

Môn: Cấu trúc dữ liệu và thuật giải 1

Bài thực hành 1:

““Các thuật giải tìm kiếm trong””

I. Mục tiêu

II. Nội dung ôn tập

III. Luyện tập

I. Mục tiêu

Cài đặt các thuật giải tìm kiếm trong

II. Nội dung ôn tập: Tạo thư mục, đặt tên là MaSV_Lab01 để lưu bài làm

- Thuật giải tìm kiếm tuyến tính
- Thuật giải tìm kiếm nhị phân

1. Thuật giải tìm kiếm tuyến tính

a. Phát biểu bài toán :

Tìm x có trong dãy a?

- Input : $a_0, a_1, \dots, a_{n-1}, x$

int a[n], x;

- Output :

- Nếu có, trả về chỉ số i đầu tiên để $a[i] = x$;
- Nếu không có, trả về -1

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 -1; // tìm hết mảng nhưng không có x
    return i; // 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 -1; // tìm hết mảng nhưng không có x
    return i; // tìm thấy x ở vị trí i
}

```

2. Thuật giải tìm kiếm nhị phân

(Chỉ sử dụng cho các dãy đã có thứ tự)

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 chỉ số i để $a[i] = x$;
 - Nếu không có, trả về -1

b. Ý tưởng:

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, dừng thuật giải
 - $a[mid] > x$: right = mid-1 // tìm tiếp trong dãy con $a_{left} \dots a_{mid-1}$
 - $a[mid] < x$: left = mid+1 // tìm tiếp trong dãy con $a_{mid+1} \dots a_{right}$
- Bước 3:
 - nếu left ≤ right: lặp lại bước 2 // còn phần tử chưa xét, tìm tiếp
 - ngược lại: 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 mid; // tìm thấy x tại vị trí mid
        else
            if (x<a[mid])
                right = mid -1;
    }
}

```

```

        else
            left = mid+1;
    }
    while (left<=right);
    return -1; // tìm hết dãy mà không có x
}

```

III. Luyện tập

Bài 1:

Tìm kiếm số nguyên x trên dãy n số nguyên a[0..n-1]:

- Nếu không có, trả về -1;
- Nếu có, trả về các trường hợp sau:
 - Chỉ số đầu tiên
 - Chỉ số cuối cùng
 - Các chỉ số tại các phần tử trong dãy trùng x.

Sử dụng thuật giải tìm kiếm tuyến tính.

Ta sẽ tạo một Project gồm 2 tập tin:

- Tập tin thư viện *h: Chứa các hàm chức năng của chương trình
- Tập tin chương trình *cpp: Chứa hàm main(), các hàm tổ chức menu, nhập xuất dữ liệu.

```

//Tập tin thư viện: h_TK_Day.h
//Chen cac tap tin thu vien can thiet
#include<iostream>
using namespace std;
//Định nghĩa hằng
#define MAX 1000
//Khai báo nguyên mẫu
int LinearSearch_First(int a[MAX], int N, int x);
int LinearSearch_First_(int a[MAX], int N, int x); //Co linh canh
int LinearSearch_End(int a[MAX], int N, int x);
void LinearSearch_All(int a[MAX], int N, int x);
//Định nghĩa các hàm
//Tim x co trong a? Khong co tra ve -1; neu co tra ve chi so i DT de a[i] = x;
//Khong co linh canh
int LinearSearch_First(int a[MAX], int N, int x)
{
    int i=0;
    while ((i<N) && (a[i]!=x)) i++;
    if (i==N)
        return -1;
    return i;
}
//Co linh canh
int LinearSearch_First_(int a[MAX], int N, int x)
{
    int i=0;
    a[N] = x;

```

```

        while (a[i]!=x )
            i++;
        if (i==N)
            return -1;
        return i;
    }
//Tim x co trong a? Khong co tra ve -1; neu co tra ve chi so i CC de a[i] = x;
//Khong co linh canh
int LinearSearch_End(int a[MAX], int N, int x)
{
    int    i=N-1;
    while ((i >= 0) && (a[i]!=x ))
        i--;
    return i;
}

//Xuat tat ca cac i lam cho a[i] ==x
void LinearSearch_All(int a[MAX], int N, int x)
{
    int    i, Dau = 0;
    for(i = 0; i < N; i++)
        if(a[i] == x)
        {
            Dau = 1;
            cout<<i<<"\t";
        }
    if (Dau == 0)
        cout<<"\n"<<x<<" khong co trong mang a!!!";
}

//Tap tin chuong trinh : Cpp_TKDay.cpp
//Chen cac tttv
#include<Z:\TK_Day\TK_Day\h_TKDay.h>
#include<stdlib.h>
//. . .
//Khai bao nguyen mau
void Nhap(int a[MAX], int N);
void Xuat(int a[MAX], int N);
void Menu();
int ChonMenu();
void XulyMenu(int a[MAX], int N, int x, int Chon);
void main()
{
    int a[MAX], N, x, Chon;
    cout<<"\nNhap N = ";
    cin>>N;

```

```

        cout<<"\nNhap du lieu cho mang a:\n";
        Nhap(a,N);
        cout<<"\nNhap khoa tim kiem x = ";
        cin>>x;
        do
        {
            Chon = ChonMenu();
            XulyMenu(a,N,x, Chon);
        }
        while(1);
    }

//Dinh nghi cac ham
void Menu()
{
    cout<<"\n-----Tim kiem tuyen tinh-----";
    cout<<"\n1. Tra ve chi so i dau tien neu co.";
    cout<<"\n2. Tra ve chi so i dau tien neu co (Co linh canh).";
    cout<<"\n3. Tra ve chi so i Cuoi cung neu co.";
    cout<<"\n4. Tra ve tat ca chi so i cuoi cung neu co.";
    cout<<"\n9. Thoat khoi chuong trinh.";
}

void Nhap(int a[MAX], int N)
{
    //...
}

void Xuat(int a[MAX], int N)
{
    //...
}

int ChonMenu()
{
    //...
}

void XulyMenu(int a[MAX], int N, int x, int Chon)
{
    int Kq;
    switch(Chon)
    {
        case 1:
            cout<<"\n1. Tra ve chi so so i dau tien neu co.";
            cout<<"\nDay so nhap:\n";

```

```

        Xuat(a, N);
        Kq = int LinearSearch_First (a,N,x);
        if(Kq == -1)
            cout<<"\n"<<x<<" không có trong dãy a!";
        else
            cout<<"\nChỉ số của pt đầu tiên == "<<x<<" là: "<<Kq;
        cout<<"\n";
        break;
    case 2:
        cout<<"\n2. Tra về chỉ số số i đầu tiên nếu có (có lĩnh canh)";
        cout<<"\nDay số nhập:\n";
        Xuat(a, N);
        Kq = LinearSearch_First_(a,N,x);
        if(Kq == -1)
            cout<<"\n"<<x<<" không có trong dãy a!";
        else
            cout<<"\nChỉ số của pt đầu tiên == "<<x<<" là: "<<Kq;
        cout<<"\n";
        break;

    case 3:
        cout<<"\n3. Tra về chỉ số số i Cuối cùng nếu có.";
        cout<<"\nDay số nhập:\n";
        Xuat(a, N);
        Kq = LinearSearch_End(a,N,x);
        if(Kq == -1)
            cout<<"\n"<<x<<" không có trong dãy a!";
        else
            cout<<"\nChỉ số của pt cuối cùng == "<<x<<" là: "<<Kq;
        cout<<"\n";
        break;

    case 4:
        cout<<"\n4. Tra về tất cả chỉ số i cuối cùng nếu có.";
        cout<<"\nDay số nhập:\n";
        Xuat(a, N);
        LinearSearch_All(a,N,x);
        cout<<"\n";
        break;

    case 9:
        cout<<"\n9. Thoát khỏi chương trình.\n";
        exit(1);
    }
}

```

Bài 2:

Giả sử có một danh sách sinh viên, mỗi một sinh viên được lưu trữ các thông tin:

- Mã số,
- họ tên,
- lớp,
- điểm trung bình,
- tổng số tín chỉ đã tích lũy được.

Thực hiện các thao tác tìm kiếm trên danh sách sinh viên:

- Tìm kiếm theo mã số
- Tìm kiếm theo họ tên: Xuất tất cả các sinh viên nếu họ tên trùng với họ tên cho trước.
- Tìm kiếm theo điểm trung bình : Xuất tất cả sinh viên có điểm $\geq x$.
- Tìm kiếm theo lớp : Xuất sinh viên thuộc lớp cho trước.

Ta sẽ định nghĩa kiểu cấu trúc chứa các thông tin sinh viên tên là SINHVIEN. Danh sách sinh viên sẽ được tổ chức bằng mảng 1 chiều các cấu trúc kiểu SINHVIEN. Việc thực hiện nhập, xuất, tìm kiếm sẽ thực hiện trên mảng các cấu trúc kiểu SINHVIEN

Ta sẽ tạo một Project gồm 2 tập tin:

- Tập tin thư viện *h: Chứa các hàm chức năng của chương trình
- Tập tin chương trình *cpp: Chứa hàm main(), các hàm tổ chức menu, nhập xuất dữ liệu.

```
//Tập tin thư viện: h_Tk.h
//Chèn các tập tin thư viện cần thiết
#include<iostream>
#include<string.h>
#include<iomanip>
#include<stdio.h>

using namespace std;
//Định nghĩa hằng
#define MAX 1000
//Định nghĩa kiểu dữ liệu
struct SINHVIEN
{
    char Maso[10];
    char HoTen[20];
    char Lop[8];
    double Dtb;
    int Tichluy;
};

//Biến toàn cục
SINHVIEN Sv[MAX];
int Sct = 0;

//Khai báo nguyên mẫu các hàm
void Chen_Ct(char Maso[10],char HoTen[20],char Lop[8],double Dtb, int Tichluy);
int Tim_Maso (char Maso[10]);
void Tim_Hoten (char Ht[20]);
void Tim_Diem (double Cd, double Ct);
void Tim_Lop (char Lop[8]);
```

```

//Tim theo ma so: tra ve chi so i ĐT sao cho sv[i].Maso = Maso
int Tim_Maso (char Maso[10])
{
    int i=0;
    while ((i < Sct) && (strcmp(Sv[i].Maso, Maso)))
        i++;
    if (i == Sct)
        return -1;
    return i;
}

//Tim theo Ho ten: neu co xuất tat ca
void Tim_Hoten (char Ht[20])
{
    int i, k=0;
    for (i = 0; i < Sct; i++)
        if(!strcmpi(Sv[i].HoTen, Ht))//Khong phan biet ky tu thuong, hoa
        {
            Output_Struct(Sv[i]) ;
            k++;
        }
    if (k==0)
        cout<<"\nKhong co sinh vien ho ten nay!";
}

//Tim theo diem
//Xuất sv co diem >=x =0,1,..., 10
void Tim_Diem (double Cd, double Ct)
{
    int i, k=0;
    for (i = 0; i < Sct; i++)
        if(Sv[i].Dtb >=Cd && Sv[i].Dtb <=Ct)
        {
            Output_Struct(Sv[i]) ;
            k++;
        }
    if (k==0)
        cout<<"\nKhong co sinh vien nao!";
}

//Tim theo Lop
//Xuất sv thuoc lop
void Tim_Lop (char Lop[8])
{
    int i, k=0;
    for (i = 0; i < Sct; i++)
        if(!_strcmpi(Sv[i].Lop, Lop))//Khong phan biet ky tu thuong, hoa
        {
            Output_Struct(Sv[i]) ;
            k = 1;
        }
    if (k==0)
        cout<<"\nKhong co sinh vien nao!";
}

```



```

//Tập tin chương trình: Cpp_Tk.cpp
//Chèn các tập tin thư viện cần thiết
#include<\TK_Cautruc\TK_Cautruc\h_Tk.h>

//Khai báo nguyên mẫu
void Menu();
int ChonMenu();
void XL_Menu(int Chon);

void Output();
void Output_Data();
void Output_Struct(SINHVIEN p);

void Setup();

void main()
{
    int Chon;
    Setup();
    do
    {
        Chon = ChonMenu();
        XL_Menu(Chon);
    }
    while(1);
}

//Xuất tên các chức năng của CT ra màn hình
void Menu()
{
    cout<<"\n-----MENU-----";
    cout<<"\n1.Tìm theo mã số";
    cout<<"\n2.Tìm theo họ tên";
    cout<<"\n3.Tìm theo Điểm Trung Bình";
    cout<<"\n4.Tìm theo Lớp";
    cout<<"\n5.Xem danh sách";
    cout<<"\n6.Thoát khỏi chương trình!!!";
}

//Điều khiển việc chọn chức năng của người SD
//int ChonMenu()
//Xử lý CT
void XL_Menu(int Chon)
{
    char Maso[10], Ht[20], Lop[8];
    int Kq;
    int Cd, Ct;
    switch(Chon)
    {
        case 1:
            cout<<"\n1.Tìm theo mã số:\n";
            //...
            break;
    }
}

```

```

        case 2:
            cout<<"\n1.Tim theo ho ten:\n";
            //...
            break;
        case 3:
            cout<<"\n3.Tim theo Diem Trung Binh\n";
            cout<<"\nNhap Can duoi Cd = ";
            cin>>Cd;
            cout<<"\nNhap Can tren Ct = ";
            cin>>Ct;

            flushall();
            cout<<"\n      Danh sach sinh vien      \n";
            Output();
            Output_Data();
            cout<<"\n\nKet qua tim:\n";
            Tim_Diem (Cd,Ct);

            cout<<"\n";
            break;
        case 4:
            cout<<"\n1.Tim theo Lop:\n";
            //...
            break;

        case 5:
            cout<<"\n8.Xem Danh sach:\n";
            cout<<"\n      Danh sach sinh vien      \n";
            Output();
            Output_Data();
            cout<<"\n";
            break;
        case 6:
            cout<<"\n9.Thoat khoi chuong trinh!!!\n";
            exit(1);

    }
}
//Danh sach sinh vien
void Setup()
{
    Chen_Ct("0213345","Nguyen Van","CTK100",4.5,41);
    Chen_Ct("0210340","Truong Van A","CTK101",8.0,39);
    Chen_Ct("0201381","Duong Mai","CTK100",4.5,38);
    //...
}
void Chen_Ct(char Maso[10],char HoTen[20],char Lop[8],double Dtb, int Tichluy)
{
    if (Sct < MAX)
    {
        Strcpy_s(Sv[Sct].Maso, Maso);
        Strcpy_s(Sv[Sct].HoTen,HoTen);
        Strcpy_s(Sv[Sct].Lop,Lop);
        Sv[Sct].Dtb = Dtb;
        Sv[Sct].Tichluy = Tichluy;
    }
}

```

```

        Sct++;
    }
    else
        cout<<"\nVuot kích thước mang!!!";
}
//xuất dữ liệu 1 cấu trúc (sinh viên)
void Output_Struct(SINHVIEN p)
{
    cout<<"\n";
    cout << setiosflags(ios::left)
        << setw(12) << p.Maso
        << setw(22) << p.HoTen
        << setw(10) << p.Lop
        << setw(10) << p.Dtb
        << setw(10) << p.Tichluy;
}

//Xuất danh sách nhân viên ra màn hình
void Output_Data()
{
    for (int i = 0; i < Sct; i++)
        Output_Struct(Sv[i]);
}
//In tiêu đề
void Output()
{
    cout<<"\n";
    cout << setiosflags(ios::left)
        << setw(12) << "MASO"
        << setw(22) << "HOTEN"
        << setw(10) << "LOP"
        << setw(10) << "DTB"
        << setw(10) << "TICHLUY";
}

```

IV. Bài tập

Yêu cầu:

Tạo một Project gồm các tập tin:

- Tập tin thư viện **h*: Chứa các hàm chức năng của chương trình
- Tập tin chương trình **cpp*: Chứa hàm *main()*, các hàm tổ chức menu, nhập xuất dữ liệu.

Giả sử có một danh sách nhân viên, mỗi một nhân viên được lưu trữ các thông tin:

- Mã nhân viên
- Họ
- Chữ lót
- Tên
- Năm sinh
- Lương

Thực hiện thao tác xuất tất cả các nhân viên:

- Có Mã NV cho trước
- Có Tên cho trước
- Có Tiền lương cho trước
- Có Năm sinh cho trước

Giả sử danh sách nhân viên lưu trữ trong tập tin “NHANVIEN.TXT” như sau :

0912345	Nguyen	Van	Minh	1970	1234567.5
1234567	Tran		Hoang	1965	3214568.0
1324326	Hoang	Hoa	Ngoc	1985	1543678.5
1121245	Truong	Van	Hau	1976	2012325.0
1121435	Nguyen	Minh	Hoang	1965	1543678.5