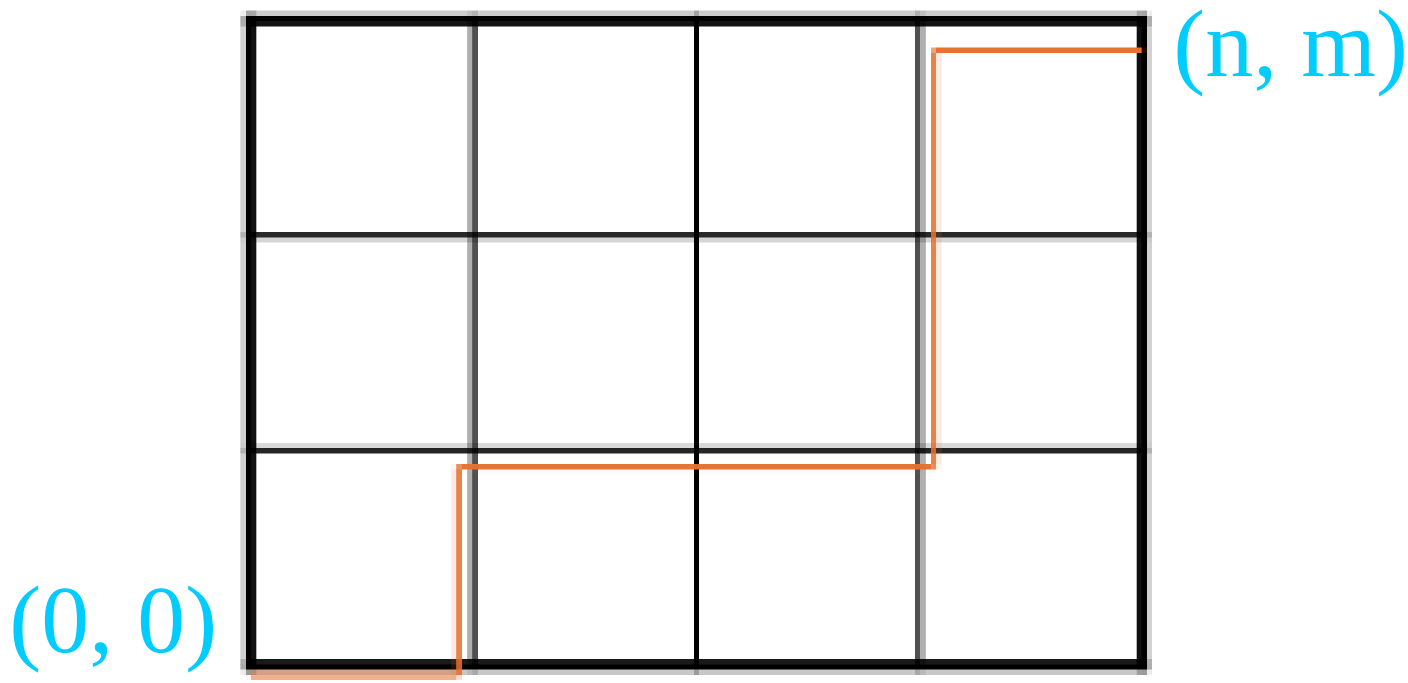
```
Nhap n = 3

Xau thu   1:  0  0  0
Xau thu   2:  0  0  1
Xau thu   3:  0  1  0
Xau thu   4:  0  1  1
Xau thu   5:  1  0  0
Xau thu   6:  1  0  1
Xau thu   7:  1  1  0
Xau thu   8:  1  1  1
```

- Thứ tự trên tập cấu hình được sắp theo g… của số nhị phân mà cấu hình biểu diễn
  - Cấu hình đầu tiên là xâu nhị phân gồm n chữ số 0
  - Cấu hình cuối cùng là xâu nhị phân 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, dừng thuật toán
    - 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 $k \le i \le n$
    - Hay: $y = x_1 x_2 \dots x_{k-1} 100 \dots 0$

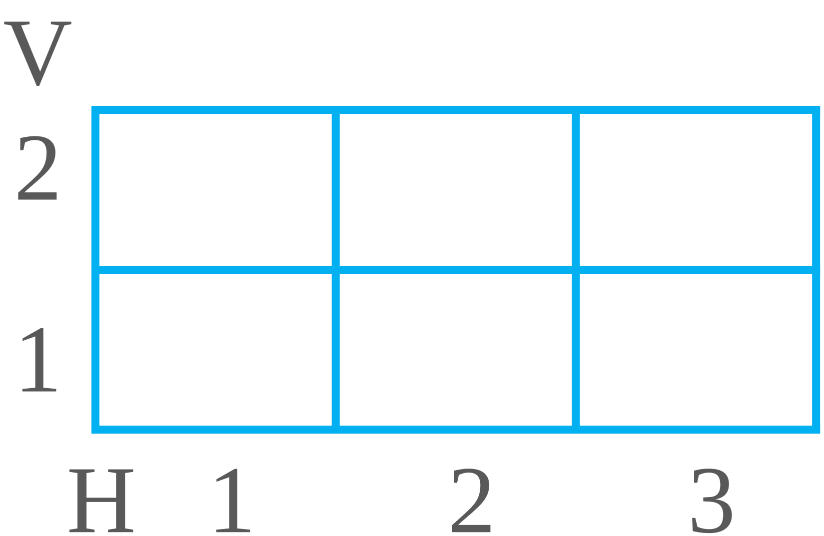
Bài tập 2 Thuật toán sinh

- Cho một hình chữ nhật gồm $n \times m$ hình vuông đơn vị. Hãy liệt kê tất cả các đường đi từ (0, 0) đến (n, m). 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 2 Thuật toán sinh

- Cho một hình chữ nhật gồm $n \times m$ hình vuông đơn vị. Hãy liệt kê tất cả các đường đi từ (0, 0) đến (n, m). Biết mỗi bước đi chỉ được phép dịch chuyển sang bên phải hoặc lên trên theo các cạnh của hình vuông đơn vị.




```
Nhap so buoc dung N, so buoc ngang M: 2 3

Cach buoc thu  1: V1 V2 H1 H2 H3
Cach buoc thu  2: V1 H1 V2 H2 H3
Cach buoc thu  3: V1 H1 H2 V2 H3
Cach buoc thu  4: V1 H1 H2 H3 V2
Cach buoc thu  5: H1 V1 V2 H2 H3
Cach buoc thu  6: H1 V1 H2 V2 H3
Cach buoc thu  7: H1 V1 H2 H3 V2
Cach buoc thu  8: H1 H2 V1 V2 H3
Cach buoc thu  9: H1 H2 V1 H3 V2
Cach buoc thu 10: H1 H2 H3 V1 V2
```

Bài tập 3, 4 Thuật toán sinh

- 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 *k* bít 1 liên tiếp.

```
Nhap do dai xau nhi phan n = 5
Nhap do dai day k bit 1 lien tiep k = 3

Xau thu  1 do dai  5 co duy nhat  3 so 1 lien tiep:  0  0  1  1  1
Xau thu  2 do dai  5 co duy nhat  3 so 1 lien tiep:  0  1  1  1  0
Xau thu  3 do dai  5 co duy nhat  3 so 1 lien tiep:  1  0  1  1  1
Xau thu  4 do dai  5 co duy nhat  3 so 1 lien tiep:  1  1  1  0  0
Xau thu  5 do dai  5 co duy nhat  3 so 1 lien tiep:  1  1  1  0  1
```

- Bài tập 4:
  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.

```
Nhap do dai xau nhi phan n = 5
Nhap do dai day m bit 1 lien tiep m = 2
Nhap do dai day k bit 0 lien tiep k = 1

Xau 1 dai 5 co duy nhat 2 so 1 lien tiep, 1 so 0 lien tiep:  0  0  1  1  0
Xau 2 dai 5 co duy nhat 2 so 1 lien tiep, 1 so 0 lien tiep:  0  1  1  0  0
```

Bài tập 5, 6 Thuật toán sinh

```
Nhap do dai chuoi ky tu AB n = 6
Nhap bac cua chuoi AB m = 4

Chuoi AB thu 1 dai 6 bac 4:  B  B  A  A  A  A
Chuoi AB thu 2 dai 6 bac 4:  B  A  A  A  A  B
Chuoi AB thu 3 dai 6 bac 4:  A  B  A  A  A  A
Chuoi AB thu 4 dai 6 bac 4:  A  A  A  A  B  B
Chuoi AB thu 5 dai 6 bac 4:  A  A  A  A  B  A
```

- Bài tập 5:
  Chuỗi ký tự $X = (x_1, x_2, \ldots, x_n)$ 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.

```
So phan tu cua day n = 8
11 12 13 14 15  0  1  2
Tong phan tu cua day k = 14
Ket qua   1:  14
Ket qua   2:  14  0
Ket qua   3:  13  1
Ket qua   4:  13  0  1
Ket qua   5:  12  2
Ket qua   6:  12  0  2
Ket qua   7:  11  1  2
Ket qua   8:  11  0  1  2
```

- Bài tập 6:
  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. Vào từ file, ra màn hình.

Ví dụ 3 Thuật toán sinh tổ hợp

- 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 $X = \{x_1, \dots, x_k\}$	STT	Tập con $X = \{x_1, \dots, x_k\}$
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 Thuật toán sinh tổ hợp

- Thuật toán sinh cấu hình (tổ hợp) tiếp theo
  - Giả sử cấu hình hiện tại là $X = (x_1, x_2, \ldots, x_k)$
  - Nếu $x_i = n - k + i$ với mọi $i = 1, 2, \ldots, k$ 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 $Y = (y_1, y_2, \ldots, y_k)$ 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 Thuật toán sinh tổ hợp

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

```
Nhap n, k: 5 3

Ket qua   1:    1   2   3
Ket qua   2:    1   2   4
Ket qua   3:    1   2   5
Ket qua   4:    1   3   4
Ket qua   5:    1   3   5
Ket qua   6:    1   4   5
Ket qua   7:    2   3   4
Ket qua   8:    2   3   5
Ket qua   9:    2   4   5
Ket qua  10:    3   4   5
```

Bài tập 7, 8 Thuật toán sinh tổ hợp

```
So phan tu cua day n = 7
 5 10 15 20 25 30 35
So phan tu cua day con k = 3
Tong phan tu cua day con P = 50

Ket qua   1:  10 15 25
Ket qua   2:   5 20 25
Ket qua   3:   5 15 30
Ket qua   4:   5 10 35
```

- Bài tập 7 (Vào từ file, ra màn hình):
  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: (5, 10, 35), (5, 20, 25), (10, 15, 25), …

```
So phan tu cua day n = 5
 1  3  2  4  5
So phan tu cua day con k = 3
Ket qua   1:   2  4  5
Ket qua   2:   3  4  5
Ket qua   3:   1  4  5
Ket qua   4:   1  2  5
Ket qua   5:   1  2  4
Ket qua   6:   1  3  5
Ket qua   7:   1  3  4
```

- Bài tập 8 (Vào từ file, ra màn hình):
  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 Thuật toán sinh hoán vị

- 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 $X = (x_1, \dots, x_n)$
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 Thuật 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 $x_1 = y_1$, $x_2 = y_2$, …, $x_{k-1} = y_{k-1}$, $x_k < y_k$.
- 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 hoán vị

- 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 $\{x_{t-1}, x_t, \ldots, x_n\} \setminus \{a\}$

Ví dụ 4 Thuật toán sinh hoán vị

```
Nhap n: 3

Ket qua   1:   1  2  3
Ket qua   2:   1  3  2
Ket qua   3:   2  1  3
Ket qua   4:   2  3  1
Ket qua   5:   3  1  2
Ket qua   6:   3  2  1
```

- Thuật toán sinh cấu hình (hoán vị) tiếp t
  - 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 $i \le t - 2$
    - 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 9 Thuật toán sinh hoán vị

```
So phan tu cua day n = 5
 3 27  7  9 15
Bac cua duong nguyen to k = 3
Ket qua   1:    3 27  7  9 15
Ket qua   2:   27  3  7  9 15
Ket qua   3:   15  9  7  3 27
Ket qua   4:   15  9  7 27  3
```

- 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 $(k \le n)$.

  VD dãy $A_n = \{3, 27, 7, 9, 15\}$ 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

Vào từ file, in ra màn hình

Nội dung Bài 4

1. Giới thiệu bài toán
   - Vét cạn: brute force, exhaustive search
2. Phương pháp sinh
3. Phương pháp quay lui
4. 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)
                InKetQua();
            else
                Back_Track (i+1);
        }
    }
}
```

Đệ quy

Ví dụ 5 Thuật toán quay lui

- 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.

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

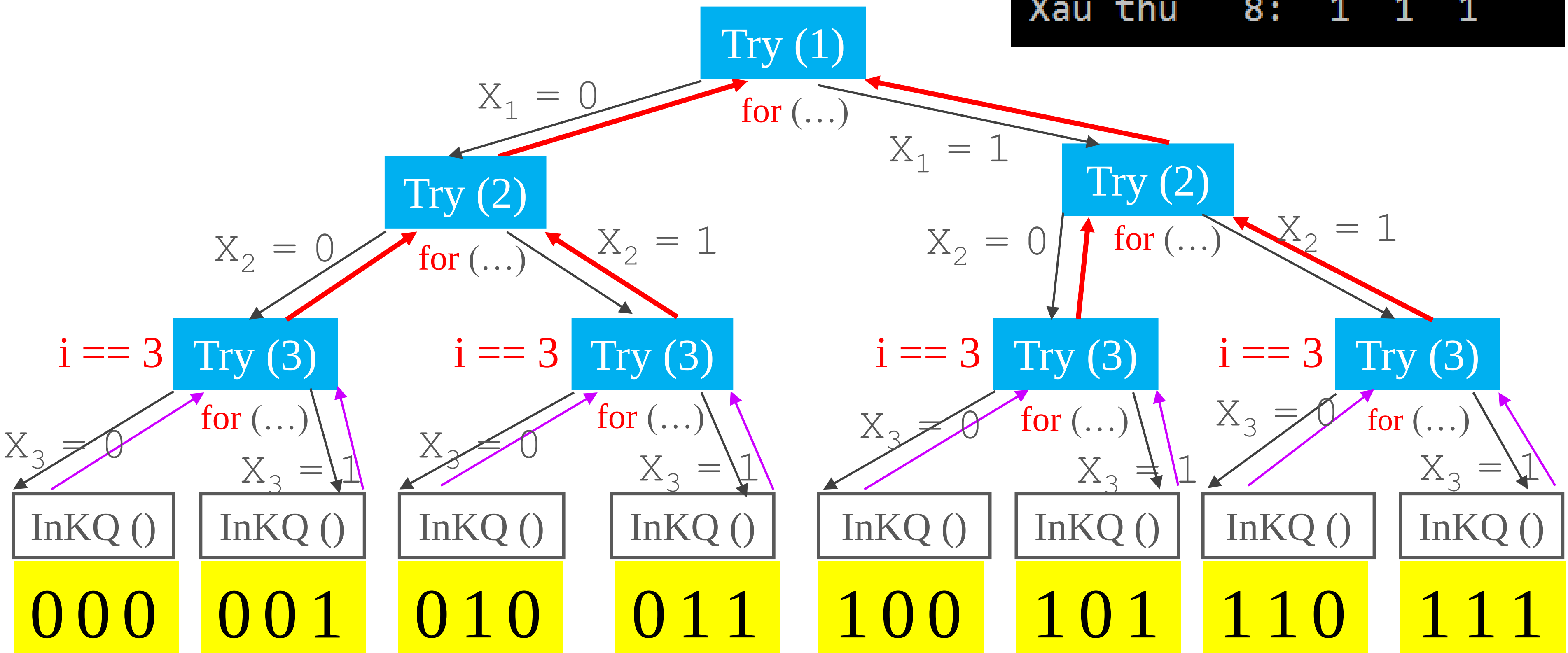
Khi đó, để duyệt các xâu nhị phân có độ dài n ta chỉ cần gọi đến thủ tục quay lui - BackTracking Try (1).

Ví dụ 5 Thuật toán quay lui

```
BackTracking, nhap n = 3

Xau thu  1:  0  0  0
Xau thu  2:  0  0  1
Xau thu  3:  0  1  0
Xau thu  4:  0  1  1
Xau thu  5:  1  0  0
Xau thu  6:  1  0  1
Xau thu  7:  1  1  0
Xau thu  8:  1  1  1
```

Quay lui



Bài tập 10 Thuật toán quay lui

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

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

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

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

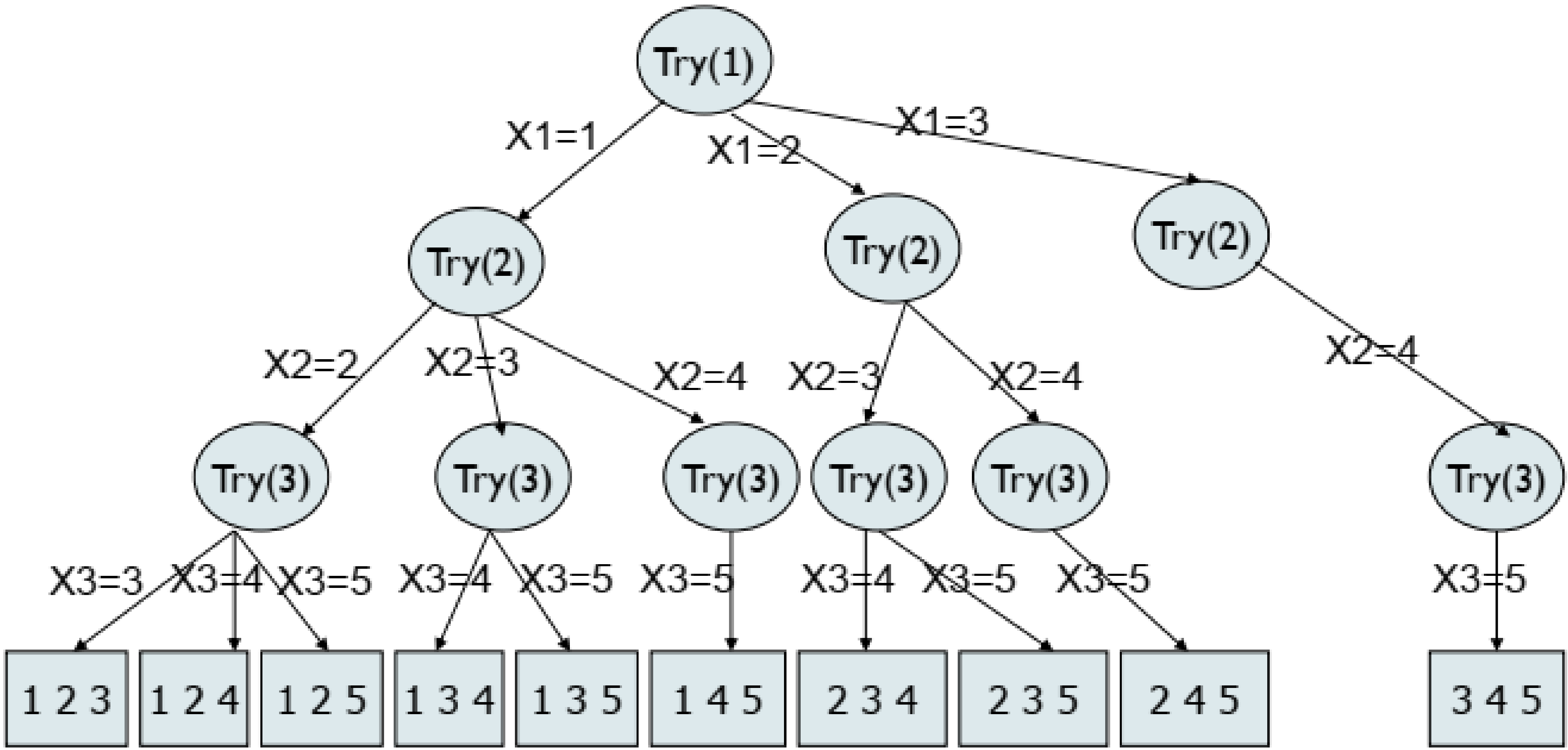
Ví dụ 6 Thuật toán quay lui

- 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)
            InKetQua ();
        else
            Try (i+1);
    }
}
```

- Khởi đầu với X[0] = 0
- Khi đó, để duyệt các tập con k phần tử của n phần tử, ta chỉ cần gọi đến thủ tục quay lui - BackTracking Try (1).

Ví dụ 6 Thuật toán quay lui



Bài tập 11 Thuật toán quay lui

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

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

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

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

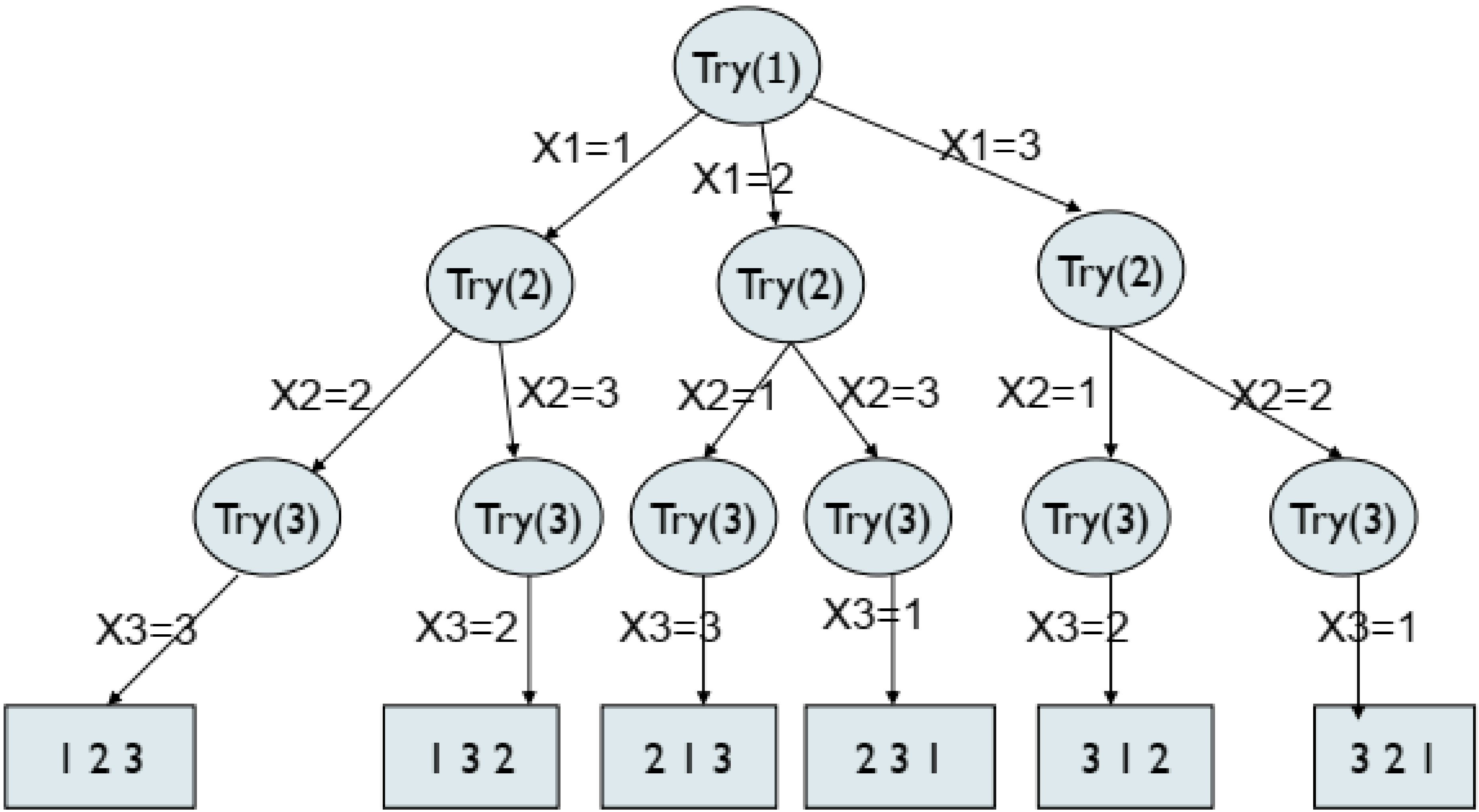
Ví dụ 7 Thuật toán quay lui có điều kiện

- Sử dụng phương pháp quay lui liệt kê (duyệt) các hoán vị của 1, 2, …, n.
- Mỗi hoán vị $X = (x_1, x_2, \ldots, x_n)$ 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*[1 .. n] (gồm n phần tử).
  - $unused[i] = 1$: giá trị i được chấp thuận
  - $unused[i] = 0$: giá trị i không được phép sử dụng

Ví dụ 7 Thuật toán quay lui có điều kiện

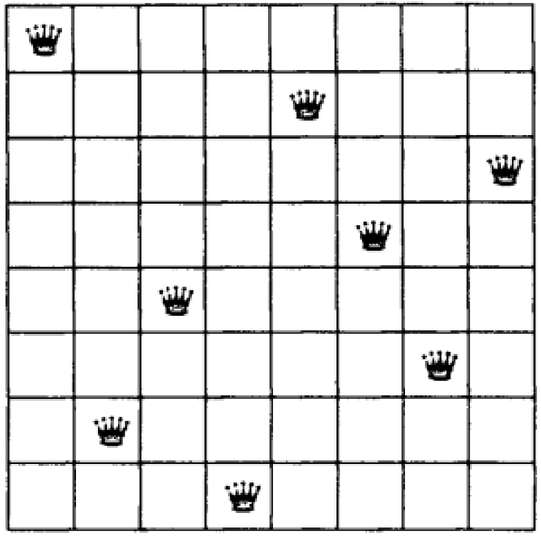
```
void Try (int i){
    for (int j = 1; j <= n ; j++){
        if (unused[j]){
            X[i] = j;
            unused[j] = 0;          //false
            if (i == n)
                InKetQua();
            else
                Try (i+1);
            unused[j] = 1;          //true
        }
    }
}
```

Ví dụ 7 Thuật toán quay lui có điều kiện



Ví dụ 8 Thuật toán quay lui

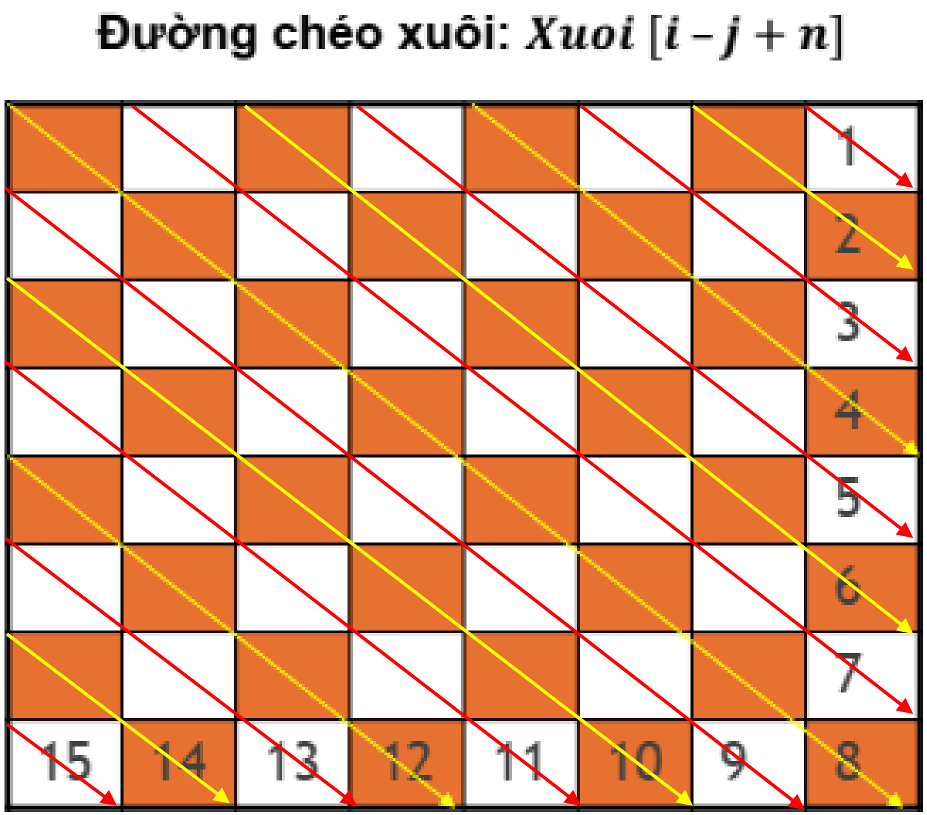
n = 8

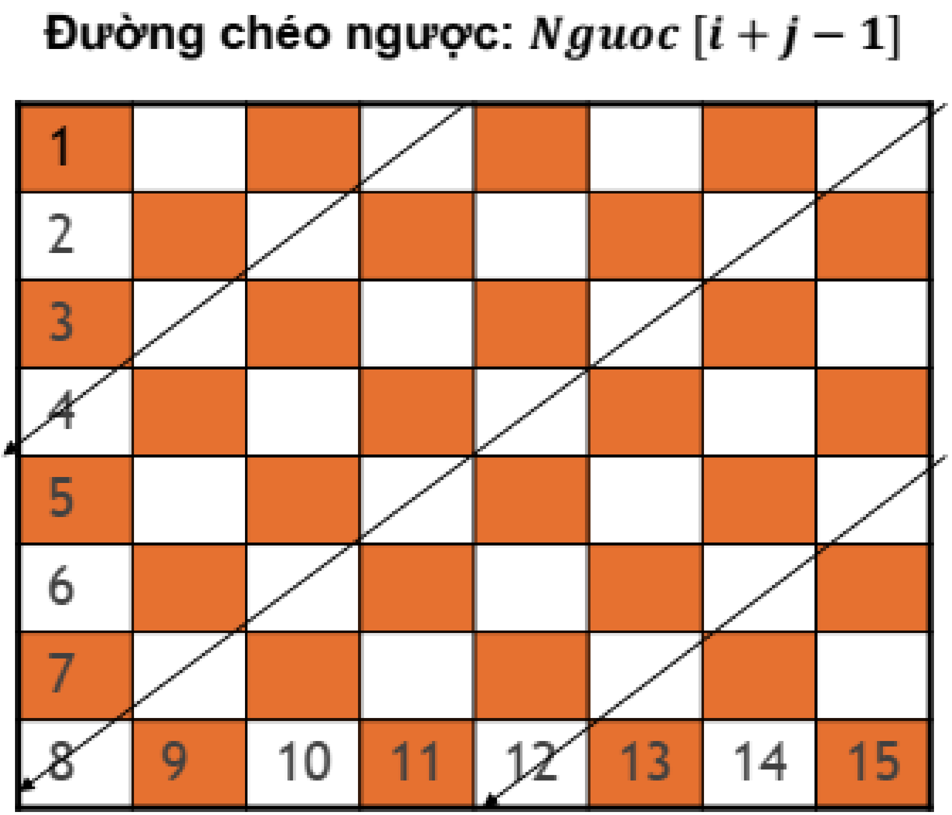


- Bài toán ***n*** quân hậu:
  - Trên bàn cờ kích cỡ $n \times n$,
    hãy đặt n quân hậu, mỗi quân trên 1 hàng
    sao cho các quân hậu đều không ăn được lẫn nhau.
- Phân tích:
  - Gọi $X = (x_1, x_2, \ldots, x_n)$ là một nghiệm của bài toán.
    $x_i = j$ nghĩa là: hậu hàng thứ i đặt ở cột j.
    Để các hậu không thể ăn lẫn nhau: mỗi hàng chỉ có 1 hậu, hậu hàng i phải ở cột chưa có hậu khác, không được cùng đường chéo xuôi, không được cùng đường chéo ngược với hậu khác.
  - Ta có n cột $A = (a_1, \ldots, a_n)$
  - Số đường chéo xuôi: $Xuoi[2 * n - 1]$,
    Số đường chéo ngược: $Nguoc[2 * n - 1]$.

Ví dụ 8 Thuật toán quay lui

Ô *i, j* thuộc đường chéo:
- Xuôi thứ: $i - j + n$
- Ngược thứ: $i + j - 1$





15 đường chéo xuôi / ngược với n = 8

Ví dụ 8 Thuật toán quay lui

```
void Try (int i){        //Mỗi hàng đặt 1 hậu
    for(int j = 1; j <= n; j++){
        if(A[j] && Xuoi[ i - j + n ] && Nguoc[i + j -1]){
            X[i] = j;
            A[j] = 0;              //false
            Xuoi[i - j + n] = 0;       //false
            Nguoc[i + j - 1] = 0;      //false
            if(i == n)       InKetQua();
            else             Try(i+1);
            A[j] = 1;
            Xuoi[i - j + n] = 1;       //true
            Nguoc[i + j - 1] = 1;      //true
        }
    }
}
```

Ví dụ 8 Th

n	1	2	3	4	5	6	7	8	9	10	…
Số nghiệm	1	0	0	2	10	4	40	92	352	724	…

```
BackTracking nQueen, nhap n = 5

Nghiem thu  1:  1  3  5  2  4
Nghiem thu  2:  1  4  2  5  3
Nghiem thu  3:  2  4  1  3  5
Nghiem thu  4:  2  5  3  1  4
Nghiem thu  5:  3  1  4  2  5
Nghiem thu  6:  3  5  2  4  1
Nghiem thu  7:  4  1  3  5  2
Nghiem thu  8:  4  2  5  3  1
Nghiem thu  9:  5  2  4  1  3
Nghiem thu 10:  5  3  1  4  2
```

Nội dung Bài 4

1. Giới thiệu bài toán
   - Vét cạn: brute force, exhaustive search
2. Phương pháp sinh
3. Phương pháp quay lui
4. Bài tập

Bài tập 12

12. Sử dụng thuật toán sinh, viết chương trình giải các bài tập sau:

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

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

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

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

12.5. 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 1 liên tiếp.

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

12.7. Liệt kê tất các các dãy con k phần tử của dãy số A_n sao cho tổng các phần tử của dãy con đó đúng bằng B.

12.8. 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.

12.9. 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.

12.10. Giải bài toán n quân hậu.

Bài tập 13

13. Sử dụng thuật toán quay lui, viết chương trình giải các bài tập sau:

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

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

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

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

13.5. 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 1 liên tiếp.

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

13.7. Liệt kê tất các các dãy con k phần tử của dãy số A_n sao cho tổng các phần tử của dãy con đó đúng bằng B.

13.8. 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.

13.9. 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.

13.10. Giải bài toán n quân hậu.

Kết thúc Bài 4

- Câu hỏi và thảo luận