数据结构实验报告

班级：计科1194 姓名：万一凡 学号：201911111420

实验日期：2020.12.2,2020,12.9

实验名称：查找和排序

实验内容：

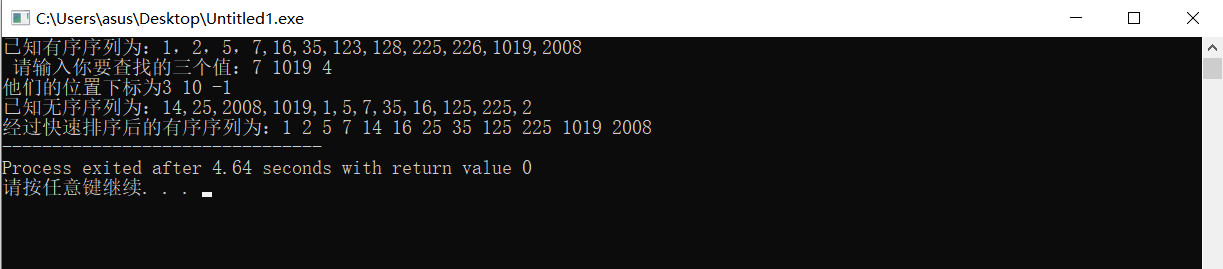
1.折半查找的实现

2.快速排序的实现

算法描述或算法思路：

1. 折半查找：查找过程从 数组的中间元素开始，如果中间元素正好是要查找的元素，则搜索过程结束；如果某一元素大于或小于中间元素，则在数组大于或小于中间元素的那一半查找，而从开始一样从中间元素开始比较，如果在某有不数组为空，则代表查不到。平均时间复杂度为：
2. 快速排序：采用分治的思想，首先选取一个基准，将比基准小的数移到一边，比基准小的数移到另一边，此时，基准的位置是正确排好序的。然后再左右子区间里不断的递归处理，选取新的基准，直至每部分只有一元素或为空为止，平均时间复杂度为：

运行情况及分析：



运行正确无误，第一个实验用例有序序列为1,2,5,7,16,35,123,128,225,226,1019,2008。第二个实验排序用例为14,25,2008,1019,1,5,7,35,16,125,225,2。

源代码：

#include <stdio.h>

int search(int a[],int n,int k){//查找的已排序数组，数组长度，寻找的关键字

int head,tail,mid;

head=0;tail=n-1;//head为头，tail为尾

while(head<=tail){

mid=(head+tail)/2;

if(a[mid]==k){//找到了返回下标

return mid;

}else if(a[mid]>k){//k比中间值大则在右半边找

tail=mid-1;

}else if(a[mid]<k){//k比中间值小则在左半边找

head=mid+1;

}

}

return -1;//找不到就返回-1

}

int part(int a[],int i,int j){//一次划分的结果

int s=i,t=j;

int aa=a[s];

while(s<t){

while(t>s&&a[t]>=aa){//从右往左找到第一个比基准小的值

t--;

}

a[s]=a[t];

while(s<t&&a[s]<=aa){//从左往右找到第一个比基准大的值

s++;

}

a[t]=a[s];//此步和上面一步是将两个值交换

}

a[s]=aa;//将基准放到正确位置

return s;//返回基准的下标

}

void quick(int a[],int i,int j){

int s;

if(i<j){

s=part(a,i,j);

quick(a,i,s-1);//在左区间递归

quick(a,s+1,j);//在右区间递归

}

}

int main (){

int a[12]={1,2,5,7,16,35,123,128,225,226,1019,2008};

int n=12;

int p,q,m;

printf("已知有序序列为：1，2，5，7,16,35,123,128,225,226,1019,2008\n 请输入你要查找的三个值：");

scanf("%d %d %d",&p,&q,&m);

printf("他们的位置下标为%d %d %d \n",search(a,n,p),search(a,n,q),search(a,n,m));

int b[12]={14,25,2008,1019,1,5,7,35,16,125,225,2};

printf("已知无序序列为：14,25,2008,1019,1,5,7,35,16,125,225,2\n");

quick(b,0,11);

int i=0;

printf("经过快速排序后的有序序列为：");

for(i=0;i<12;i++){

printf("%d ",b[i]);

}

return 0;

}

实验体会：

算法思想比较简单，主要是线性表和递归的运用，加强练习对提高编程能力有很大的帮助。