

## Lab 02

### Cài đặt một số lược đồ thuật giải

#### A. Mục tiêu

Tìm hiểu và cài đặt các thuật giải tìm kiếm và sắp xếp trong

#### B. Yêu cầu

- Sinh viên thực tập trong phòng lab.
- Hình thức thu bài: Giáo viên sẽ thu bài (tập tin nén trên) qua hệ thống thu bài trực tuyến tại phòng lab sau khi hết thời gian quy định.

#### C. Ôn tập lý thuyết

##### I. Thuật giải tìm kiếm:

##### 1. Thuật giải tìm kiếm tuyến tính

a. Phát biểu bài toán :

Tìm x có trong dãy a?

- Input :  $a_0, a_2, \dots, a_{n-1}, x$   
**int a[n], x;**
- Output :
  - Nếu có, trả về 1 ;
  - Nếu không có, trả về 0

b. Mô tả thuật giải:

- Bước 1: Xuất phát từ phần tử đầu tiên của dãy: **i=0**
- Bước 2: So sánh  $a[i]$  với giá trị x, có 2 trường hợp:
  - $a[i] = x$ : tìm thấy, dừng thuật giải
  - $a[i] \neq x$ : sang bước 3
- Bước 3: Xét phần tử kế tiếp trong mảng: **i = i+1**
  - nếu  $i > n-1$ : hết mảng, không tìm thấy, dừng thuật giải
  - ngược lại: quay lại bước 2.

b. Cài đặt:

- TH1: Không dùng lính canh

```
int LinearSearch (int a[], int n, int x)
```

```
{
```

```
    int i = 0;
```

```
    while ((i<n) && (a[i])!=x)
```

```
        i++;
```

```
    if (i==n)
```

```

        return 0; // tìm hết mảng nhưng không có x
    return 1; // tìm thấy x tại vị trí x
}

```

- TH2: Dùng lính canh

- Đặt thêm phần tử có giá trị x vào cuối mảng (luôn tìm thấy x trong mảng)
- Dựa vào vị trí tìm thấy x để kết luận.

```

int LinearSearch (int a[], int n, int x)
{
    int i = 0;
    a[n] = x; // đặt phần tử lính canh
    while (a[i]!=x)
        i++;
    if (i==n)
        return 0; // tìm hết mảng nhưng không có x
    return 1; // tìm thấy x
}

```

## 2. Thuật giải tìm kiếm nhị phân

(Chỉ sử dụng cho các dãy đã có thứ tự, giả sử là dãy tăng)

a. Phát biểu bài toán :

Tìm x có trong dãy tăng a?

- Input :  $a_0, a_2, \dots, a_{n-1}, x$   
**int a[n], x;**
- Output :
  - Nếu có, trả về 1 ;
  - Nếu không có, trả về 0

b. Ý tưởng (chia để trị) :

Giả sử dãy đã có thứ tự tăng :  $i < j \Rightarrow a_i \leq a_j$

- Nếu  $x > a_k$  thì x chỉ có thể xuất hiện trong đoạn  $[a_{k+1}, a_{n-1}]$

- Nếu  $x < a_k$  thì  $x$  chỉ có thể xuất hiện trong đoạn  $[a_0, a_{k-1}]$

c. Mô tả thuật giải:

- Bước 1:  $left=0$ ;  $right = n-1$ ; // tìm trên tất cả các phần tử
- Bước 2:  $mid = (left+right)/2$ ; // lấy mốc so sánh  
So sánh  $a[mid]$  với giá trị  $x$ , có 3 trường hợp:
  - $a[mid] = x$ : tìm thấy, trả về 1, dừng thuật giải
  - $a[mid] > x$ :  $right = mid-1$  // tìm tiếp trong dãy con  $a_{left} .. a_{mid-1}$
  - $a[mid] < x$ :  $left = mid+1$  // tìm tiếp trong dãy con  $a_{mid+1} .. a_{right}$
- Bước 3:
  - nếu  $left \leq right$ : lặp lại bước 2 // còn phần tử chưa xét, tìm tiếp
  - ngược lại: trả về 0, dừng // đã xét hết mọi phần tử

d. Cài đặt:

int BinarySearch (int a[], int n, int x)

```
{
    int left = 0, right = n-1, mid;
    do
    {
        mid = (left+right)/2;
        if (x==a[mid])
            return 1; // tìm thấy x tại vị trí mid
        else
            if (x<a[mid])
                right = mid -1;
            else
                left = mid+1;
    }
    while (left<=right);
    return 0; // tìm hết dãy mà không có x
}
```

Có thể cài đặt đệ quy như sau :

int BinarySearch\_R (int a[],int x,int l, int r)

```

{
    int mid;
    if ( l > r )
        return 0;
    mid = (l+r)/2;
    if ( x == a[mid] )
        return 1;
    else
        if ( x > a[mid] )
            return BinarySearch_R (a,x,mid+1,r);
        else
            return BinarySearch_R (a,x,l,mid-1);
}

```

### 3. Thuật giải tìm kiếm nội suy

Hoạt động như tìm kiếm nhị phân, chỉ khác việc chọn chỉ số mid như sau :

$$mid = left + \frac{x - a[left]}{a[right] - a[left]} * (right - left);$$

- Cài đặt thuật toán :

```

int TimNoiSuy(int a[MAX], int n, int x)
{
    int left = 0, right = n - 1, mid;
    while (left <= right)
    {
        mid = left + (right - left) * (x - a[left]) / (a[right] - a[left]);

        if (a[mid] == x)
            return 1;
        else

```

```

        if (a[mid] > x)
            right = mid - 1;
        else
            left = mid + 1;
    }
    return 0; // không tìm thấy
}

```

## II. Thuật giải sắp xếp trong:

### a. Phát biểu bài toán :

Cho dãy  $a[0, \dots, n-1]$  gồm  $n$  số, sắp dãy này tăng dần.

- Input :  $a_0, a_2, \dots, a_{n-1}$   
**int a[n];**
- Output :
  - a tăng dần

**Ghi chú :** Tổng quát, sắp tăng một mảng cấu trúc theo một trường dữ liệu nào đó (có thứ tự).

### b. Các thuật giải sắp xếp trong:

1. Chọn trực tiếp (Selection Straight sort)
2. Chèn trực tiếp (Insertion Straight sort)
3. Đổi chỗ trực tiếp (Interchange Straight sort)
4. Nổi bọt (Bubble sort)
5. Chèn nhị phân (Binary Insertion Sort)
6. Quick sort
7. Heap Sort
8. Merge Sort
9. Radix Sort

### c. Ý tưởng các thuật giải :

#### 1. Chọn trực tiếp (Selection Straight sort)

Thuật giải tiến hành trong  $n-1$  bước, Với mọi  $i = 0, \dots, n-2$ , tìm min trong đoạn  $[a_i, a_{n-1}]$  rồi đưa Min về đầu dãy  $a_i, a_{i+1}, \dots, a_{n-1}$

#### 2. Chèn trực tiếp (Insertion Straight sort):

Thuật giải tiến hành trong  $n-1$  bước, Với mọi  $i = 1, \dots, n-1$ , tìm vị trí thích hợp của dãy con tăng dần  $a_0, a_{i+1}, \dots, a_{i-1}$  để chèn  $a[i]$  vào tạo ra dãy con  $i+1$  phần tử  $a_0, a_{i+1}, \dots, a_i$  tăng dần.

Tìm vị trí thích hợp này bằng thuật giải tìm kiếm tuyến tính từ vị trí  $i-1$  về 0.

### 3. Đổi chỗ trực tiếp (Interchange Straight sort):

Thuật giải tiến hành trong  $n-1$  bước, Với mọi  $i = 0, \dots, n-2$ , tìm các phần tử sau  $a_i$  và tạo với  $a_i$  thành một cặp nghịch thế, rồi triệt tiêu các cặp nghịch thế này.

### 4. Nổi bọt (Bubble sort):

Thuật giải tiến hành trong  $n-1$  bước, Với mọi bước  $i = 0, \dots, n-2$ , xuất phát từ cuối dãy đổi chỗ các cặp phần tử kế cận để đưa phần tử nhỏ nhất về đầu dãy.

### 5. Chèn nhị phân (Binary Insertion Sort):

Thuật giải tiến hành trong  $n-1$  bước, Với mọi  $i = 1, \dots, n-1$ , tìm vị trí thích hợp của dãy con tăng dần  $a_0, a_{i+1}, \dots, a_{i-1}$  để chèn  $a[i]$  vào tạo ra dãy con  $i+1$  phần tử  $a_0, a_{i+1}, \dots, a_i$  tăng dần.

Tìm vị trí thích hợp này bằng thuật giải tìm kiếm nhị phân

### 6. Thuật giải Quick sort:

Cải tiến thuật giải đổi chỗ trực tiếp bằng cách đổi chỗ các cặp không đúng thứ tự có thể xa nhau. Thuật toán thực hiện việc lặp phân đoạn mảng đã cho thành 2 phần, nửa mảng bên trái  $\leq x$ , nửa mảng bên phải  $\geq x$ , cho đến khi các mảng con chỉ có 1 phần tử.

### 7. Thuật giải Heap Sort:

Cải tiến thuật giải chọn trực tiếp bằng cách tại mỗi bước, tận dụng thông tin các phép toán so sánh ở bước trước.

Từ heap ban đầu phát triển thành heap đầy đủ ban đầu, hoán vị 2 phần tử đầu và cuối để đưa giá trị lớn nhất của dãy về cuối dãy.

Loại phần tử cuối (giá trị lớn nhất), và phần tử đầu  $a_0$ .

Các phần tử còn lại là heap, bổ sung  $a_0$  vào heap này để tạo thành một heap nhiều hơn 1 phần tử, rồi lặp lại các công việc trên để đưa giá trị lớn nhất về cuối.

Lặp quá trình trên đến khi chỉ còn 1 phần tử.

### 8. Thuật giải Merge Sort:

Thuật giải tiến hành nhiều bước lặp, mỗi bước gồm 2 giai đoạn :

- Tách luân phiên  $p$  phần tử của dãy đã cho vào 2 dãy trung gian.
- Trộn từng  $p$  phần tử của 2 dãy trung gian để tạo ra  $2p$  phần tử tăng dần rồi lưu vào dãy ban đầu.
- $p$  khởi tạo ban đầu bằng 1. Tại mỗi bước  $p$  khởi tạo lại bằng  $2p$ .

Thuật giải kết thúc khi  $p > n$ .

## 9. Thuật giải Radix sort:

Mô phỏng cách phân phối thư của bưu điện.

**D. Bài tập thực hành****Bài 1: Sinh viên tạo project với tên: MSSV\_HoVaTen\_Lab02\_Bai1.**

**Viết chương trình tìm kiếm phần tử x trong dãy số a, các thuật giải:**

0. Thoát khỏi chương trình
1. Đọc mảng từ file.
2. Xuất mảng.
3. Tìm kiếm tuyến tính trả về chỉ số tìm thấy.
4. Tìm kiếm tuyến tính có lĩnh canh: trả về 1 (tìm thấy) hoặc 0 (không tìm thấy).
5. Tìm kiếm nhị phân trả về chỉ số tìm thấy.
6. Tìm kiếm nội suy: trả về 1 (tìm thấy) hoặc 0 (không tìm thấy).

- Tập tin dữ liệu "*test1.txt*":

8 //Kích thước mảng

12    2    8    5    1    6    4    15

- Tập tin dữ liệu "*test2.txt*":

12 //Kích thước mảng

7013 8425 1239 428 1424 7009 4518 3252 9170 999 1725 701

**Bài 2: Sinh viên tạo project với tên: MSSV\_HoVaTen\_Lab02\_Bai2.**

**Viết chương trình tùy chọn sắp tăng dãy n số nguyên với các thuật giải:**

0. Thoát khỏi chương trình
1. Đọc mảng từ file.
2. Xuất mảng.
3. Chọn trực tiếp (Selection Straight sort)
4. Chèn trực tiếp (Insertion Straight sort)
5. Đổi chỗ trực tiếp (Interchange Straight sort)
6. Nổi bọt (Bubble sort)
7. Chèn nhị phân (Binary Insertion Sort)
8. Radix

- Tập tin dữ liệu "*test1.txt*":

8 //Kích thước mảng

12    2    8    5    1    6    4    15

- Tập tin dữ liệu "*test2.txt*":

12 //Kích thước mảng

7013 8425 1239 428 1424 7009 4518 3252 9170 999 1725 701

Bài tập 1:

**Thông tin sử dụng cho bài 3 và bài 4:****Giả sử có một danh sách nhân viên, mỗi nhân viên được lưu trữ các thông tin:**

- Mã nhân viên, //Một chuỗi có 7 ký tự, không có ký tự trắng
- Họ nhân viên, //một chuỗi có không quá 10 ký tự
- Tên lót nhân viên, // một chuỗi có không quá 10 ký tự
- Tên nhân viên, // một chuỗi có không quá 10 ký tự
- Ngày tháng năm sinh của nhân viên, dạng : dd/mm/yyyy
- Địa chỉ, // một chuỗi có không quá 15 ký tự
- Lương, //Một số thực dương làm tròn, không quá 7 chữ số

**Danh sách nhân viên được cho trong tập tin văn bản :**

- test1.txt :

```

=====
:Ma NV   :Ho      tLot   Ten      :NTNSinh  :Dia Chi   :Luong   :
=====
:2523480 :Uo       Minh    Duong    :3 /1 /1980:Da_Lat    :15000543 :
:1134547 :Hoang    Hoa      Tra      :12/12/1987:Khanh_Hoa :18183210 :
:2130007 :Truong   Thanh    Hang     :2 /2 /1980:Phu_Yen   :15000543 :
:1212345 :Nguyen   Thi      Hang     :10/8 /1987:Binh_Dinh :20232343 :
:2223453 :Le       Van      Trong    :12/5 /1991:Hue       :18183210 :
=====
:2003205 :Van      Minh    Duong    :10/10/1988:Da_Lat    :18183210 :
:1500348 :Tran     Hoang    Hoa      :15/10/1996:Binh_Dinh :18183210 :
:2134007 :Ly       Thi      Hang     :18/7 /1988:Khanh_Hoa :20232343 :
:2030007 :Dinh     Thai     Duong    :21/12/1980:Da_Lat    :15000543 :
:2011453 :Le       Van      Tam      :25/6 /1988:Hue       :18183210 :
=====
:2022205 :Trinh    Hoai     Duong    :19/10/1991:Da_Lat    :18183210 :
:1652345 :Nguyen   Minh     Tam      :22/9 /1980:Binh_Dinh :20232343 :
=====

```

- test2.txt :

Ma NV	Ho	tLot	Ten	NTNSinh	Dia Chi	Luong
LD12045	Nguyen	Tuan	Vo	1/1/1986	Lam_Dong	2500000
LD13210	Truong	Thi	Hoa	12/12/1985	Ninh_Thuan	3700000
LD13452	Tran	Ngoc	Ninh	2/2/1974	Khanh_Hoa	8000000
LD14432	Nguyen	_	Vo	10/8/1975	Phu_Yen	15000000
LD15332	Le	Thi	Lieu	12/5/1974	Binh_Dinh	12000000
LD22032	Van	Thi	Hoa	10/10/1984	Lam_Dong	6000000
LD22052	Vo	Ngoc	Hoa	1/10/1984	Lam_Dong	7000000
LD22140	Nguyen	Van	Vu	1/7/1990	Binh_Dinh	6200000
LD22145	Nguyen	Thi	Vo	10/6/1986	Khanh_Hoa	8000000
LD23045	Tran	Trong	Hieu	10/10/1991	Ha_Noi	15000000
LD24042	Ly	Van	Hoa	12/9/1983	Ninh_Thuan	30000000
LD30432	Le	_	Vo	12/12/1975	Lam_Dong	12000000

**Bài 3: Sinh viên tạo project với tên: MSSV\_HoVaTen\_Lab02\_Bai3.****Viết chương trình thực hiện các thao tác tìm kiếm trên danh sách nhân viên:**

0. Thoát chương trình.
1. Tạo danh sách nhân viên.
2. Xem danh sách nhân viên.
3. Tìm kiếm theo họ, tên: Xuất tất cả các nhân viên trùng họ và tên cho trước.
4. Tìm kiếm năm sinh: Xuất tất cả các nhân viên cùng năm sinh.
5. Tìm kiếm theo họ, tên và năm sinh: Xuất tất cả các nhân viên trùng họ,tên cho trước và có năm sinh < x



6. Tìm kiếm theo tên và địa chỉ : Xuất tất cả nhân viên cùng tên và địa chỉ cho trước.
7. Tìm kiếm theo năm sinh và lương : Xuất các nhân viên có mức lương  $\geq x$  và có năm sinh  $\leq y$
8. Tìm kiếm nhị phân theo mã nhân viên khi cho trước mã nhân viên (cần phải kiểm tra tính đơn điệu của dữ liệu)

Các chuỗi lưu trữ thông tin về họ, tên lót, tên, địa chỉ có thể gồm nhiều từ, các từ được nối với nhau bởi dấu gạch dưới.

Về tên lót của nhân viên, nếu không có sẽ được thay thế bằng ký tự ‘\_’.

#### **Bài 4: Sinh viên tạo project với tên: MSSV\_HoVaTen\_Lab02\_Bai4.**

**Viết chương trình thực hiện các thao tác sắp xếp trên danh sách nhân viên:**

0. Thoát chương trình.
1. Tạo danh sách nhân viên.
2. Xem danh sách nhân viên.
3. Dùng thuật giải chọn trực tiếp sắp danh sách tăng dần theo mã NV.
4. Dùng thuật giải chèn trực tiếp sắp danh sách tăng dần theo địa chỉ.
5. Dùng thuật giải đổi chỗ trực tiếp sắp danh sách giảm dần theo tiền lương.
6. Dùng thuật giải chèn nhị phân sắp danh sách giảm dần theo mã NV.
7. Dùng thuật giải Heap sort sắp danh sách tăng dần theo tiền lương.
8. Dùng thuật giải Quick sort sắp danh sách tăng dần theo năm sinh