



太原理工大学  
TAIYUAN UNIVERSITY OF TECHNOLOGY

## 本科试验汇报

课程名称：\_\_\_\_\_ 算法设计与分析 \_\_\_\_\_

试验项目：\_\_\_\_\_ 算法设计与分析试验 \_\_\_\_\_

试验地点：\_\_\_\_\_ 致远楼 403 \_\_\_\_\_

专业班级：\_\_\_\_\_ 学号： \_\_\_\_\_

学生姓名：\_\_\_\_\_

指导教师：\_\_\_\_\_

2017 年 3 月 28 日

# 试验一 分治法合并排序

## 一、试验目的

- 1.掌握合并排序的基本思想
- 2.掌握合并排序的实现措施
- 3.学会分析算法的时间复杂度
- 4.学会用分治法处理实际问题

## 二、试验内容

随机产生一种整型数组，然后用合并排序将该数组做升序排列，规定输出排序前和排序后的数组。

## 三、试验环境

程序设计语言：c++

编程工具：microsoft visual studio 2023

## 四、程序代码

```
#include "stdafx.h"  
  
#include<iostream>  
  
#include<cassert>
```

```
#include "SortTestHelper.h"

using namespace std;

template<typename T>

void mergeSort(T arr[], int n)

{ _mergeSort(arr, 0, n-1); }

template<typename T>

void __mergeSort(T arr[], int l, int r)

{

if(l>=r)

    return;

    int mid=(l+r)/2;

    __mergeSort(arr, l, mid);

    __mergeSort(arr, mid+1, r);

    if(arr[mid]>arr[mid+1])

        __merge(arr, l, mid, r);

}

template<typename T>

void __merge(T arr[], int l, int mid, int r)

{

    T *aux=new T[r-l+1];
```

```
for(int i=1;i<=r;i++)

    aux[i-1]=arr[i];

int i=1, j=mid+1;

for(int k=1;k<=r;k++)

{

if(i>mid)

    {

        arr[k]=aux[j-1];

        j++;

    }

else if(j>r)

    {

        arr[k]=aux[i-1];

        i++;

    }

else if(aux[i-1]<aux[j-1])

    {

        arr[k]=aux[i-1];

        i++;

    }

}
```

```

        else{

            arr[k]=aux[j-1];

            j++;

        }

    }

    delete[] aux;
}

int _tmain(int argc, _TCHAR* argv[])
{

    int n=10;

    int *arr=SortTestHelper::generateRandomArray(n, 0, n);

    cout<<"未排序的数组为: ";

    for(int j=0;j<10;j++)

        cout<<arr[j]<<" ";

    cout<<endl;

    mergeSort(arr, 10);

    cout<<"通过排序的数组为: ";

    for(int i=0;i<9;i++)

        cout<<arr[i]<<" ";

```

```
        cout<<endl;
    }

#include"stdafx.h"

#include<iostream>

#include<ctime>

#include<cassert>

using namespace std;

namespace SortTestHelper{

int *generateRandomArray(int n,int randL,int randR)

{

    assert(randL<=randR);

    int *arr=new int[n];

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

        arr[i]=rand()%(randR-randL+1)+randL;

    return arr;

}

}
```

## 五、试验成果截图

```
C:\WINDOWS\system32\cmd.exe
排序前的数组为: 8 9 9 1 7 5 5 10 1 0
排序后数组顺序为: 0 1 1 5 5 7 8 9 9
请按任意键继续. . .
```

## 六、试验总结

一定要先找到递归函数式后，设计递归程序

## 试验二 贪心法多机调度

### 一、 试验目的

1. 掌握贪心算法的基本思想
2. 掌握贪心算法的经典问题求解
3. 深入多机调度的基本思想和算法设计措施
4. 学会用贪心法分析和处理实际问题

### 二、 试验内容

设计贪心算法实现作业调度，规定按作业调度次序输出作业序列。如已知  $n=8$ ，效益  $p=(35, 30, 25, 20, 15, 10, 5, 1)$ ，时间期限  $d=(4, 2, 4, 5, 6, 4, 5, 7)$ ，求该条件下的最大效益。

### 三、 试验环境

程序设计语言：c++

编程工具：microsoft visual studio 2023

### 四、 措施描述和程序代码

```
#include "stdafx.h"

#include "iostream"

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{

    double work[99], time[99], t, w;

    double temp[99], usetemp[99], a, s=0;

    int A[99];

    cout<<"作业数为(x<99):"<<endl;

    cin>>w;

    cout<<"作业收益为:"<<endl;

    for(int i=0;i<w;i++){cin>>work[i];}

    cout<<"作业收益: "<<endl;

    for(int i=0;i<w;i++){cout<<" "<<work[i]<<" ";}

    cout<<endl;

    cout<<"每个作业的时限为:"<<endl;
```



```

for(int i=0;i<w;i++){cin>>time[i];}

cout<<"作业时限："<<endl;

for(int i=0;i<w;i++){cout<<" " <<time[i]<<" " ;}

cout<<endl;

cout<<"总作业时限为："<<endl;

cin>>t;//初始化录入数据

for(int i=0;i<w;i++){temp[i]=work[i]/time[i];usetemp[i]=temp[i];}//平均时间
盈利

cout<<"作业平均时间盈利排序为："<<endl;

for(int m=0;m<w;m++){

    a=temp[0],A[m]=0;

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

        if(a<temp[i]){

            A[m]=i;a=temp[i];}

        temp[A[m]]=0;cout<<" " <<A[m]<<" " ;

    }//作业平均时间盈利排序

cout<<endl;

cout<<"可进行的作业有："<<endl;

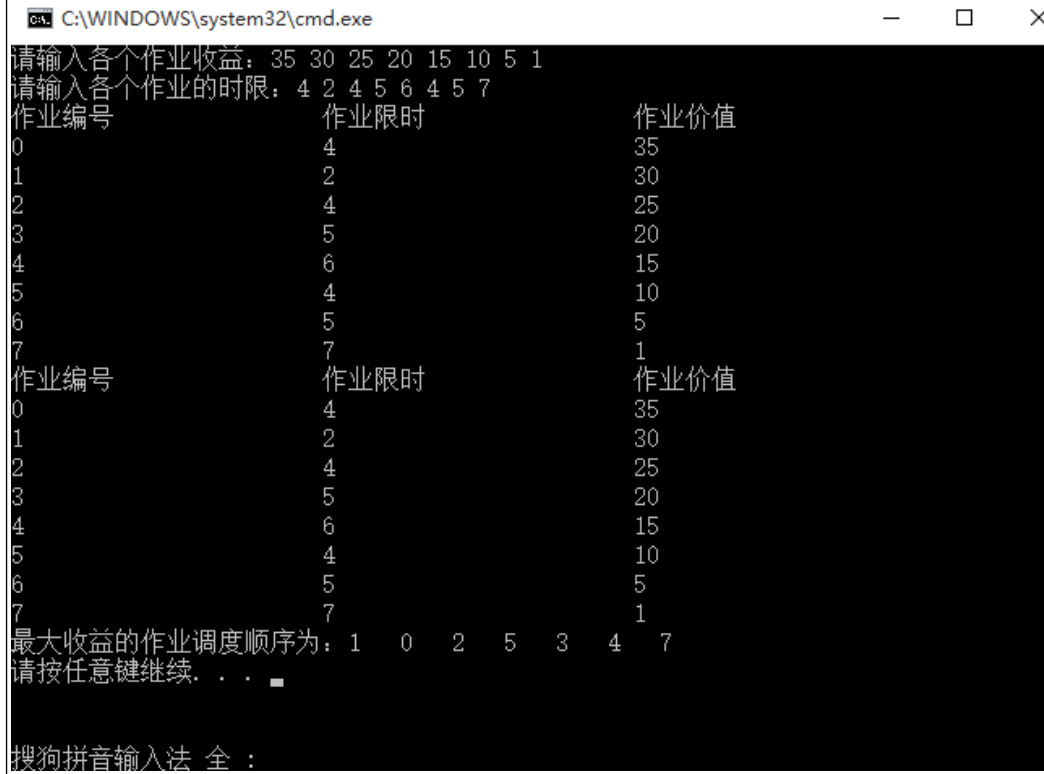
for(int i=0;i<w;i++){

    double x=s+time[A[i]];

```

```
if(x<t){  
  
    s=s+time[A[i]];cout<<" "<<A[i]<<" ";}  
  
}
```

## 五、试验成果截图



```
C:\WINDOWS\system32\cmd.exe  
请输入各个作业收益: 35 30 25 20 15 10 5 1  
请输入各个作业的时限: 4 2 4 5 6 4 5 7  
作业编号      作业限时      作业价值  
0              4              35  
1              2              30  
2              4              25  
3              5              20  
4              6              15  
5              4              10  
6              5              5  
7              7              1  
作业编号      作业限时      作业价值  
0              4              35  
1              2              30  
2              4              25  
3              5              20  
4              6              15  
5              4              10  
6              5              5  
7              7              1  
最大收益的作业调度顺序为: 1 0 2 5 3 4 7  
请按任意键继续. . .  
搜狗拼音输入法 全 :
```

## 六、试验总结

贪心算法设计的关键是贪心方略的选择。

## 试验三 动态规划法求多段图问题

### 一、 试验目的

- 1.掌握动态规划算法的基本思想

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：

<https://d.book118.com/245312233334011230>