**实 验 报 告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：数据结构 | **班级**：软工21101 | **实验成绩**： |
| **实验名称**：查找的实现 | **学号**：2105006207 | **批阅教师签字：董傲霜** |
| **实验编号**：实验十一 | **姓名**：方福涛 | **实验日期：**2022 年 11 月24 日 |
| **指导教师**：董傲霜 | **组号**：wuyanzu | **实验时间**： 时 分－ 时 分 |

1. **实验设计思想**

说明查找表实现的设计思想。

**顺序查找基本思想：**

从查找表的一端开始，逐个将记录的关键字值和给定值进行比较，如果某个记录的关键字值和给定值相等，则称查找成功；否则，说明查找表中不存在关键字值为给定值的记录，则称查找失败。

**二分查找基本思想：**

先取查找表的中间位置的关键字值与给定关键字值作比较，若它们的值相等，则查找成功；如果给定值比该记录的关键字值大，说明要查找的记录一定在查找表的后半部分，则在查找表的后半部分继续使用折半查找；若给定值比该记录的关键字值小，说明要查找的记录一定在查找表的前半部分，则在查找表的前半部分继续使用折半查找。…直到查找成功，或者直到确定查找表中没有待查找的记录为止，即查找失败。

**二、程序说明**

1. 给出查找表存储结构定义及其属性含义的说明；

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

1. 给出你设计的顺序和折半查找函数的说明（函数，参数，执行的结果）

**顺序查找：**

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;

}

**折半查找：**

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;

}

**三、实验环境**

Windows10，DEV-C++

**四、实验过程分析**

实验调试过程中的问题及改正的事件举例说明（只要调程序就不可能没有错误，现象、原因、修正方法，可以截图但不能过多）。

Ps（注意点）：

1.在做查找的过程中，如果 low 指针和 high 指针的中间位置在计算时位于两个关键字中间，即求得 mid 的位置不是整数，需要统一做取整操作。

2.当查找表使用链式存储结构表示时，折半查找算法无法有效地进行比较操作（排序和查找操作的实现都异常繁琐）。

3.折半查找要基于数据是有序的才可以实现。所在在查找之前要进行数据的排序。

**五、实验结果总结**

1、阐述顺序查找和折半查找的性能分析。

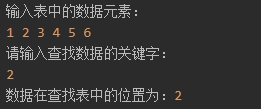
**顺序查找：**

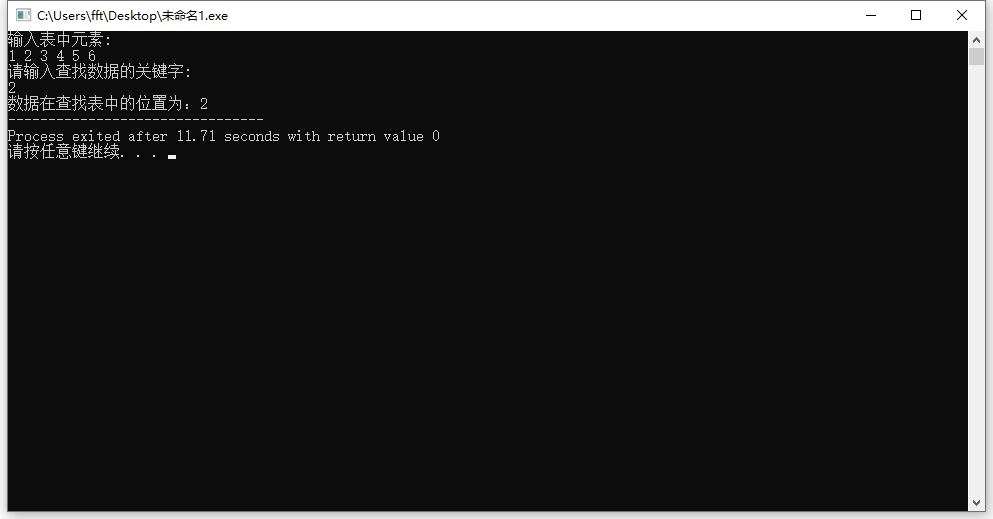
时间复杂度为O(n),平均复杂度为O(n/2)

**折半查找：**

时间复杂度为O(logn)

2、给出你的测试方法和测试截图（不能超过三个截图）；





**六、附录**

1. 意见和建议（没有可不写）。

2. 思考题：

回答以下问题：

1. 举例说明折半查找的应用。

主要分为两种：一种是遍历非递归形式，

一种是采用递归的形式。

例如：可以用于排序数组中元素的查，

也可以用于有序数组中插入元素等等。

1. 说明静态查找和查找表实现的区别。

**静态查找**(只是查询)：

·查询某个 ”特定的“ 数据元素是否在查找表中

·检索某个 ”特定的“ 数据元素和各种属性

**查找表的实现（**相比于静态查找表，查找过程中会修改元素**）：**

·查找时**插入**数据元素

·查找时**删除**数据元素

**七、打分表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 考核点 | 分数 | **得分** | 备注 |
| 程序 | 逻辑是否正确  程序可读性  创新点 | 50 | **0** |  |
| 报告完整性 | 实验过程阐述是否完整  测试数据设计是否合理  运行结果是否正确 | 40 | **0** |  |
| 调试问题及解决方法 | 是否对调试过程问题进行阐述 | 5 | **0** |  |
| 思考题目 | 回答是否正确 | 5 | **0** |  |
| 合计 |  | 100 | **0** |  |