

# 《算法设计与分析》期终试卷 1

考生须知

\*若试卷中试题字迹不清，考生可以在审题时举手请求解释，由考务人员加以说明。涉及题意理解问题，则不得提问且考务人员不予解答。

\*考生上机编程时应在指定目录下工作，并请每隔 5 分钟存盘一次。发生机器故障时由考务人员确认补给修复时间，且最长不超过 10 分钟。

\*对考生答题测试有严格时间限制，若超时则该测试项判为 0 分。考生应注意优化算法。

\*考生应严格遵守考场规则，不得违纪。

\*考试时间为 8 时 30 分至 11 时 30 分，计 180 分钟。

## 试题 1、乘法表问题

### ★问题描述:

定义于字母表 $\Sigma=\{a, b, c\}$ 上的乘法表如下

	a	b	c
a	b	b	a
b	c	b	a
c	a	c	c

依此乘法表，对任一定义于 $\Sigma$ 上的字符串，适当加括号后得到一个表达式。例如，对于字符串  $x=bbbbba$ ，它的一个加括号表达式为  $(b(bb))(ba)$ 。依乘法表，该表达式的值为  $a$ 。试设计一个算法，对任一定义于 $\Sigma$ 上的字符串  $x = x_1x_2 \cdots x_n$ ，计算有多少种不同的加括号方式，使由  $x$  导出的加括号表达式的值为  $a$ 。

### ★编程任务:

对于给定的字符串  $x = x_1x_2 \cdots x_n$ ，计算有多少种不同的加括号方式，使由  $x$  导出的加括号表达式的值为  $a$ 。

### ★数据输入:

由文件 `input.txt` 提供输入数据。文件的第 1 行中给出一个字符串。

### ★结果输出:

程序运行结束时，将计算结果输出到文件 `output.txt` 中。文件的第 1 行中的数是计算出的加括号方式数。

输入文件示例  
`input.txt`  
bbbbba

输出文件示例  
`output.txt`  
6

## 试题 2、工作分配问题

### ★问题描述:

设有  $n$  件工作分配给  $n$  个人。将工作  $i$  分配给第  $j$  个人所需的费用为  $c_{ij}$ 。试设计一个算法，为每一个人都分配 1 件不同的工作，并使总费用达到最小。

### ★编程任务:

设计一个算法，对于给定的工作费用，计算最佳工作分配方案，使总费用达到最小。

### ★数据输入:

由文件 input.txt 给出输入数据。第一行有 1 个正整数  $n$  ( $1 \leq n \leq 30$ )。接下来的  $n$  行，每行  $n$  个数，表示工作费用。

### ★结果输出:

将计算出的最小总费用输出到文件 output.txt。

#### 输入文件示例

```
input.txt
3
10 2 3
2 3 4
3 4 5
```

#### 输出文件示例

```
output.txt
9
```

### 试题 3、飞行员配对方案问题

#### ★问题描述:

第二次世界大战时期，英国皇家空军从沦陷国征募了大量外籍飞行员。由皇家空军派出的每一架飞机都需要配备在航行技能和语言上能互相配合的 2 名飞行员，其中 1 名是英国飞行员，另 1 名是外籍飞行员。在众多的飞行员中，每一名外籍飞行员都可以与其他若干名英国飞行员很好地配合。如何选择配对飞行的飞行员才能使一次派出最多的飞机。对于给定的外籍飞行员与英国飞行员的配合情况，试设计一个算法找出最佳飞行员配对方案，使皇家空军一次能派出最多的飞机。

#### ★编程任务:

对于给定的外籍飞行员与英国飞行员的配合情况，编程找出一个最佳飞行员配对方案，使皇家空军一次能派出最多的飞机。

#### ★数据输入:

由文件 input.txt 提供输入数据。文件第 1 行有 2 个正整数  $m$  和  $n$ 。 $n$  是皇家空军的飞行员总数( $n < 100$ )； $m$  是外籍飞行员数。外籍飞行员编号为  $1 \sim m$ ；英国飞行员编号为  $m+1 \sim n$ 。接下来每行有 2 个正整数  $i$  和  $j$ ，表示外籍飞行员  $i$  可以和英国飞行员  $j$  配合。文件最后以 2 个 -1 结束。

#### ★结果输出:

程序运行结束时，将最佳飞行员配对方案输出到文件 output.txt 中。第 1 行是最佳飞行员配对方案一次能派出的最多的飞机数  $M$ 。接下来  $M$  行是最佳飞行员配对方案。每行有 2 个正整数  $i$  和  $j$ ，表示在最佳飞行员配对方案中，飞行员  $i$  和飞行员  $j$  配对。

如果所求的最佳飞行员配对方案不存在，则输出 ‘No Solution!’。

#### 输入文件示例

```
input.txt
5 10
1 7
1 8
2 6
2 9
2 10
3 7
3 8
4 7
4 8
5 10
-1 -1
```

#### 输出文件示例

```
output.txt
4
1 7
2 9
3 8
5 10
```