**西南石油大学实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **项目编号** | 001063000505 | **项目名称** | 查找及排序算法的综合应用 | **成绩** |  |
| **专业年级** | **软件工程2022级** | | | **指导教师** | **胡卫东** |
| **姓名** | **李浩楠** | **学号** | **202231060920** | **实验日期** | **2023/6/5** |

1. **实验目的**
2. 掌握顺序表查找中不同查找方法的查找思想，并能用C/C++语言实现。
3. 掌握Hash表查找中的查找思想，并能用C/C++语言实现。
4. 能够针对具体实际，灵活选用适宜的查找方法。
5. 掌握各种排序方法的特点及适用情形，并能在解决实际问题的过程中灵活选用不同的排序方法。
6. 掌握各种排序方法的操作过程及其依据。
7. **实验工具**

PC微机、Windows、PTA或Dev C++。

1. **实验步骤**
2. 创建哈希表

输入不大于m的n个不为0（0表示空值）的数，用线性探查法解决冲突构造散列表。（程序填空题）

1. 堆排序

本题要求利用堆排序将数组R[]中的元素按照非递减顺序排序。（程序填空题）

函数void Sift(int R[],int low,int high)用于调整堆结构，调整范围为节点序号为low到high.

HeapSort函数用于堆排序，将R[1]到R[n]元素进行非递减排序。

1. **数据结构考题 - 快速排序 - 主调函数**

用顺序表作存储结构，输入一组数据，用快速排序法对其进行排序。

1. **编写函数实现直接插入排序**

编写函数实现直接插入排序。输入若干正整数，以0结束，采用直接插入排序将这些整数按照从大到小的顺序排序。

1. **实验结果**
2. 测试用例：12

19 14 23 1 68 20 84 27 55 11 10 79

代码：HT[H0].key==NULLKEY

Hi=(H0+i)/m

HT[Hi].key==NULLKEY

运行结果：0 14 1 68 27 55 19 20 84 79 23 11 10 0 0 0

1. 测试用例：无

代码：j\*2

temp

n/2

Sift(R,1,n-1)

运行结果：无

1. 测试用例：8

19 -18 13 26 275 26 7 35

代码：

void QuickSort(SqList &L, int low, int high)

{

int i;

if (low<high )

{

i=Partition(L, low, high );

QuickSort(L, low, i-1 );

QuickSort(L, i+1 , high );

}

}

运行结果：-18 7 13 19 26 26 35 275

1. 测试用例：5 9 1 3 0

代码：void InsertSort(SqList &L)

{

int i,j;

for(i=2;i<=L.length;i++)

{

if(L.r[i]>L.r[i-1])

{

L.r[0]=L.r[i];

L.r[i]=L.r[i-1];

for(j=i-2;L.r[0]>L.r[j];j--)

{

L.r[j+1]=L.r[j];

}

L.r[j+1]=L.r[0];

}

}

}

运行结果：9 5 3 1

1. **实验总结**
2. 线性表的查找主要包括顺序查找、折半查找和分块查找。其中顺序查找的时间复杂度为O(n),折半查找的时间复杂度为O(nlogn)。顺序查找算法简单，对表的结构无任何要求，但查找效率较低，适用于不经常插入和删除的线性表；折半查找对表的结构要求较高，查找效率较高，适用于有序并且不经常插入和删除的线性表。
3. 散列表的查找依靠所建立的散列函数，但存在一个地址对应多个值的问题，解决这种冲突问题的办法主要分为两大类，即开放地址法和链地址法。其中开放地址法包括线性探测法、二次探测法和伪随机探测法。
4. 排序分为内部排序和外部排序。其中内部排序主要有直接插入排序、折半插入排序、希尔排序、冒泡排序、简单选择排序、快速排序和堆排序。时间复杂度分别为O(n^2),O(n^2),O(n^1.3),O(n^2),O(n^2),O(nlogn)和O(nlogn).希尔排序快速排序和堆排序为不稳定排序.当关键字分布随机,对稳定性不做要求时,选择快速排序;当关键字基本有序,对稳定性不做要求时,选择堆排序;当关键字基本有序时,对稳定性不做要求时,选择归并排序.