**中国矿业大学计算机学院实验报告**

|  |
| --- |
| 课程名称  算法导论           实验名称 算法导论  班级 计科17-6 姓名 郭子杨 学号 08173042  仪器组号\_\_\_\_\_\_\_  实验日期 10.22  实验报告要求：1.实验目的   2.实验内容（题目描述，源代码，运行截图，调试情况）    3.实验体会 |
| 一、实验目的   1. 熟悉C/C++语言的集成开发环境； 2. 通过本实验加深对递归过程的理解   二、实验内容  1、第一题  1.1题目描述   |  |  | | --- | --- | | **1.** | 【问题描述】  任意输入一个整数，输出结果能够用递归方法实现整数的划分。  【输入形式】一个整数 【输出形式】所有该数字的划分 【样例输入】          6 【样例输出】          6          5+1          4+2          4+1+1          3+3          3+2+1          3+1+1+1          2+2+2          2+2+1+1          2+1+1+1+1          1+1+1+1+1+1 |   1.2 源代码  #include<iostream>  #include<stdio.h>  using namespace std;  int mark[10];  int a;  void divide(int n,int m,int p)  {    int i;  if(n > a) return;  if(n == a)  {  for(i = 0; i < m-1; i++)  printf("%d+",mark[i]);  printf("%d\n",mark[i]);  }  else  {  for(i = p; i > 0; i--)  {  if(i <= p)  {  mark[m]=i;  n+=i;  divide(n,m+1,i);  n-=i;  }  }  }  }  int main()  {  cin>>a;  if(a>0)  {  cout<<a<<endl;  divide(0,0,a-1);  }  return 0;  }  1.3 运行截图    1.4 调试情况  无  2、第二题  2.1 题目描述  【问题描述】  1、设a[0:n-1]是一个已排好序的数组。请改写二分搜索算法，使得当搜索元素x不在数组中时，返回小于x的最大元素的位置i和大于x的最大元素位置j。当搜索元素在数组中时，i和j相同，均为x在数组中的位置。  【输入形式】  输入有两行：  第一行是n值和x值；  第二行是n个不相同的整数组成的非降序序列，每个整数之间以空格分隔。  【输出形式】  第一行输出小于x的最大元素的最大下标i和大于x的最小元素的最小下标j。  当搜索元素在数组中时，i和j相同。  提示：若x小于全部数值，则输出：-1 0 ;若x大于全部数值，则输出：n-1 n  第二行若找到返回下标,否则返回-1  【样例输入】  6 5 2 4 6 8 10 12  【样例输出】  1 2  -1  2.2 源代码  #include <iostream>  using namespace std;  int index=0;  void binarySearch(int list[],int left,int right,int number)  {  if(list==NULL)  {  cout<<"NULL"<<endl;  index=-1;  }  int mid=(right+left)/2;  if(left>right&&index==0)  {  index=-1;  cout<<right<<" "<<left<<endl;  }  if(number==list[mid]&&index==0)  {  index=-2;  cout<<mid<<" "<<mid<<endl;  cout<<mid<<endl;    }  else if(number>list[mid]&&index==0)  {  binarySearch(list,mid+1,right,number);  }  else if(number<list[mid]&&index==0)  {  binarySearch(list,left,mid-1,number);  }  }  int binarycodeSearch(int list[],int left,int right)  {  int middle=(left+right)/2;  while(left<=right)  {  middle=(left+right)/2;  if(middle==list[middle])  {  return middle;  }  if(middle>list[middle])  {  left=middle+1;  }  if(middle<list[middle])  {  right=middle-1;  }  if(left>right)  {  return -1;  }  }  }  int main()  {  int n;  cin>>n;  int x;  cin>>x;  int a[200];  for(int i=0;i<n;i++)  {  cin>>a[i];  }  int left = 0;  int right = n-1;  binarySearch(a,left,right,x);  if(index!=-2)  {  int m=binarycodeSearch(a,0,n-1);  cout<<m<<endl;  }  return 0;  }  2.3 运行截图    2.4 调试结果  无  3、第三题  3.1 问题描述  【问题描述】  给定线形序列集中n个元素和一个整数k，1≤k≤n，输出这n个元素中第k小元素的值及其位置。  注：使用分治法编程。  【输入形式】   第一行输入n的值，第二行输入n个数，中间用空格隔开，第三行输入k的值。  【输出形式】   n个数中的第k小元素的值及其位置，中间用空格隔开。  【样例输入】  5  8 1 3 6 9  4  【样例输出】  8 1  3.2 源代码  #include<iostream>  #include<cstdio>  #include<cstring>  #include<stack>  #include<algorithm>  using namespace std;    void bubbleSort(int a[],int b[],int p,int r){  for(int i=p;i<r;i++){  for(int j=i+1;j<=r;j++){  if(a[j]<a[i])  {  swap(a[i],a[j]);  swap(b[i],b[j]);  }  }  }  }    int Partition(int a[],int b[],int p,int r,int val){  int pos;  for(int q=p;q<=r;q++){  if(a[q]==val)  {  pos=q;  break;  }  }  swap(a[p],a[pos]);  swap(b[p],b[pos]);    int i=p,j=r+1,x=a[p],y=b[p];  while(1){  while(a[++i]<x&&i<r);  while(a[--j]>x);  if(i>=j)break;  {swap(a[i],a[j]);  swap(b[i],b[j]);}  }  a[p]=a[j];  b[p]=b[j];  a[j]=x;  b[j]=y;  return j;  }    int Select(int a[],int b[],int p,int r,int k)  {  if(r-p<75)  {  bubbleSort(a,b,p,r);  return p+k-1;    }    for(int i=0;i<=(r-p-4)/5;i++){  int s=p+5\*i,t=s+4;  for(int j=0;j<3;j++){  for(int n=s;n<t-j;n++){  if(a[n]>a[n+1])  {  swap(a[n],a[n-1]);  swap(b[n],b[n-1]);  }  }  }  swap(a[p+i],a[s+2]);  swap(b[p+i],b[s+2]);  }    int x=Select(a,b,p,p+(r-p-4)/5,(r-p-4)/10);  int i=Partition(a,b,p,r,x),j=i-p+1;  if(k<=j)return Select(a,b,p,i,k);  else return Select(a,b,i+1,r,k-j);  }  int main(){  int a[1000],b[1000];  int n;  cin>>n;  for(int i=0;i<n;i++)  {  cin>>a[i];  b[i]=i+1;  }  int k;  cin>>k;  int m=Select(a,b,0,n-1,k);  cout<<a[m]<<" "<<b[m]<<endl;    return 0;  }  3.3 运行截图    3.4 调试结果  无  三、实验体会  通过本次上机学习，了解分治法的基本思想，掌握递归程序编写方法对课本上理论知识更加巩固，调试代码时也不免出现错误。 |