顺序查找的实现

#include<bits/stdc++.h>

using namespace std;

typedef struct {

int key;//查找表中数据的值

//按需求添加其他属性

}ElemType;

typedef struct {

ElemType \*elem;//存放查找表中数据元素的数组

int length;//记录查找表中的数据总数量

}SSTable;

//创建查找表

void Creat(SSTable \*\*st,int length) {

(\*st) =(SSTable\*) new SSTable;

(\*st)->length = length;

(\*st)->elem = (ElemType \*) new ElemType[length + 1];

//根据查找表中数据元素的总长度，在存储时，从数组下标为 1 的空间开始存储数据

cout << "输入表中元素:\n";

for (int i = 1; i <= length; ++i) {

cin >> ((\*st)->elem[i].key);

}

}

//查找表查找的功能函数，其中key为关键字

int Search\_seq(SSTable \*st, int key) {

st->elem[0].key = key;//将关键字作为一个数据元素存放在查找表的第一个位置，监视哨作用

int i = st->length;

//从查找表的最后一个元素依次遍历，直到下标为0

while (st->elem[i].key != key)

i--;

//如果 i=0，说明查找失败；反之，返回的是含有关键字key的数据元素在查找表中的位置

return i;

}

int main() {

SSTable \*st;

Creat(&st, 6);

getchar();

printf("请输入查找数据的关键字:\n");

int key;

cin >> key;

int location = Search\_seq(st, key);

if (location == 0) {

printf("查找失败\n");

}

else {

printf("数据在查找表中的位置为：%d", location);

}

return 0;

}

# 二分查找（折半查找）算法

#include<bits/stdc++.h>

using namespace std;

#define MAX\_INT 20

//int a[MAX\_INT];//存储元素

int n;//数组大小

int BinarySearch(int a[], int value) {

int left = 0;

int right = n - 1;

while (left <= right) {

//最好改用位运算符而不是 /2

//https://www.cnblogs.com/Kanna/p/12371164.html (位运算详解)

if (value > a[(left + right) / 2])

left = (right + left) / 2;

else if (value < a[(left + right) / 2])

right = (right + left) / 2;

else

{

return (left + right) / 2;

}

}

return -1;

}

int main() {

int a[] = { 1,2,3,4,5 };//sorted arrays

n = 5;

int ans = BinarySearch(a, 3);

if (ans == -1)cout << " Not find " << endl;

else cout << ans + 1 << endl;

return 0;

}