



ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA

TOÁN RỜI RẠC



Khoa Công Nghệ Thông Tin
TS. Nguyễn Văn Hiệu

D
BACH KHOA

N
A
N
G

D BACH KHOA

TOÁN RỜI RẠC

Bài 3: BÀI TOÁN LIỆT KÊ

N
A
N
G

Nội dung

1. GIỚI THIỆU
2. PHƯƠNG PHÁP SINH
3. PHƯƠNG PHÁP QUAY LUI

GIỚI THIỆU

Mục đích

- Liệt kê tất cả các cấu hình có thể có

Bản chất

- Tìm thuật toán:
 - Dữ liệu vào: cấu hình hiện tại
 - Dữ liệu ra: cấu hình kế tiếp.

GIỚI THIỆU

Nguyên tắc

- Không được lặp cấu hình
- Không được bỏ sót cấu hình

Tiếp cận

- Bài toán chưa có cách giải

PHƯƠNG PHÁP SINH

Mục đích

- Giải bài toán bằng liệt kê tất cả cấu hình

Điều kiện

- Xác định được **thứ tự** cần phải liệt kê
 - Biết được cấu hình đầu tiên
 - Biết được cấu hình cuối cùng
- Xây dựng được thuật toán

PHƯƠNG PHÁP SINH

Bản chất

Method(...) {

(1) cấu hình ban đầu;

(2) **stop = islastconfigure(...);**

while (**top==0**) {
 (3) sinh kế_tiếp;
}

}

Chú thích

(1):

(2):

(3):

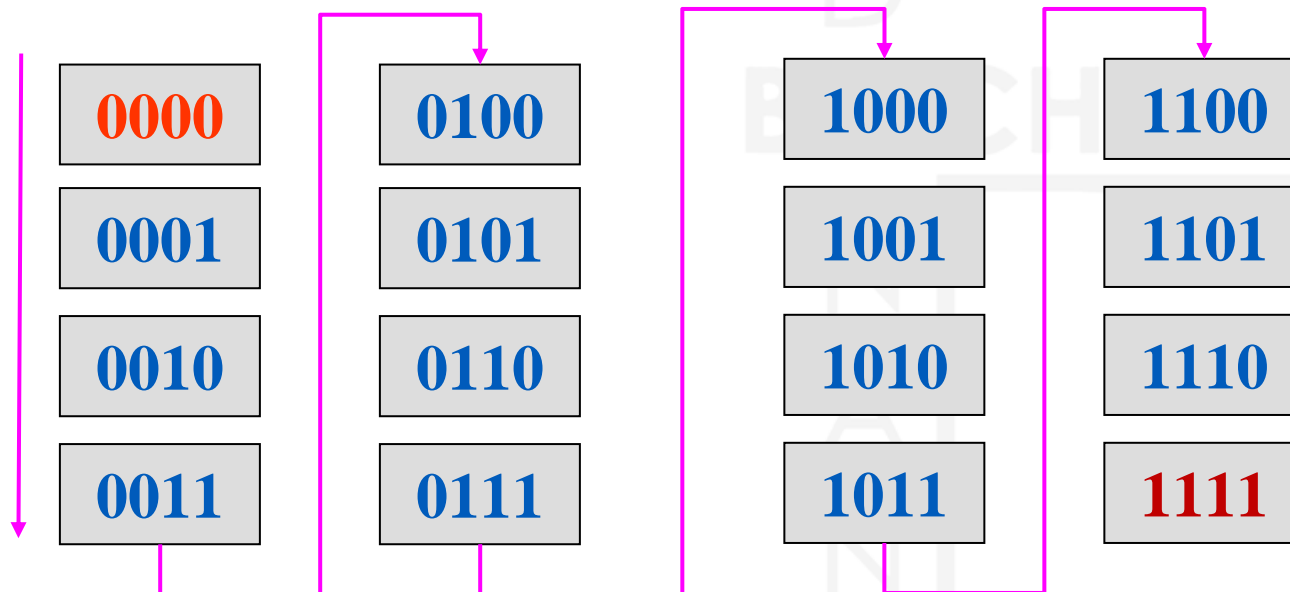
PHƯƠNG PHÁP SINH

(1)- Bài toán liệt kê chuỗi nhị phân có độ dài n

- Bài toán liệt kê chuỗi nhị phân có độ dài n với số bit 0 bằng số bit 1.
- Bài toán liệt kê chuỗi nhị phân có độ dài n với số bit 0 là số chẵn.

PHƯƠNG PHÁP SINH

- Bài toán liệt kê chuỗi nhị phân có độ dài n



PHƯƠNG PHÁP SINH

- Bài toán liệt kê chuỗi nhị phân có độ dài n

Xác định thứ tự cần liệt kê

★ Chuỗi nhị phân được biểu diễn:

$$b = (b_1 \ b_2 \ \dots \ b_n) \text{ với } b_i \in \{0,1\}$$

★ Định nghĩa thứ tự “**từ điển**” :

$$b = (b_1 \ b_2 \ \dots \ b_n) \text{ và } *b = (*b_1 \ *b_2 \ \dots \ *b_n)$$

chuỗi b được liệt kê trước $*b$:

$$q(b) < q(*b)$$

PHƯƠNG PHÁP SINH

- Bài toán liệt kê chuỗi nhị phân có độ dài n

Xây dựng thuật toán sinh

| | | |
|------|---|------|
| 0000 | → | 0001 |
| 0001 | → | 0010 |
| 0011 | → | 0100 |
| 0111 | → | 1000 |

★ Tìm i từ bên phải:
 $b_i = 0$.

★ Gán lại $b_i = 1$ và
 $b_j = 0$ với $j > i$.

PHƯƠNG PHÁP SINH

- Bài toán liệt kê chuỗi nhị phân có độ dài n

Demo

PHƯƠNG PHÁP SINH

```
0 #include <stdio.h>
1 int n, stop, count;
2 void Init(char b[]) {
3     count = 1;
4     int i;
5     for(i=1; i<=n; i++) b[i]='0';
6 }
7 void Out(char b[]) {
8     printf("\n%d:=  ", count);
9     int i;
10    for(i=1; i<=n; i++) printf("%c", b[i]);
11 }
12 void Next_bit_string(char b[]) {
13     int i=n;
14     while(i>=1 && b[i]=='1') b[i--]='0';
15     b[i]='1';
16 }
```

PHƯƠNG PHÁP SINH

```

17 int Lastconfigure(char b[]) {
18     int i;
19     for(i=1;i<=n;i++){ if(b[i]=='0') return 0;}
20     return 1;
21 }
22 void Genarate(char b[]) {
23     Init(b); Out(b);
24     stop =Lastconfigure(b);
25     while(stop==0) {
26         count++;
27         Next_bit_string(b); Out(b);
28         stop = Lastconfigure(b);
29     }
30 main() {
31     printf("n = ");scanf("%d",&n);
32     char b[n];
33     Genarate(b);
34     getch();
35 }

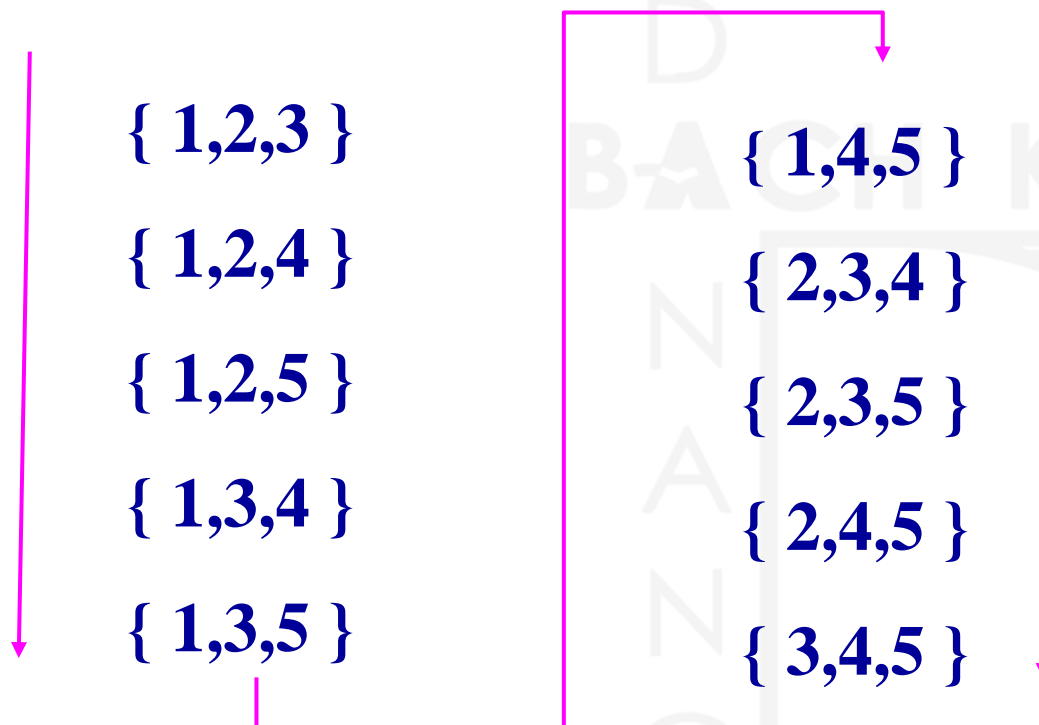
```


PHƯƠNG PHÁP SINH

- (2) - Bài toán liệt kê các tập con của tập n phần tử
- Bài toán liệt kê tập con k phần tử của tập n phần tử
 - Bài toán liệt kê tổ hợp không lặp chập k của tập n phần tử

PHƯƠNG PHÁP SINH

(2) - Bài toán liệt kê các tập con của tập n phần tử



PHƯƠNG PHÁP SINH

Bài toán liệt kê các tập con của tập n phần tử

- Ánh xạ tập n phần tử vào tập X :

$$X = \{1, 2, \dots, n\}$$

- Một tập con k phần tử - **bộ k thành phần có thứ tự**

$$a = (a_1 \ a_2 \ a_3 \ \dots \ a_k)$$

$$1 \leq a_1 < a_2 < \dots < a_k \leq n, \quad a_i \in X.$$

PHƯƠNG PHÁP SINH

- Bài toán liệt kê các tập con của tập n phần tử

Xác định thứ tự cần liệt kê

★ Tập k phần tử được biểu diễn tập **có thứ tự**

$$b = (b_1 \ b_2 \ \dots \ b_n): b_i \in X$$

$$a = (a_1 \ a_2 \ \dots \ a_k): a_i \in X$$

★ Định nghĩa thứ tự từ điển:

$$a = (a_1 \ a_2 \ \dots \ a_k) \text{ và } b = (b_1 \ b_2 \ \dots \ b_k)$$

tập a liệt kê trước tập b : tồn tại j

$$a_1 = b_1, \dots, a_{j-1} = b_{j-1}, \mathbf{a_j < b_j}$$

PHƯƠNG PHÁP SINH

- Bài toán liệt kê các tập con của tập n phần tử

Xác định thuật toán sinh



$\{3, 4, 5\}$

$\{1, 4, 5\}$



$\{2, 4, 5\}$

B1: Tìm vị trí i đầu tiên từ bên phải của dãy:

$$a_i \neq n - k + i.$$

B2: Thay a_i bởi $a_i + 1$.

B3: Thay a_j bởi $a_i - i + j$,
với $j = i+1, \dots,$

PHƯƠNG PHÁP SINH

- Bài toán liệt kê các tập con của tập n phần tử

Xác định thuật toán sinh

B1:

$i=k;$
while ($a[i]==n-k+i$) $i--$;

B2:

$a[i]= a[i] + 1;$

B3:

for ($j=i+1; j \leq k; j++$)
 $a[j]= a[i]+ j -i;$

B1: Tìm vị trí i đầu tiên từ
bên phải của dãy:

$$a_i \neq n-k+i.$$

B2: Thay a_i bởi $a_i + 1$.

B3: Thay a_j bởi $a_i - i + j$,
với $j = i+1, \dots,$

PHƯƠNG PHÁP SINH

- Bài toán liệt kê các tập con của tập n phần tử

Demo

PHƯƠNG PHÁP SINH

```
0 #include <stdio.h>
1 int n, stop;
2 void Init(int a[], int k) {
3     int i;
4     for(i=1; i<=k; i++) a[i]=i;
5 }
6 void Out(int a[], int k) {
7     int i; printf("\n");
8     for(i=1; i<=k; i++) printf("%d ", a[i]);
9 }
10 int lastconfigure(int a[], int k)
11 {
12     int i;
13     for(i=k; i>=1; i--) if(a[i]!=n-k+i) return 0;
14     return 1;
15 }
```

PHƯƠNG PHÁP SINH

```

16 void next_combination(int a[],int k) {
17     int i=k;
18     while(i>=1&&a[i]==n-k+i) i--;
19     a[i]++;
20     int j;
21     for(j=i+1;j<=k;j++) a[j]=a[i]+j-i;
22 }
23 void genarate(int a[],int k) {
24     Init(a,k); Out(a,k);|
25     stop = lastconfigure(a,k);
26     while(stop==0) {
27         next_combination(a,k); Out(a,k);
28         stop = lastconfigure(a,k);
29     } }
30 main() {
31     int k; printf("pt me n:=");scanf("%d",&n);
32     printf(" pt con k:=");scanf("%d",&k);
33     int a[k]; genarate(a,k); getch();
34 }
    
```

(3) - Bài toán liệt kê hoán vị không lặp của tập n phần tử

(3) - Bài toán liệt kê hoán vị không lặp của tập n phần tử

PHƯƠNG PHÁP SINH

(3) - Bài toán liệt kê hoán vị không lặp của tập n phần tử

{1,2,3,4}

{1,2,4,3}

{1,3,2,4}

{1,3,4,2}

{1,4,2,3}

{1,4,3,2}

{2,1,3,4}

{2,1,4,3}

{2,3,1,4}

{2,3,4,1}

{2,4,1,3}

{2,4,3,1}

{3,1,2,4}

{3,1,4,2}

{3,2,1,4}

{3,2,4,1}

{3,4,1,2}

{3,4,2,1}

{4,1,2,3}

{4,1,3,2}

{4,2,1,3}

{4,2,3,1}

{4,3,1,2}

{4,3,2,1}

PHƯƠNG PHÁP SINH

(3) - Bài toán liệt kê hoán vị không lặp của tập n phần tử

Xác định thứ tự cần liệt kê

★ Một hoán vị của X - bộ có thứ tự:

$$a = (a_1 a_2 \dots a_n): a_i \in X, a_p \neq a_q, p \neq q.$$

★ Định nghĩa thứ tự từ điển:

$$a = (a_1 a_2 \dots a_k) \text{ và } b = (b_1 b_2 \dots b_k)$$

tập a liệt kê trước tập b : tồn tại j

$$a_1 = b_1, \dots, a_{j-1} = b_{j-1}, a_j < b_j$$

PHƯƠNG PHÁP SINH

(3) - Bài toán liệt kê hoán vị không lặp của tập n phần tử

Xác định thuật toán sinh



{4, 3, 2, 1}



{1, 3, 4, 2}



{1, 4, 2, 3}

B1: Tìm Right \rightarrow Left, j đầu tiên:

$$a_j \leq a_{j+1}$$

B2: Tìm Right \rightarrow Left, k đầu tiên:

$$a_k > a_j$$

B3: hoán vị a_j và a_k

B4: Lật ngược đoạn a_{j+1} đến a_n

PHƯƠNG PHÁP SINH

(3) - Bài toán liệt kê hoán vị không lặp của tập n phần tử

```
B1: int j = n-1;  
    while(a[j]>a[j+1])j--;  
B2: int k = n;  
    while(a[j]>=a[k])k--;  
B3: int temp = a[j];  
    a[j] = a[k]; a[k] = temp;  
B4: int l = j+1;int r = n;  
    while(l<r) {  
        int temp = a[l];  
        a[l] = a[r]; a[r] = temp;  
        l++;    r--;  
    }
```

B1: Tìm Right \rightarrow Left, j đầu tiên:

$$a_j \leq a_{j+1}$$

B2: Tìm Right \rightarrow Left, k đầu tiên:

$$a_k > a_j.$$

B3: hoán vị a_j và a_k

B4: Lật ngược đoạn a_{j+1} đến a_n

PHƯƠNG PHÁP SINH

- Bài toán liệt kê hoán vị không lặp của tập n phần tử

Demo

PHƯƠNG PHÁP SINH

```
0 #include <stdio.h>
1 int n, stop, count=1;
2 void Init(int a[]){
3     int i;
4     for(i=1; i<=n; i++) a[i]=i;
5 }
6 void Out(int a[]){
7     printf("\n%d: ", count);
8     count++;
9     int i;
10    for(i=1; i<=n; i++) printf("%d", a[i]);
11 }
12 int isLastconfigure(int a[]){
13     int i;
14     for(i=1; i<=n-1; i++) if(a[i]<a[i+1]) return 0;
15     return 1;
16 }
```

PHƯƠNG PHÁP SINH

```

17 void nextconfigure(int a[]) {
18     int j = n-1;
19     while(j>=1 && a[j]>a[j+1]) j--;
20     int k = n;
21     while(a[j]>a[k]) k--;
22     int temp = a[j];
23     a[j] = a[k];
24     a[k] = temp;
25     int l = j+1; int r = n;
26     while(l<r) {
27         int temp = a[l];
28         a[l] = a[r];
29         a[r] = temp;
30         l++; r--;
31     }
32 }

```


PHƯƠNG PHÁP SINH

```

33 void genarate(int a[]) {
34     Init(a); Out(a);
35     stop = isLastconfigure(a);
36     while(stop==0) {
37         nextconfigure(a); Out(a);
38         stop = isLastconfigure(a);
39     }
40 }
41 main() {
42     printf("pt hoan vi n:= "); scanf("%d", &n);
43     int a[n];
44     genarate(a);
45     getch();
46 }
    
```

Nội dung

1. GIỚI THIỆU
2. PHƯƠNG PHÁP SINH
3. PHƯƠNG PHÁP QUAY LUI

PHƯƠNG PHÁP QUAY LUI

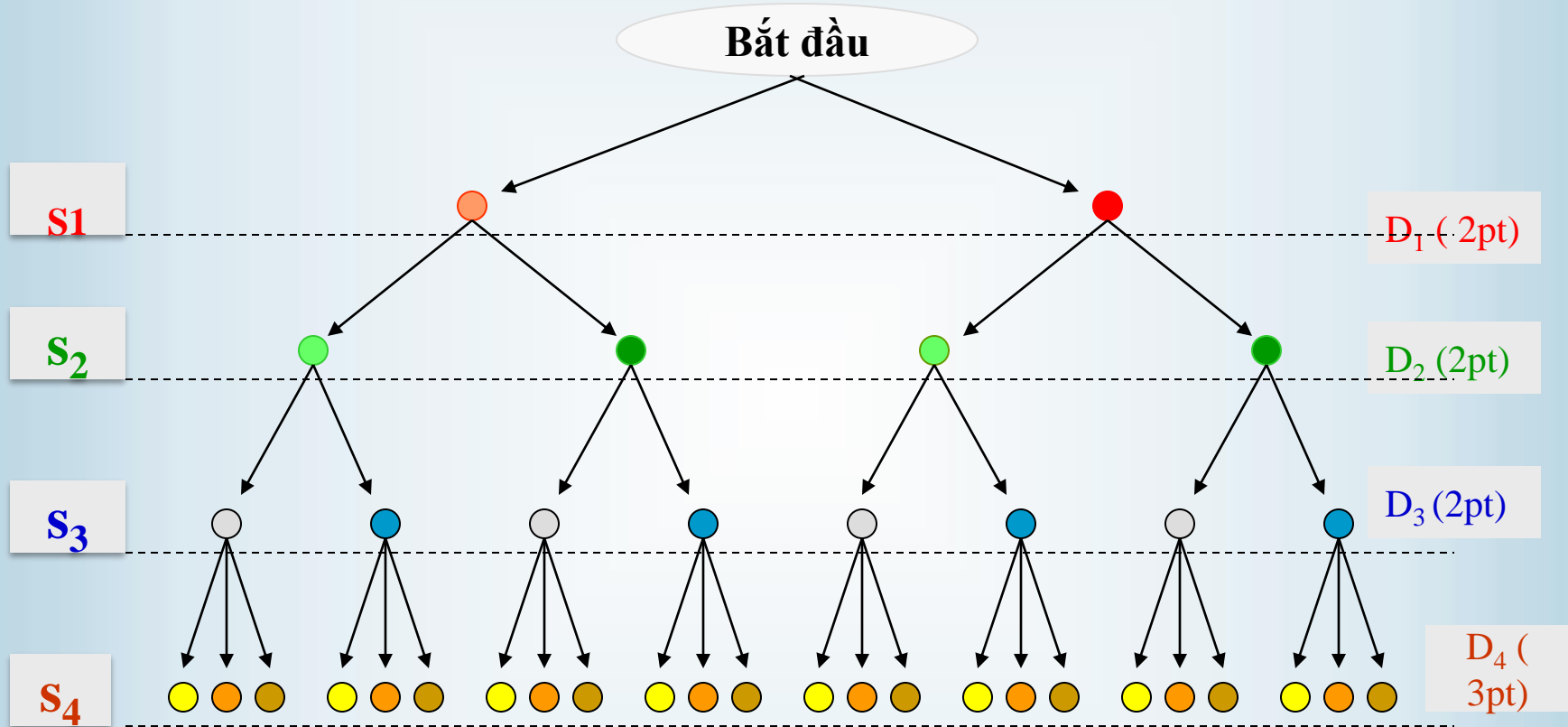
- Mục đích dùng để liệt kê tất cả các cấu hình có thể có
- Các khái niệm liên quan: Backtracking - Поиск с возвратом- Quay lui
- Phương pháp quay lui - dùng để giải bài toán mà nghiệm của bài toán là cấu hình hoặc một loạt cấu hình:
 - Tính chất P: xác định cấu hình
 - Tính chất Q: xác định tính dừng
- Tầm quan trọng của phương pháp:
 - Bài toán mã đi tuần
 - Bài toán xếp hậu
 - Bài toán mê cung
 - Bài toán người giao hàng

PHƯƠNG PHÁP QUAY LUI

```
void try (i, n){  
    (1)  $\forall j \in D_i$   
    (2) if (< chấp nhận j >){  
         $s_i = j$ ;  
        (3) if (i==n)  
            <ghi cấu hình>  
        (4) else try (i+1, n);  
    }  
}
```

- ★ $S = s_1 \dots s_n$
- ★ (1)- Duyệt tất mọi j đề cử cho s_i .
- ★ (2) - chấp nhận j thì $s_i = j$,
- ★ (3)- ghi cấu hình
- ★ (4)- xác định thành phần s_{i+1}

PHƯƠNG PHÁP QUAY LUI



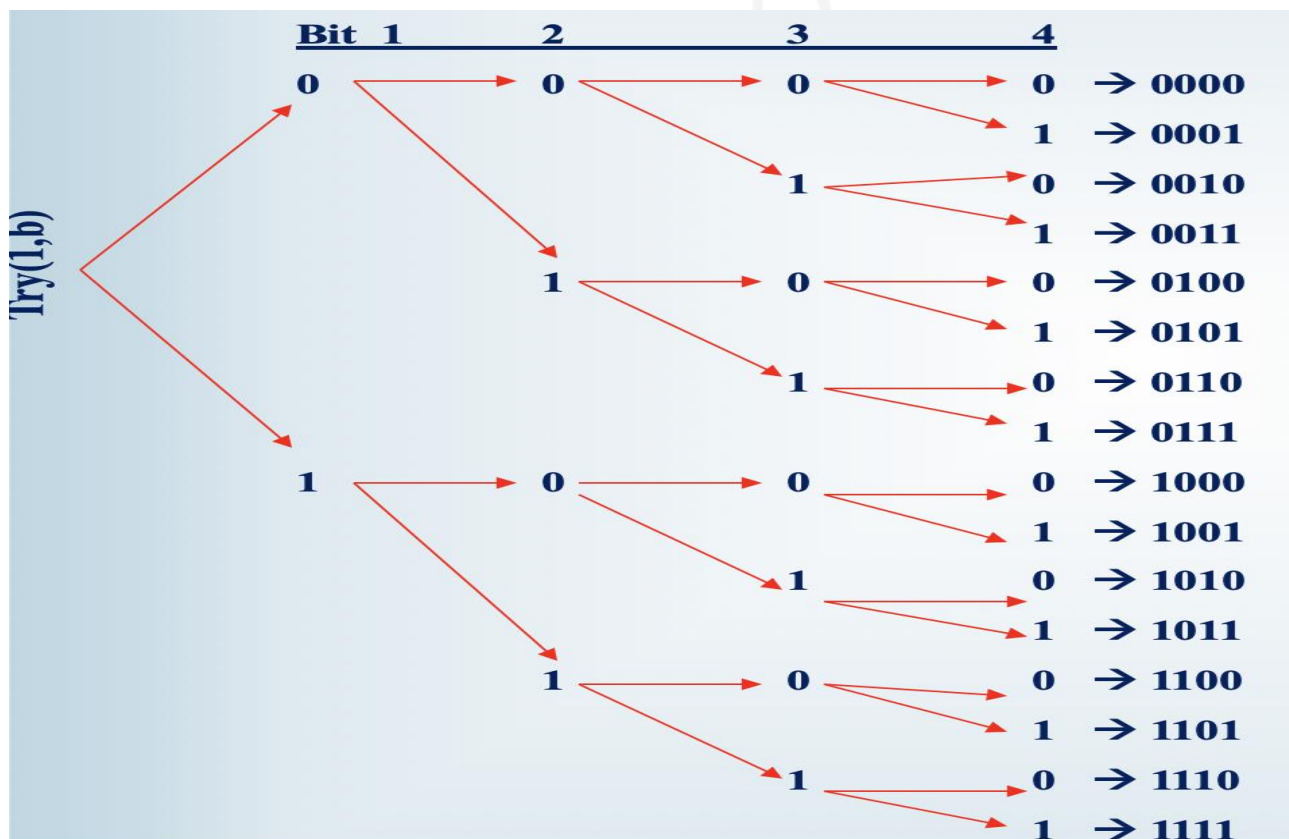
PHƯƠNG PHÁP QUAY LUI

- Bài toán liệt kê chuỗi nhị phân có độ dài n

```
void Try(int i, char b[]){  
    char j;  
    for(j='0'; j<='1';j++){  
        //P – bỏ qua kiểm tra  
        b[i]=j;  
        if(i==n) Out(b);  
        else Try(i+1,b);  
    }  
}
```

PHƯƠNG PHÁP QUAY LUI

- Bài toán liệt kê chuỗi nhị phân có độ dài n



PHƯƠNG PHÁP QUAY LUI

- Bài toán liệt kê chuỗi nhị phân có độ dài n

Demo

PHƯƠNG PHÁP QUAY LUI

```

0 #include <stdio.h>
1 int n, count=1;
2 void Out(char b[]) {
3     printf("\n%d:\t", count);
4     int i; count++;
5     for(i=1; i<=n; i++) putchar(b[i]);
6 }
7 void Try(int i, char b[]) {
8     char j;
9     for(j='0'; j<='1'; j++) {
10         b[i]=j;
11         if(i==n) Out(b);
12         else Try(i+1, b);
13     }
14 }
15 main() {
16     printf("nhap n = "); scanf("%d", &n);
17     char b[n]; printf("xau nhi phan %d bit", n);
18     Try(1, b); getch();
19 }

```

PHƯƠNG PHÁP QUAY LUI

- Bài toán liệt kê hoán vị không lặp của tập gồm n phần tử

```
void Try(int i,int b[],int s[]){  
    int j;  
    for(j=1;j<=n;j++){  
        if(b[j]==1){  
            s[i]=j;  
            b[j]=0;  
            if(i==n) Out(s);  
            else Try(i+1,b,s);  
            b[j]=1;  
        }  
    }  
}
```

PHƯƠNG PHÁP QUAY LUI

```
0 #include <stdio.h>
1 int n, count=1;
2 void Init(int b[]) {
3     int j;
4     for(j=1; j<=n; j++) b[j]=1;
5 }
6 void Out(int s[]) {
7     printf("\n %d:\t", count);
8     int i;
9     for(i=1; i<=n; i++) printf("%d", s[i]);
10    count++;
11 }
```

PHƯƠNG PHÁP QUAY LUI

```

12 void Try(int i,int b[],int s[]) {
13     int j;
14     for(j=1;j<=n;j++) {
15         if(b[j]==1) {
16             s[i]=j;
17             b[j]=0;
18             if(i==n) Out(s);
19             else Try(i+1,b,s);
20             b[j]=1;
21         }
22     }
23 main() {
24     printf("pt hoan vi n:= ");scanf("%d",&n);
25     int b[n],s[n];
26     Init(b); Try(1,b,s);
27     getch();
28 }

```

PHƯƠNG PHÁP QUAY LUI

- Bài toán liệt kê cách xếp n con hậu lên bàn cờ $n \times n$, sao cho các con hậu không thể ăn được nhau.

PHƯƠNG PHÁP QUAY LUI

Minh họa

■ Bài toán xếp hậu

| | | | | | | | |
|---|---|---|--|---|---|---|---|
| | | | | | | | 9 |
| 1 | | | | | | 9 | |
| | 1 | | | | 9 | | |
| | | 1 | | 9 | | | |
| | | | | | | | |
| | | 9 | | 1 | | | |
| | 9 | | | | 1 | | |
| 9 | | | | | | 1 | |

- ✓ Đánh số cột từ 1.. 8
- ✓ Đánh số hàng từ 1.. 8
- ✓ Biểu diễn cách xếp:

$x_1 \ x_2 \ \dots \ x_8$

- ✓ x_i được xếp ở hàng i
- ✓ $D = \{1, \dots, 8\}$
 - ✓ $x_i = j$: Hậu thứ i được xếp vào cột j
- ✓ j được chấp nhận nếu ô (i,j) được tự do

PHƯƠNG PHÁP QUAY LUI

Minh họa

- ✓ Ô (i,j) được tự do
 - ✓ **Quản lý cột**
 - ✓ Quản lý đường chéo thuận
 - ✓ Quản lý đường chéo nghịch

| | | | | | | | | |
|---|---|---|--|---|---|---|---|---|
| | | | | | | | | 9 |
| 1 | | | | | | | 9 | |
| | 1 | | | | | 9 | | |
| | | 1 | | | 9 | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | 9 | | 1 | | | | |
| | 9 | | | | 1 | | | |
| 9 | | | | | | 1 | | |

Quản lý cột

PHƯƠNG PHÁP QUAY LUI

Minh họa

✓ Quản lý đường chéo thuận

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|---|---|---|---|---|---|---|----|
| X_1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| X_2 | | | | | | 8 | | 10 |
| X_3 | | | | | 8 | | | 11 |
| X_4 | | | | 8 | | | | 12 |
| X_5 | | | 8 | | | | | 13 |
| X_6 | | 8 | | | | | | 14 |
| X_7 | 8 | | | | | | | 15 |
| X_8 | | | | | | | | 16 |

Quản lý đường chéo thuận

PHƯƠNG PHÁP QUAY LUI

Minh họa

✓ Quản lý đường chéo nghịch

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|---|---|---|---|---|---|---|----|
| X_1 | | | | | | | | -7 |
| X_2 | 1 | | | | | | | -6 |
| X_3 | | 1 | | | | | | -5 |
| X_4 | | | 1 | | | | | -4 |
| X_5 | | | | 1 | | | | -3 |
| X_6 | | | | | 1 | | | -2 |
| X_7 | | | | | | 1 | | -1 |
| X_8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Quản lý đường chéo nghịch

PHƯƠNG PHÁP QUAY LUI

Minh họa

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------|---|---|---|---|---|---|---|---|
| x_1 | | | | | | | | |
| x_2 | | | | | | | | |
| x_3 | | | | | | | | |
| x_4 | | | | | | | | |
| x_5 | | | | | | | | |
| x_6 | | | | | | | | |
| x_7 | | | | | | | | |
| x_8 | | | | | | | | |

Điều kiện để đặt hậu tại vị trí ô vàng

PHƯƠNG PHÁP QUAY LUI

```
0 #include<stdio.h>
1 int a[100],b[100],c[100],x[100];
2 int count;
3 int init(int n){
4     int i;
5     for(i=1;i<=n;i++) a[i]=1;
6     for(i=2;i<=2*n;i++) b[i]=1;
7     for(i=1-n;i<=n-1;i++) c[i]=1;
8 }
9 int kq(int n){
10    int i;
11    for(i=1;i<=n;i++)printf("%3d", x[i]);
12    printf("\n");
13 }
```

PHƯƠNG PHÁP QUAY LUI

```
14 int try(int i, int n) {  
15     int j;  
16     for (j=1; j<=n; j++) {  
17         if (a[j] && b[i+j] && c[i-j]) {  
18             x[i]=j;  
19             a[j]=0;  
20             b[i+j]=0;  
21             c[i-j]=0;  
22             if (i==n) { kq(i); count++; }  
23             else try(i+1, n);  
24             a[j]=1;  
25             b[i+j]=1;  
26             c[i-j]=1;  
27     } } }
```

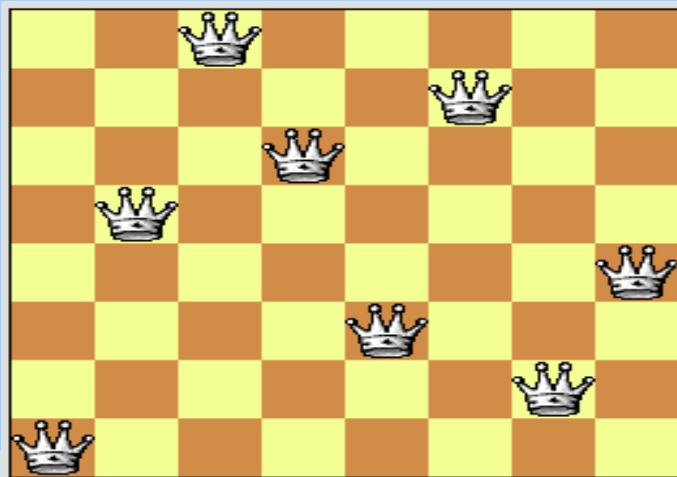

PHƯƠNG PHÁP QUAY LUI

```
33 int main() {  
34     int n;  
35     printf("  n=  ");  
36     scanf("%d", &n);  
37     init(n);  
38     try(1, n);  
39     printf("%d", count);  
40     getch();  
41 }
```

PHƯƠNG PHÁP QUAY LUI

Đọc kết quả

Result



Chương trình

Source Code

$x = (3 \ 6 \ 4 \ 2 \ 8 \ 5 \ 7 \ 1)$

PHƯƠNG PHÁP QUAY LUI

- Liệt kê tổ hợp chập k của n phần tử

- $K = 3, n = 4$

– 1 2 3

– 1 2 4

– 1 3 4

– 2 3 4

- $j = s[i-1] + 1 \dots n - k + i$

- $s[0] = 0$

Bấm để thêm nội dung

- Chương trình

```
void Try(int i, int n, int k, int s[]){
    for(int j = 1; j <= n; j++){
        if(j >= s[i-1]+1 && j <= n-k+i){
            s[i] = j;
            if(i==k) Out(n,k,s);
            else Try(i+1,n,k,s);
        }
    }
}

void TH(int n, int k, int s[]){
    if(k >= 0 && k <= n) s[0] = 0;
    Try(1,n,k,s);
}
```

PHƯƠNG PHÁP QUAY LUI

- Liệt kê chỉnh hợp chập k của n phần tử

- $K = 2, n = 3$

- 1 2
 - 1 3
 - 2 1
 - 2 3
 - 3 1
 - 3 2

- $c[1] = 1, c[2] = 1, c[3] = 1$

- Chương trình

```
void Try(int i, int n, int k, int s[], int c[]){
    for(int j = 1; j <= n; j++){
        if(c[j] == 1){
            s[i] = j;
            c[j] = 0;
            if(i == k) Out(n, k, s);
            else Try(i+1, n, k, s, c);
            c[j] = 1;
        }
    }
}

void TH(int n, int k, int s[], int c[]){
    Test(c, n);
    if(k >= 0 && k <= n) s[0] = 0;
    Try(1, n, k, s, c);
}
```


Tóm lại

□ Kỹ thuật sinh:

- (1) Xác định trạng thái đầu của bài toán.
- (2) Xác định trạng thái kết thúc.
- (3) Xác định một thứ tự cho các trạng thái.
- (4) Tìm giải thuật đi từ trạng thái này sang trạng thái khác.

□ Kỹ thuật quay lui:

- (1) Tại 1 thời điểm, chỉ xét thành phần thứ i của cấu hình
- (2) Với mọi trị j trong miền trị của thành phần này

2.1- Nếu chọn được 1 trị hợp lệ thì

Gán $x_i = j$

Xử lý cấu hình ở thành phần thứ $i+1$

Nếu $i=0$ thì bài toán không có lời giải.

Bài Tập

1. Liệt kê nghiệm nguyên của pt $x_1 + x_2 + x_3 = 15$ với $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$.
2. Liệt kê số chuỗi có độ dài 3 ký tự xyz với $x \in \{a,b,c\}$, $y \in \{d,e\}$, $z \in \{m,n,t\}$
3. Viết lại các bài mẫu về giải thuật quay lui nhưng ghi kết quả lên file.

Bài Tập

Cấu hình ban đầu: trị đầu tiên của mỗi miền trị

Cách sinh: Lấy trị kế tiếp của mỗi miền trị theo cơ chế vòng tròn

Cấu hình cuối: trị cuối cùng của mỗi miền trị

| <u>x</u> | <u>y</u> | <u>z</u> |
|----------|----------|----------|
| a | d | m |
| a | d | n |
| a | d | t |
| a | e | m |
| a | e | n |
| a | e | t |
| b | d | m |
| b | d | n |
| b | d | t |
| b | e | m |
| b | e | n |
| b | e | t |
| c | d | m |
| c | d | n |
| c | d | t |
| c | e | m |
| c | e | n |
| c | e | t |

Dùng thứ tự từ điển để so sánh:

adm < adn



THAT'S ALL; THANK YOU

- **WHAT NEXT?**