|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学号 | 2022212080 | 算法思路(30%) | 编码实现与算法分析(50%) | 实验报告(20%) | 得分 |
| 姓名 | 刘纪彤 |  |  |  |  |
| 评语 |  | | | | |

### 《算法设计与分析》实验报告

实验2 分治算法实验

**一、实验目的**

1. 加深对分治法算法设计的理解，包括其基本原理和递归调用的机制。

2. 学习如何将大问题分解为小问题，并独立解决这些小问题。

3. 掌握分治法算法的三个步骤：分解、解决、合并。

4. 分析分治法算法的时间复杂度和空间复杂度。

5. 提高编程能力和复杂问题解决能力。

**二、实验内容(题目)**

用分治法实现一组无序序列的两路合并排序和快速排序。要求清楚合并排序及快速排序的基本原理，编程实现分别用这两种方法将输入（或者利用随机函数生成）的一组（不得少于50个数据）无序序列排序为有序序列后输出。

。

**三、算法设计思路**

二路合并排序算法思路，利用递归，对左右两个分别进行递归调用进行排序，同时再把已经大体上算好的进行合并。

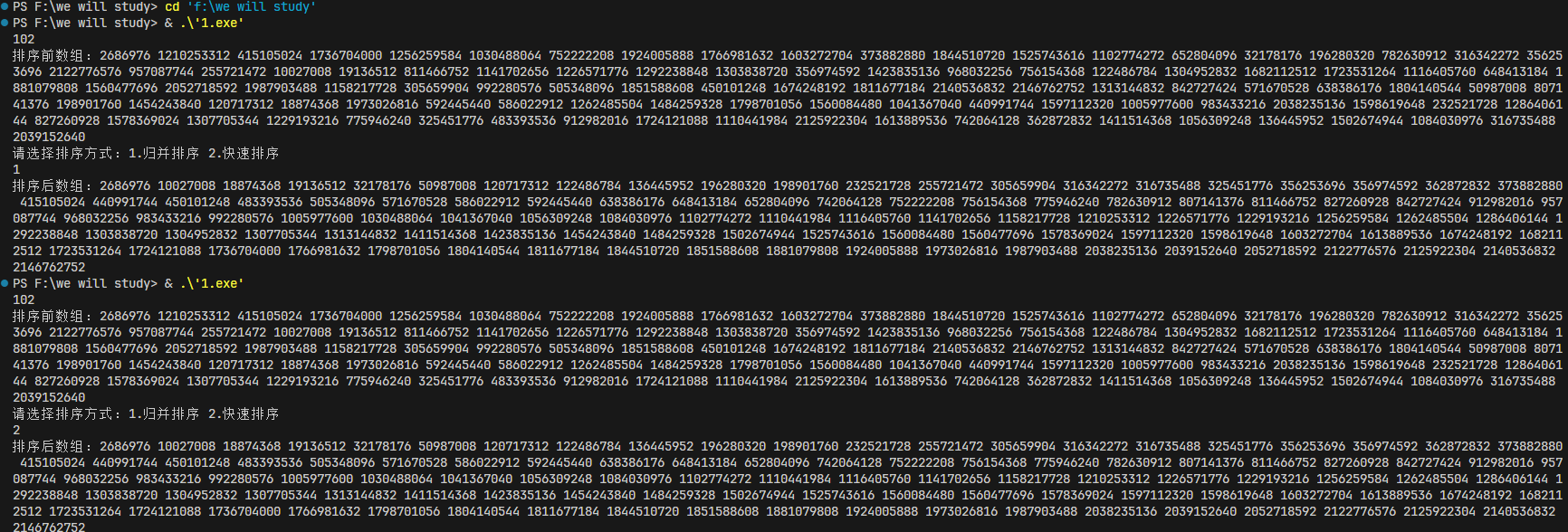
快速排序就是，利用分治的策略，将比Key小的放在左侧，key大的放在右侧，按照递归分治的思想最终输出一个排好序的值

**四、各功能模块设计**

当他

#include <bits/stdc++.h>  
using namespace std;  
  
  
void merge(vector<int> &b,int l,int mid,int r)  
{  
 vector<int> temp(r-l+1);  
 int i=l;  
 int j=mid+1;  
 int k=0;  
 while((i<=mid)&&(j<=r))  
 {  
 if(b[i]<=b[j]) temp[k++]=b[i++];  
 else temp[k++]=b[j++];  
 }  
 while(i<=mid) temp[k++]=b[i++];  
 while(j<=r) temp[k++]=b[j++];  
 for(i=l,k=0;i<=r;i++,k++) b[i]=temp[k];//将临时存储还给  
}  
void mergeSort(vector<int> &b,int l,int r)  
{  
 if(l<r){  
 int mid=l+(r-l)/2;  
 mergeSort(b,l,mid);  
 mergeSort(b,mid+1,r);  
 merge(b,l,mid,r);  
 }  
}  
void quickSort(vector<int> &b,int l,int r)  
{  
 if(l>=r) return;  
 int i=l;  
 int j=r;  
 int key=b[l];//关键值  
 while(i<j)  
 {  
 while(i<j&&b[j]>=key) j--;  
 b[i]=b[j];  
 while(i<j&&b[i]<=key) i++;  
 b[j]=b[i];  
 }  
 b[i]=key;  
 quickSort(b,l,i-1);  
 quickSort(b,i+1,r);  
}  
int main(){  
 vector<int> a;  
 int n;  
 cin>>n;  
 for(int i=0;i<n;i++)  
 {  
 a.push\_back(((float)(1.0\*rand()/(RAND\_MAX + 1)))\*(50));  
 }  
 cout<<"排序前数组：";  
 for(int b:a)  
 {  
 cout<<b<<" ";  
 }  
 cout<<endl;  
 int ch;  
 cout<<"请选择排序方式：1.归并排序 2.快速排序"<<endl;  
 cin>>ch;  
 if(ch==1) mergeSort(a,0,n-1);  
 if(ch==2) quickSort(a,0,n-1);  
 cout<<"排序后数组：";  
 for(int b:a)  
 {  
 cout<<b<<" ";  
 }  
 cout<<endl;  
}

**五、运行结果与分析**

2-1

如图所示2-1对于排好序的输出结果均为单调递增，故输出都是正确的值。

**六、实验总结**

通过本次实验我已经了解并掌握了分治的基本逻辑，结合前述所学的递归知识，能够将分治的思想利用已学的递归结合在一起，能够使用分治的思路解决实际应用中的问题，受益匪浅。