# 实验一：递归与分治

**一、实验目的**

理解递归算法的思想和递归程序的执行过程，并能熟练编写递归程序。

掌握分治算法的思想，对给定的问题能设计出分治算法予以解决。

**二、实验内容**

段合并排序算法：

将数组划分为个子数组，每个子数组有个元素。然后递归地对分割后的子数组进行排序，最后将所得到的个排好序的子数组合并排序。

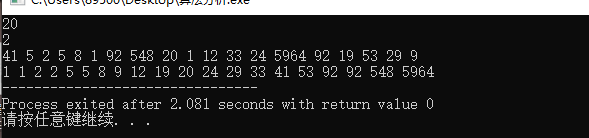
1. **分治算法基本思想**

先将数组分成根号n个大小为根号n的数组，然后对这些数组进行排序并进行根号n个数组合并。

1. **实验过程**

实现算法发现如果区间取的太小会发生死循环，所以把最小区间取为3，再小不在分，排序完合并也要进行根号n次合并

1. **实验结果**



代码：

#include<stdio.h>

#include<math.h>

#include<stdlib.h>

void merge(int num[],int l1,int r1,int l2,int r2){

int size1 = r1 - l1;

int size2 = r2 - l2;

int \*arr1 = (int \*)malloc(sizeof(int)\*size1);

int \*arr2 = (int \*)malloc(sizeof(int)\*size2);

for(int i=0;i<size1;i++){

arr1[i] = num[l1+i];

}

for(int i=0;i<size2;i++){

arr2[i] = num[l2+i];

}

int \*arr3;

if( l1 < l2){

arr3 = num+l1;

}else{

arr3 = num+l2;

}

int p=0,q=0,a=0;

while(p<size1 && q<size2){

if(arr1[p] < arr2[q]){

arr3[a] = arr1[p];

a++;

p++;

}else{

arr3[a] = arr2[q];

a++;

q++;

}

}

if(p<size1){

while(p<size1){

arr3[a] = arr1[p];

p++;

a++;

}

}else if(q<size2){

while(q<size2){

arr3[a] = arr2[q];

q++;

a++;

}

}

}

//选择排序

int sort(int\* arr, int size)

{

if (size == 0)

{

return 0;

}

for (int i = 0; i < size; i++)

{

for (int j = i + 1; j < size; j++)

{

if (arr[j] < arr[i])

{

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

return 0;

}

void devide(int num[],int l,int r){

int size = r-l;

int sq = sqrt(size);

if(size >3){

//分出根号n个区块

for(int i=0;i<sq;i++){

devide(num,l+i\*sq,l+(i+1)\*sq);

}

//为了把最后的数字也排序进去

devide(num,l+sq\*(sq-1),r);

}else{

sort(num+l,size);

}

//合并根号n个区块

int firstl = l;

int firstr = l+sq;

for(int i=1;i<sq;i++){

int lastl = l +sq\*i;

int lastr = lastl+sq;

merge(num,firstl,firstr,lastl,lastr);

firstr += sq;

}

merge(num,firstl,firstr,firstr,r);

}

int main(){

int n;

scanf("%d",&n);

int num[n];

for(int i=0;i<n;i++){

scanf("%d",&num[i]);

}

devide(num,0,n);

for(int i=0;i<n;i++){

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

}

}