**实验报告**

**课程名称：算法设计与分析**

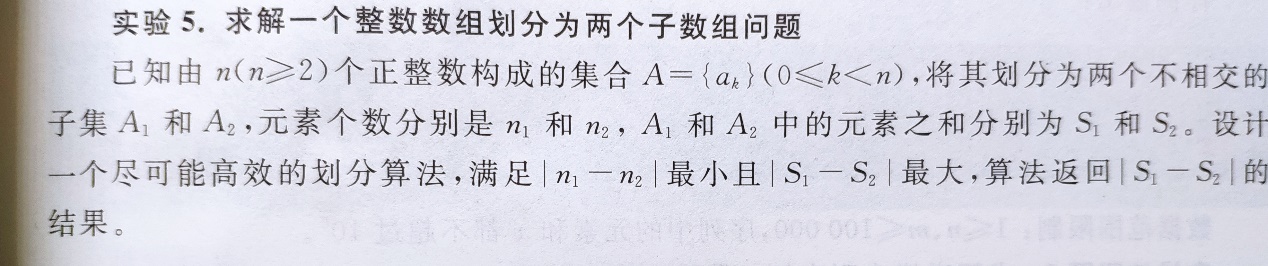
**专业班级：大数据182**

**学号姓名：3180439031陈佳婧**

**实验日期：2020年10月12日**

**第三章上机实验题5： 求解一个整数数组划分为两个子数组问题**

1. **实验目标和要求：**



**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

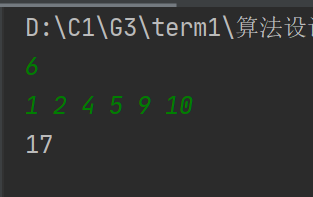
**快速排序至中位数时停止，S1为前半段之和，后半段为S2之和。**

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
using namespace std;  
  
int QuickSelect(int a[],int s,int t,int k,int n){  
 int sum1=0,sum2=0;  
 int i=s,j=t;  
 int tmp;  
 if(s<t){  
 tmp=a[s];  
 while(i!=j){  
 while(j>i&&a[j]>=tmp){  
 j--;  
 }  
 a[i]=a[j];  
 while(i<j&&a[i]<=tmp){  
 i++;  
 }  
 a[j]=a[i];  
 }  
 a[i]=tmp;  
 if(k-1==i){  
 int x=0;  
 for(x=0;x<k-1;x++){  
 sum1+=a[x];  
 }  
 for(x;x<n;x++){  
 sum2+=a[x];  
 }  
 return (sum1-sum2)>=0?(sum1-sum2):-(sum1-sum2);  
 }  
 else if(k-1<i)return QuickSelect(a,s,i-1,k,n);  
 else return QuickSelect(a,i+1,t,k,n);  
 }else if(s==t&&s==k-1){  
 int x=0;  
 for(x=0;x<k-1;x++){  
 sum1+=a[x];  
 }  
 for(x;x<n;x++){  
 sum2+=a[x];  
 }  
 return (sum1-sum2)>=0?(sum1-sum2):-(sum1-sum2);  
 }  
}  
int main() {  
 int a[200];  
 int n;  
 cin>>n;  
 for(int i=0;i<n;i++)cin>>a[i];  
 cout<<QuickSelect(a,0,n-1,n/2+1,n)<<endl;  
 return 0;  
}

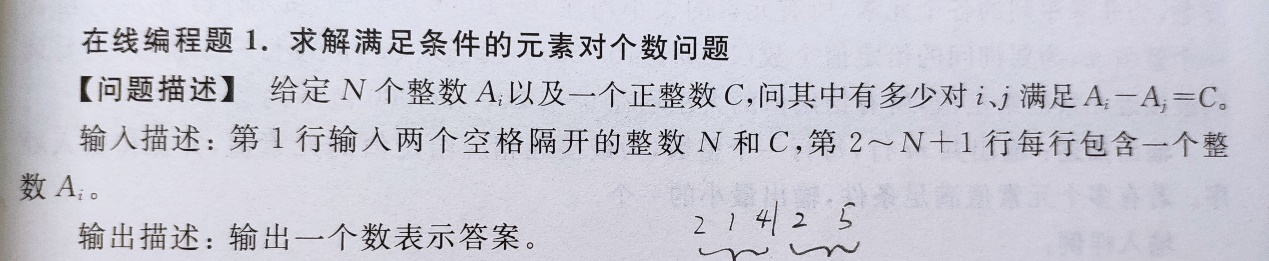
**6. 运行结果展示：**



**7. 实验体会（可选）：**

**第三章在线编程题1： 求解满足条件的元素对个数问题**

**1.实验目标和要求：**



**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

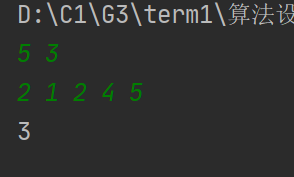
**归并排序后，折半查找出输入数+C的值，有则计入输出数。**

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
using namespace std;  
void Merge(int a[],int low,int mid,int high){  
 int \*tmpa;  
 int i=low,j=mid+1,k=0;  
 tmpa=(int \*)malloc((high-low+1)\*sizeof(int));  
 while(i<=mid&&j<=high){  
 if(a[i]<=a[j]){  
 tmpa[k]=a[i];  
 i++;k++;  
 }else{  
 tmpa[k]=a[j];  
 j++;k++;  
 }  
 }  
 while(i<=mid){  
 tmpa[k]=a[i];  
 i++;k++;  
 }  
 while(j<=high){  
 tmpa[k]=a[j];  
 j++;k++;  
 }  
 for(k=0,i=low;i<=high;k++,i++){  
 a[i]=tmpa[k];  
 }  
 free(tmpa);  
}  
void MergeSort(int a[],int low,int high){  
 int mid;  
 if(low<high){  
 mid=(low+high)/2;  
 MergeSort(a,low,mid);  
 MergeSort(a,mid+1,high);  
 Merge(a,low,mid,high);  
 }  
}  
  
int BinSearch(int a[],int low,int high,int k){  
 int mid;  
 if(low<=high){  
 mid=(low+high)/2;  
 if(a[mid]==k){  
 return mid;  
 }  
 if(a[mid]>k){  
 return BinSearch(a,low,mid-1,k);  
 }else{  
 if(mid==low&&abs(high-low)==1)return -1;  
 if(a[high]==k)return high;  
 else return BinSearch(a,mid,high,k);  
 }  
 }else return -1;  
}  
int main() {  
 int n,c,a[100005],cnt=0;  
 cin>>n>>c;  
 for(int i=0;i<n;i++){  
 cin>>a[i];  
 }  
 MergeSort(a,0,n-1);  
 for(int i=0;i<n;i++){  
 if(BinSearch(a,0,n-1,a[i]+c)!=-1)cnt++;  
 }  
 cout<<cnt<<endl;  
 return 0;  
}

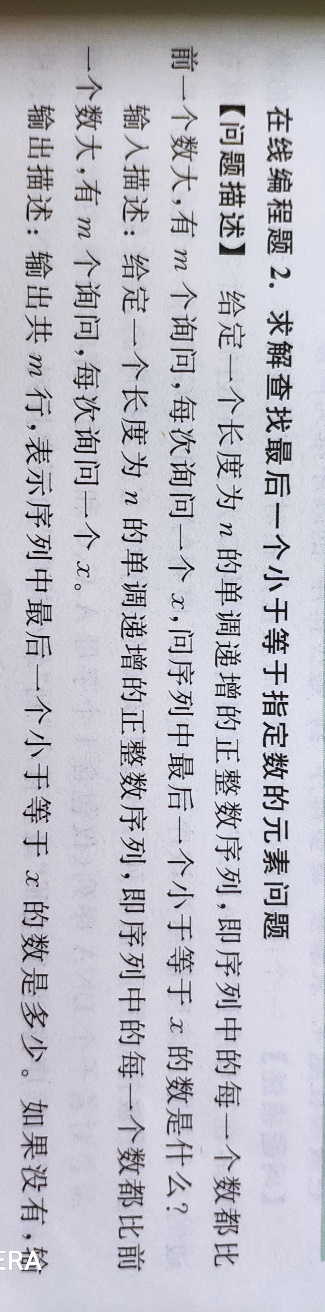
**6. 运行结果展示：**



**7. 实验体会（可选）：**

**第三章在线编程题2： 求解查找最后一个小于等于指定数的元素问题**

**1.实验目标和要求：**



**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

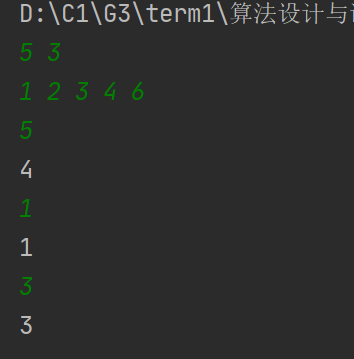
**快速排序在a[mid]>k时加判定if(mid==low)return mid，返回小于等于数。**

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
using namespace std;  
  
int BinSearch(int a[],int low,int high,int k){  
 int mid;  
 if(low<=high){  
 mid=(low+high)/2;  
 if(a[mid]==k){  
 return mid;  
 }  
 if(a[mid]>k){  
 return BinSearch(a,low,mid-1,k);  
 }else{  
 if(mid==low)return mid;  
 else return BinSearch(a,mid,high,k);  
 }  
 }else return -1;  
}  
int main() {  
 int n,m,a[100005];  
 cin>>n>>m;  
 for(int i=0;i<n;i++)cin>>a[i];  
 while(m--){  
 int x;  
 cin>>x;  
 cout<<a[BinSearch(a,0,n-1,x)]<<endl;  
 }  
 return 0;  
}

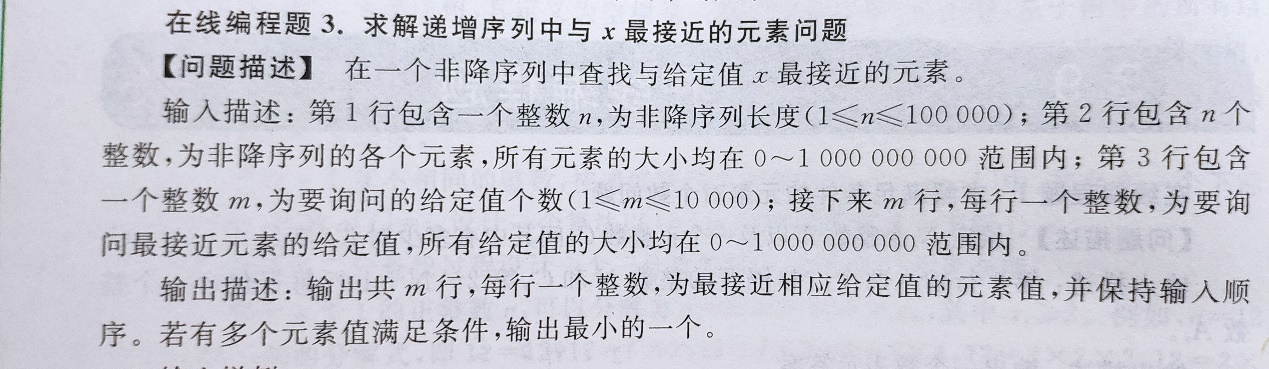
**6. 运行结果展示：**



**7. 实验体会（可选）：**

**在线编程题3： 求解递增序列中与x最接近的元素问题**

**1.实验目标和要求：**



**2. 实验环境：（操作系统、语言、编译工具…）WIN10、C++、Clion**

**3. 关键问题及解决思路：**

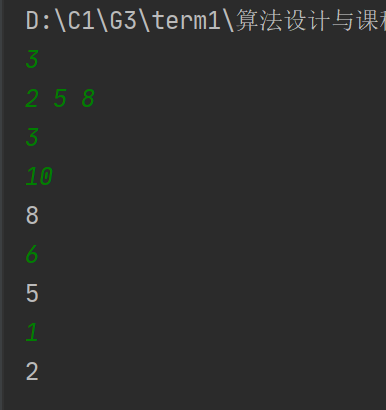
**快速排序找到只剩两个数时，判断哪个更接近给定值。**

**4. 程序流程：**

**5. 完整程序代码：**

#include <iostream>  
using namespace std;  
  
int BinSearch(int a[],int low,int high,int k){  
 int mid,left,right;  
 if(low<=high){  
 left=low;  
 right=high;  
 mid=(low+high)/2;  
 if(a[mid]==k){  
 return mid;  
 }  
 if(a[mid]>k){  
 if(mid==high)right=mid;  
 else return BinSearch(a,low,mid-1,k);  
 }else{  
 if(mid==low)left=mid;  
 else return BinSearch(a,mid,high,k);  
 }  
 if(abs(k-a[left])<=abs(a[right]-k))return left;  
 else return right;  
 }else return -1;  
}  
int main() {  
 int n,m,a[100005];  
 cin>>n;  
 for(int i=0;i<n;i++)cin>>a[i];  
 cin>>m;  
 while(m--){  
 int x;  
 cin>>x;  
 cout<<a[BinSearch(a,0,n-1,x)]<<endl;  
 }  
 return 0;  
}

**6. 运行结果展示：**



**7. 实验体会（可选）：**