

# ĐỒ ÁN

# GIẢI THUẬT & LẬP TRÌNH

ĐỀ 7: Tìm hiểu phương pháp **quay lui** + viết chương trình minh họa một số giải thuật **sắp xếp**

GVHD: Trương Ngọc Châu

SVTH: Nguyễn Quốc Tấn

Lớp 15T2

Nhóm 15.11D

# LỜI MỞ ĐẦU

- Trong quá trình thực hiện đề tài, em không tránh khỏi gặp nhiều thiếu sót, nhưng đã nhận được sự chỉ dẫn, ý kiến đóng góp của quý thầy cô để hoàn thiện đề tài của mình.
- Đồng thời em xin gửi lời cảm ơn chân thành đến thầy **Trương Ngọc Châu** đã giúp em hoàn thành đề tài này.

# I. GIỚI THIỆU ĐỀ TÀI

- *Mô tả: Tìm hiểu phương pháp và ứng dụng vào các bài toán:  $n$ -hậu hòa bình, mã đi tuần, liệt kê*
- các cấu hình tổ hợp, một số thuật toán sắp xếp.
- Dữ liệu vào: Tập chứa dữ liệu nhập của bài toán tương ứng.
- Phương pháp gợi ý: Phương pháp Backtracking, đệ quy,...
- Dữ liệu ra: Tập chứa dữ liệu xuất của bài toán tương ứng.

## II. THUẬT TOÁN

- **A. QUAY LUI**

- **1. Xếp n con hậu**

- Hàm `trienVong()` xác định điều kiện tiếp tục được trình bày như sau:

- *int* `trienVong(int k)`
- {
- *int* `i;`
- *for* (`i=1; i<k; i++`)
- *if* (`s[k]==s[i] || abs(i-k)==abs(s[i]-s[k])`) *return* 0;
- *return* 1;
- }

- Điều kiện tiếp tục là mỗi hàng, mỗi cột và mỗi đường chéo chỉ chứa 1 con hậu.

# 1. Xếp n con hậu

- Hàm xepHau() in vị trí con Hậu ra màn hình

- `int xepHau(int k)`

- {

- `int i;`

- `if (k==n+1)`

- {

- `d1++;`

- `printf("\n%d: ",d1);`

- `for (i=1;i<=n;i++) printf("%d ", s[i]);`

- `printf("\n");`

- }

- `else for (i=1;i<=n;i++)`

- {

- `s[k]=i;`

- `if (trienVong(k)) xepHau(k+1);`

- }

- }

# Một cách khác về 8 hậu

```
• int S[8], a[8], b[15], c[15], sol=0;
• void Try(int i)
• {   int j;
•     for (j=0; j<8; j++)
•         if (a[j]&&b[i-j+7]&&c[i+j])
•         {
•             S[i]=j;
•             a[j]=0;
•             b[i-j+7]=0;
•             c[i+j]=0;
•             if (i==7)
•             {
•                 printf("\n\n%3d: ",++sol);
•                 for (int i=0; i<8; i++)
•                     printf("%d ", 1+S[i]);
•             }
•             else Try(i+1);
•             a[j]=1;
•             b[i-j+7]=1;
•             c[i+j]=1;
•         }
• }
```



# Kết quả khi chạy Dev C++ với n=6

Chon Bai Toan: 8

8.Bai toan Xep Hau nhap tu ban phim bat ki la :  
Nhap n >= 4: 6

1:	2	4	6	1	3	5
2:	3	6	2	5	1	4
3:	4	1	5	2	6	3
4:	5	3	1	6	4	2

## 2. Hoán vị của n

- Hàm in hoán vị ra màn hình

```
int hoanVi(int k)
```

```
{
```

```
•     int i;
```

```
•     if (k==n+1)
```

```
•     {
```

```
•         d2++;
```

```
•         printf("\n%d: ",d2);
```

```
•         for (i=1;i<=n;i++) printf("%d ", s[i]);
```

```
•         printf("\n");
```

```
•     }
```

```
•     else for(i=1;i<=n;i++)
```

```
•     {
```

```
•         s[k]=i;
```

```
•         if (trienVong1(k)) hoanVi(k+1);
```

```
•     }
```

```
• }
```



# Hoán vị = **phương pháp lặp**

- for (i=1; i<=n; i++) S[i]=i; print();
- while (c<n!){
- i=n-1; while (S[i]>S[i+1]) i--;
- j=n; while (S[j]<S[i]) j--;
- tam=S[i]; S[i]=S[j]; S[j]=tam;
- j=i+1; k=n;
- while (j<k)
- { tam=S[j]; S[j]=S[k]; S[k]=tam; j++; k--;}
- print();
- }

## 2. Hoán vị của n

9. Hoán vị của 1 số nhập từ bàn phím bất kì là :  
Nhập n: 3

1:	1	2	3
2:	1	3	2
3:	2	1	3
4:	2	3	1
5:	3	1	2
6:	3	2	1

# 3. Tập con của 1 tập hợp

```
• void tapCon(int k)
• {
•     int i;
•     if(k==n+1){
•         d3++;
•         printf("\n%d: ",d3);
•         printf("{ ");
•         for (i=1;i<=n;i++)
•             if(s[i]==1)
•                 printf("%d ", a[i]);
•         printf("}\n");
•     }
•     else{
•         s[k]=0;tapCon(k+1);
•         s[k]=1;tapCon(k+1);
•     }
• }
```

Độ phức tạp  $O(2^n)$

# 4. Tổ hợp

- Nguyên lý chung
- Để liệt kê tất cả các cấu hình  $S=s_1s_2..s_k$  thuật toán giả sử đã có cấu hình con  $s_1s_2...s_{i-1}$ . Ở bước thử tìm giá trị cho  $s_i$ ,  $Try(i)$ , duyệt qua mọi giá trị  $j$  đề cử được cho  $s_i$  và thực hiện 4 bước sau:
  - a)  $s_i=j$ ;
  - b) *<Thay đổi trạng thái>*;
  - c) Nếu đủ cấu hình ( $i=k$ ) thì Print( $S$ ) ngược lại, gọi đệ quy để thử cho  $s_{i+1}$ ,  $Try(i+1)$ ;
  - d) *<Trả lại trạng thái cũ>*;
- Chương trình chính gọi  $Try(1)$ .

# 4. Tổ hợp

- Chương trình chính gọi Try(1) nên mảng S có S[0]=0.
- void Try(int i)
- {
- int j;
- for (j=1+S[i-1]; j<=n-k+i; j++)
- {
- S[i]=j;
- if (i==k)
- {         printf("\n%3d:", ++c);
- for (int i=1; i<=k; i++) printf("%d ", S[i]);
- }
- else Try(i+1);
- }
- }

# 4. Tổ hợp

- **Phương pháp lặp**
- Liệt kê tổ hợp theo thứ tự tăng ( $s_i < s_{i+1}$ ). Giá trị lớn nhất cho  $s_i$  là  $n-k+i$ .
- `for (i=1; i<=k; i++) S[i]=i; print();`
- `while (c<C(n,k)){`
- `i=k; while (S[i]==n-k+i) i--;`
- `S[i]++;`
- `for (j=i+1; j<=k; j++) S[j]=S[j-1]+1;`
- `print();`
- `}`

# 5. Chỉ hợp không lặp

- Cần mảng **a** đánh dấu các giá trị *j* đã dùng, với ý nghĩa  $a[j] = \text{TRUE} \iff j$  chưa dùng.
- `void Try1(int i)`
- `{`
- `int j;`
- `for (j=1; j<=n; j++) if (a[j])`
- `{`
- `s[i]=j;`
- `a[j]=0;`
- `if (i==k)`
- `{`
- `printf("\n%3d:", ++c1);`
- `for (int i=1; i<=k; i++) printf("%d ", s[i]);`
- `}`
- `else Try1(i+1);`
- `a[j]=1;`
- `}`
- `}`



## 5.Chỉnh hợp không lặp

Chon Bai Toan: 14

14.Chỉnh hợp của 2 số nhập từ bàn phím  $n \geq k$  là :  
Nhập  $n \geq k$ : 4 2

1:	1	2
2:	1	3
3:	1	4
4:	2	1
5:	2	3
6:	2	4
7:	3	1
8:	3	2
9:	3	4
10:	4	1
11:	4	2
12:	4	3

# 6. Chỉnh hợp lặp

- Dùng đệ quy
- void Try2(int i)
- {
- int j;
- for (j=1; j<=n; j++)
- {
- s[i]=j;
- if (i==k)
- {
- printf("\n%3d:", ++c2);
- for (int i=1; i<=k; i++) printf("%d ", s[i]);
- }
- else Try2(i+1);
- }
- }

# 6. Chỉ hợp lặp

- Phương pháp lặp
- `for (i=1; i<=k; i++) S[i]=1; print();`
- `while (c<nk){`
- `i=k; while (S[i]==n) i--;`
- `S[i]++;`
- `for (j=i+1; j<=k; j++) S[j]=1;`
- `print();`
- `}`

# 7. Mã đi tuần

- Hàm kiểm tra vị trí con Mã có đúng hay k
- `int ok(int u, int k)`
- `{`
- `if (u >= 1 && u <= k) return 1;`
- `else return 0;`
- `}`

# 7. Mã đi tuần

- Hàm nhập dữ liệu vị trí con Mã
- void nhapdl()
- {
- printf("\nNhap so hang M cua ban co : ");
- scanf("%d",&m1);
- printf("Nhap so cot N cua ban co : ");
- scanf("%d",&n1);
- do
- {
- printf("Nhap toa do hang ban dau X cua con ma ( $1 \leq X \leq M$ ) : ");
- scanf("%d",&x1);
- }
- while (!ok(x1, m1));
- do
- {
- printf("Nhap toa do cot ban dau Y cua con ma ( $1 \leq Y \leq N$ ) : ");
- scanf("%d",&y1);
- }
- while (!ok(y1, n1));
- }

# 7. Mã đi tuần

- Khởi tạo vị trí =1, biến đếm sc=0
- void khoitao()
  - {
  - sc = 0;
  - daqua[x1][y1] = 1;
  - }
- Hàm in vị trí con Mã đi hết bàn cờ
  - void xuat(int c)
  - {
  - printf("-----\n");
  - printf("Cach thu %d de con ma di het ban co : \n",c);
  - for (int i = 1; i <= m1; i++)
  - {
  - for (int j = 1; j <= n1; j++)
  - printf("%d ",daqua[i][j]);
  - printf("\n\n");
  - }
  - }

# 7. Mã đi tuần

- **Hàm kiểm tra vị trí con Mã**

- `void mxt(int u, int v, int c)`

- `{`

- `int uu, vv;`

- `if (c == m1 * n1)`

- `{`

- `sc++;`

- `xuat(sc);`

- `xuat1(sc);`

- `}`

- `else`

- `{`

- `for (int i = 0; i <= 7; i++)`

- `{`

- `uu = u + h[i][0];`

- `vv = v + h[i][1];`

- `if (ok(uu, m1) && ok(vv, n1))`

- `if (!daqua[uu][vv])`

- `{`

- `daqua[uu][vv] = c + 1;`

- `mxt(uu, vv, c + 1);`

- `daqua[uu][vv] = 0;`

- `}`

- `}`

- `}`

- `}`



# 7. Mã đi tuần

Cách thu 303 de con ma di het ban co :

1      10      15      18      23

16      5      22      9      14

11      2      17      24      19

6      21      4      13      8

3      12      7      20      25

---

Cách thu 304 de con ma di het ban co :

1      10      15      20      23

16      5      22      9      14

11      2      19      24      21

6      17      4      13      8

3      12      7      18      25

---

Vay co tong cong 304 cach de con ma di het ban co!(^\_^)

# 8.Sudoku

- `int isOK(int i, int x) //kiem tra hang, cot, vung`
- `{`
- `int k, t, t1, t2;`
- `int tmpX, tmpY;`
- `//kiem tra hang thu i da co cai nao trung chua`
- `t1= i%(n*n);`
- `t2= i/(n*n);`
- `for (k=(t2*n*n); k<(t2*n*n+n*n); k++)`
- `if (a1[k] == x)`
- `return 0;`
- `//kiem tra cot thu j da co cai nao trung chua`
- `for (k=0; k<n*n; k++)`
- `if (a1[k*n*n+t1] == x)`
- `return 0;`
- `//kiem tra trong o nxn`
- `tmpX = t2%n; tmpY = t1%n;`
- `for (k=t2-tmpX; k<=t2-tmpX+n-1; k++)`
- `for (t=t1-tmpY; t<=t1-tmpY+n-1; t++)`
- `if (a1[k*n*n+t] == x)`
- `return 0;`
- `return 1;`
- `}`

# 8.Sudoku

- `int Try0(int i) //Thu dien so vao o trong`
- `{`
- `int x1;`
- `while (a1[i]!=0)`
- `i++;`
- `for (x1=1; x1<=n*n; x1++)`
- `{`
- `if (isOK(i, x1))`
- `{`
- `a1[i] = x1;`
- `if (i==lastK){`
- `dem++;`
- `printf("\n-----\n");`
- `printf("Cach %d la \n",dem);`
- `xuatfile();`
- `printf("\n");`
- `}`
- `else`
- `Try0(i+1);`
- `a1[i] = 0;`
- `}`
- `}`
- `return 0;`
- `}`

# 8.Sudoku

Chon Bai Toan: 18

Chon De bai 1->6 : 3

De bai:

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

Giai

Cach 1 la

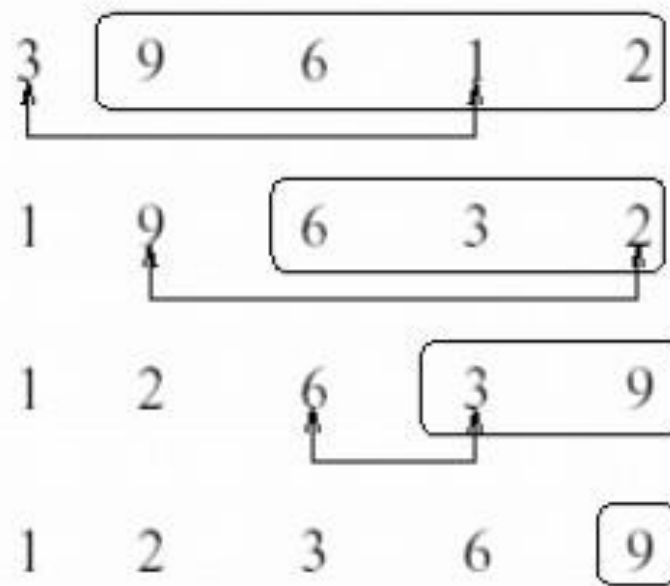
```
2 1 5 9 8 4 3 7 6
4 9 8 3 7 6 2 1 5
6 3 7 5 1 2 9 4 8
7 5 3 8 6 9 1 2 4
1 6 9 2 4 7 5 8 3
8 4 2 1 5 3 6 9 7
9 2 4 6 3 8 7 5 1
5 7 6 4 2 1 8 3 9
3 8 1 7 9 5 4 6 2
```

Vay tong cong co 1 cach giai Sudoku nay

# 9. Sắp xếp chọn

```
• void SelectSort(int *a,int n )
• {
•     int min;                                //min là chỉ số phần tử
    nhỏ nhất
•     for (i = 0; i < n - 1; i++)
•     {
•         int min = i;
•         for (j = i + 1; j < n; j++)
•             if (*(a+min)>*(a+j))
•                 min = j;                    //tìm min trong phần tử
    còn lại
•         swap(*(a+min),*(a+i));            //đổi chỗ nếu tìm thấy
•     }
• }
```

## 9. Sắp xếp chọn



# 10. Sắp xếp nổi bọt

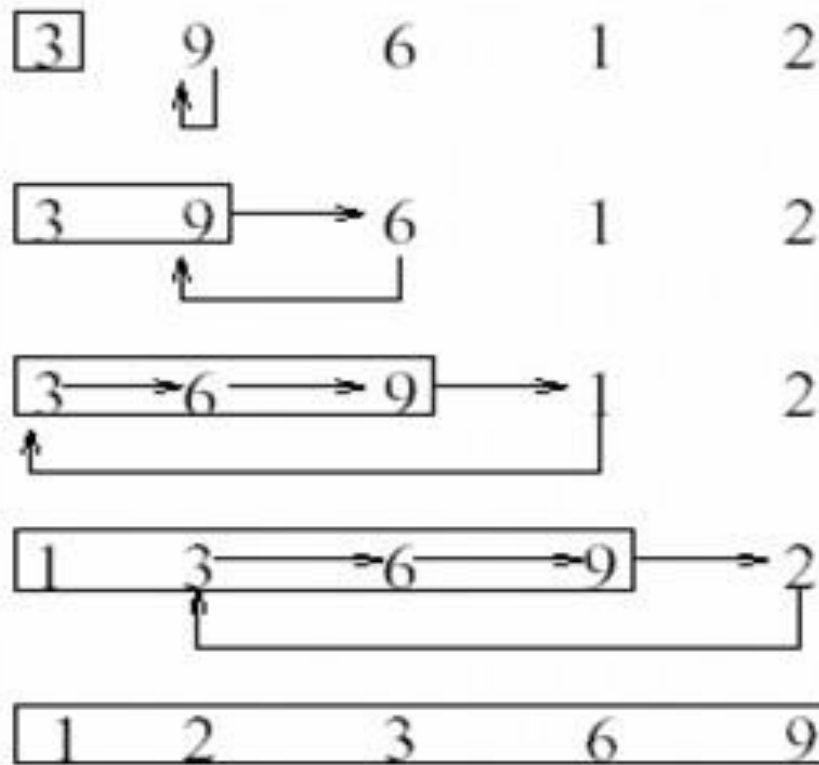
- void BubbleSort(int \*a, int n)
- {
- for (i=0; i<n; i++)             //sắp từ cuối lên đầu
- {
- for (j=n-1; j>i; j--)
- if(\* (a+j)>\* (a+i))
- swap(\* (a+i),\* (a+j));
- }
- }



# 11. SẮP XẾP CHÈN

- void InsertSort (int \*a, int n)
- {
- int t;
- for(i=1;i<n;i++)
- {         j=i-1;
- t=\*(a+i);
- while(t<\*(a+j)&& j>=0)
- {         \*(a+j+1)=\*(a+j);
- j--;
- }
- \*(a+j+1)=t;
- }
- }

# 11. SẮP XẾP CHÈN



# 12. ĐỔI CHỖ TRỰC TIẾP

- void InterchangeSort(int \*a,int n)
- {
- for(i=0;i<n-1;i++)
- for(j=i+1;j<n;j++)
- if(\*(a+j)<\*(a+i))
- swap(\*(a+i),\*(a+j));
- }

# 13 SHELLSORT

```
• void ShellSort(int *a,int n, int *h, int k)
• {
•     int step,i,j, x,len;
•     for (step = 0 ; step <k; step++)
•     {
•         len = *(h+step);
•         for (i = len; i<n; i++)
•         {
•             x = a[i];
•             j = i-len;           //a[j] dung ke truoc a[i] trong cung
day con
•             while ((x<*(a+j)&&(j>=0))) // sap xep day con chua x = pp chen
truc tiep
•             {
•                 *(a+j+len) = *(a+j);
•                 j = j - len;
•             }
•             *(a+j+len) = x;
•             }
•     }
• }
```

# 14 SẮP XẾP NHANH

```
• int Partition(int *a,int l,int r)
• {
•     int p=*(a+l);
•     int i=l+1;
•     int j=r;
•     while(1)
•     {
•         while(*(a+i)<=p&& i<r)
•             ++i;
•         while(*(a+j)>=p&& j>l)
•             --j;
•         if(i>=j)
•         {
•             swap(*(a+j),*(a+l));
•             return j;
•         }
•         else swap(*(a+i),*(a+j));
•     }
• }
```

# 14 SẮP XẾP NHANH

- **Sắp xếp nhanh**
- `void QuickSort(int *a,int l,int r)`
- `{`
- `if(r>l)`
- `{`
- `int p=Partition(a,l,r);`
- `QuickSort(a,l,p-1);`
- `QuickSort(a,p+1,r);`
- `}`
- `}`

# 14 SẮP XẾP NHANH

- Quick sort có độ phức tạp là  $O(n \cdot \log(n))$ , và trong hầu hết các trường hợp Quick sort là thuật toán sắp xếp nhanh nhất, ngoại trừ trường hợp tồi nhất, khi đó Quick sort còn chậm hơn so với Bubble sort.



# SO SÁNH

## Sắp xếp chọn:

- Trung bình đòi hỏi  $n^2/2$  phép so sánh,  $n$  bước đổi chỗ.
- Trường hợp xấu nhất tương tự.

## Sắp xếp chèn:

- Trung bình cần  $n^2/4$  phép so sánh,  $n^2/8$  bước đổi chỗ.
- Xấu nhất cần gấp đôi các bước so với trường hợp trung bình.
- Thời gian là tuyến tính đối với các file hầu như đã sắp và là nhất trong số các thuật toán sắp xếp cơ bản.

## Sắp xếp nổi bọt:

- Trung bình cần  $n^2/2$  phép so sánh,  $n^2/2$  thao tác đổi chỗ.
- Xấu nhất cũng tương tự.

# TÀI LIỆU THAM KHẢO

- Giáo trình Toán rời rạc của thầy Phan Thanh Tao
- Giáo trình Phân tích và Thiết kế Giải Thuật  
Nguyễn Thanh Bình, Phạm Minh Tuấn, Đặng  
Thiên Bình
- Google



CẢM ƠN QUÝ THẦY CÔ &  
CÁC BẠN ĐÃ THEO DÕI LẮNG NGHE