



# CCF青少年计算机程序设计 评级标准

中国计算机学会  
2014年7月



# CCF 青少年计算机程序设计 评级标准

CCF

中国计算机学会  
2014年7月

CCCF

# 前言

算法设计是计算机科学的核心, 计算机程序设计是学习计算机科学的基本技能。以算法和数据结构为核心, 运用数学知识构建合适的模型, 并采用计算机程序设计语言编写程序来解决实际问题, 对于学习者能力训练非常有益。通过计算机程序设计的学习, 可以让青少年了解计算机科学, 热爱计算机科学, 用计算机解决实际问题。

1984年, 邓小平同志提出“计算机的普及要从娃娃抓起”, 这促进了中小学计算机教育的发展, 也由此, CCF 创建了 NOI。三十年过去了, 我国青少年计算机科学教育取得了很大的发展, 为了科学地引领青少年学习程序设计知识, 激发青少年学习、探索计算机科学的奥秘, 亟需一个能为青少年入门、提供学习支持、为社会提供甄别的参考评级机制。

制定青少年程序设计评级标准, 有利于规范和引导青少年学习计算机程序设计知识, 有助于计算机教学的开展, 有利于评价青少年计算机程序设计的水平, 有利于发现和培养未来学习计算机科学的人才。

为此, 2013年5月, 中国计算机学会(CCF)设立了《青少年计算机程序设计评级标准》课题, 并组织了全国青少年信息学奥林匹克竞赛(NOI)部分优秀教师对该课题进行研究, 通过一年多的探索和研讨, 形成了目前的评级标准体系, 现予以发布。

参与该课题研究的核心人员有长沙雅礼中学特级教师朱全民组长，福州第一中学特级教师陈颖，常州高级中学高级教师曹文，东北育才学校高级教师邱桂香，杭州学军中学高级教师徐先友，绍兴柯桥中学高级教师吴建锋，绍兴市第一中学特级教师陈合力，北京第八十中高级教师贾志勇。NOI 科学委员会委员参与了修订意见，部分中学教师也参与了一些工作，对他们的贡献表示感谢。

中国计算机学会

2014 年 7 月

# 目录

## 前言

一级标准 .....	1
1.1 定义 .....	1
1.2 知识要求 .....	1
1.3 能力要求 .....	2
1.4 评价方法 .....	2
1.5 题例 .....	2
二级标准 .....	4
2.1 定义 .....	4
2.2 知识要求 .....	4
2.3 能力要求 .....	4
2.4 评价方法 .....	5
2.5 题例 .....	5
三级标准 .....	7
3.1 定义 .....	7
3.2 知识要求 .....	7
3.3 能力要求 .....	7
3.4 评价方法 .....	8
3.5 题例 .....	8
四级标准 .....	10
4.1 定义 .....	10
4.2 知识要求 .....	10
4.3 能力要求 .....	10

4.4	评价方法 .....	11
4.5	题例 .....	11
五级标准	.....	14
5.1	定义 .....	14
5.2	知识要求 .....	14
5.3	能力要求 .....	15
5.4	评价方法 .....	15
5.5	题例 .....	15
六级标准	.....	18
6.1	定义 .....	18
6.2	知识要求 .....	18
6.3	能力要求 .....	18
6.4	评价方法 .....	19
6.5	题例 .....	19
七级标准	.....	24
7.1	定义 .....	24
7.2	知识要求 .....	24
7.3	能力要求 .....	24
7.4	评价方法 .....	25
7.5	题例 .....	25
八级标准	.....	29
8.1	定义 .....	29
8.2	知识要求 .....	29
8.3	能力要求 .....	29
8.4	评价方法 .....	30
8.5	题例 .....	30
九级标准	.....	35

9.1	定义 .....	35
9.2	知识要求 .....	35
9.3	能力要求 .....	35
9.4	评价方法 .....	36
9.5	题例 .....	36
十级标准	.....	40
10.1	定义 .....	40
10.2	知识要求 .....	40
10.3	能力要求 .....	41
10.4	评价方法 .....	41
10.5	题例 .....	41
积分计算表	.....	47
1~10 级评分表	.....	48

# 一级标准

## 1.1 定义

了解什么是计算机程序，能够编写计算机程序解决简单问题。

## 1.2 知识要求

1. 程序的基本结构。
2. 标识符与关键字。
3. 基本数据类型。
4. 常量和变量。
5. 算术表达式和关系表达式。
6. 整除，求余运算，常用数学函数。
7. 赋值语句，输入输出语句，复合语句，条件语句（不嵌套），循环语句（不嵌套）。

## 1.3 能力要求

1. 能用自然语言描述解决简单问题的方法和步骤。
2. 能用顺序、分支、循环语句实现 A 中的方法和步骤，编写完整程序。
3. 初步理解算法的意义。

## 1.4 评价方法

1. NOIP 初赛 60 分以上。
2. 本级不颁发证书。

## 1.5 题例

**试题名：**求最小、最大数

**试题描述：**

给出  $N$  个数，请找出这  $N$  个数的最小数和最大数。

**输入数据：**

第 1 行，一个整数  $n$ ， $n \leq 1000$ 。

接下来的一行，包含  $n$  个数，两个数之间用一个空格分隔。

**输出数据:**

第 1 行, 最小数。

第 2 行, 最大数。

**输入样例:**

```
4
1 2 3 4
```

**输出样例:**

```
1
4
```

**参考题解:**

主要用两个变量 `min` 和 `max` 分别记录当前最小最大值, 然后用循环边读数, 边将该数与 `min` 和 `max` 比较, 每次都更新最小最大值即可。

---

# 二级标准

## 2.1 定义

了解什么是算法，能够用程序设计语言实现简单算法，解决问题。

## 2.2 知识要求

1. 逻辑表达式。
2. 条件嵌套，循环嵌套，数组。
3. 枚举，简单排序，简单查找算法。
4. 素数与合数，最大公约数，最小公倍数，互质数。

## 2.3 能力要求

1. 能用简单枚举算法解决实际问题，能对数据进行简单排序和查找。
2. 具备独立编写和调试简短程序的能力。

## 2.4 评价方法

1. NOIP 初赛 80 分以上。
2. 本级不颁发证书。

## 2.5 题例

**试题名：**求第 k 小数

**试题描述：**

给出 N 个数，请找出第 K 小的数并输出该数值。

**输入数据：**

第 1 行，二个整数 n,k,  $n,k \leq 1000$ 。

接下来的一行，包含 n 个数，两个数之间用 1 个空格分隔。

**输出数据：**

只有 1 行，为第 k 小数。

**输入样例：**

```
4 3  
1 2 3 4
```

**输出样例:**

3

**参考题解:**

先对这  $k$  个数按从小到大顺序排序, 则第  $k$  小的数就是数组中第  $k$  个位置的数, 直接输出该数即可。



---

# 三级标准

## 3.1 定义

具有较强的程序实现能力,使用一种计算机程序设计语言编写程序,解决问题。

## 3.2 知识要求

1. 数制及其转化,信息编码,位运算。
2. 字符串类型。
3. 子程序。
4. 递归。
5. 逻辑运算,整数的质因数分解,随机函数。
6. 筛选法,欧几里德算法。

## 3.3 能力要求

1. 全面掌握一种计算机程序设计语言。

2. 具有运用简单数学知识编写程序解决问题的能力。

## 3.4 评价方法

1. 进入 NOIP 复赛。

## 3.5 题例

**试题名：**分解质因素

**试题描述：**

给一个整数  $N$ ，将  $N$  写成质因数的乘积。

**输入数据：**

一个整数  $n$ ， $n \leq 100000$ 。

**输出数据：**

质因数的乘积表达式（请将质因数按从小到大顺序输出）。

**输入样例：**

12

**输出样例：**

12=2\*2\*3

**参考题解：**

本题可先求出  $N$  平方根范围内的素数放入数组，然后用循环将  $N$  逐个整除每个素数，若能整除，则输出该素数即可。



---

# 四级标准

## 4.1 定义

了解几种常用的算法，并运用这些算法编写程序，解决问题。

## 4.2 知识要求

1. 结构类型，文件操作。
2. 数据类型的内在含义。
3. 贪心法，递推，回溯法，模拟算法。
4. 简单的字符串处理。
5. 集合及集合的运算，加法原理与乘法原理，简单的排列和组合。

## 4.3 能力要求

1. 能根据实际问题选择合适的数据类型。
2. 能运用贪心、递推、回溯、模拟等算法解决实际问题。

3. 能独立设计简单的测试数据，测试自己程序的正确性。

## 4.4 评价方法

1. 与 NOIP 复赛相衔接。采用上机编程考核方式。普及组复赛共 4 个试题，考核时间：3 小时；提高组复赛分两次测试，每次 3 个试题，考 4 小时，共 6 道题，考 8 小时。
2. 在 NOIP 普及组(或提高组)复赛中成绩列全国前 50%。

## 4.5 题例

**试题名：**校门外的树

**文件名：**tree

**试题描述：**

某校大门外长度为  $L$  的马路上有一排树，每两棵相邻的树之间的间隔都是 1 米。我们可以把马路看成一个数轴，马路的一端在数轴 0 的位置，另一端在  $L$  的位置；数轴上的每个整数点，即 0, 1, 2, …,  $L$ ，都种有一棵树。

由于马路上有一些区域要用来建地铁。这些区域用它们在数轴上的起始点和终止点表示。已知任一区域的起始点和终止点的坐标都是整数，区域之间可能有重合的部分。现在要把这些区域

中的树（包括区域端点处的两棵树）移走。你的任务是计算将这些树都移走后，马路上还有多少棵树。

**输入数据：**

输入文件名为 `tree.in`。

第一行有两个整数  $L$  和  $M$ ， $L$  代表马路的长度， $M$  代表区域的数目， $L$  和  $M$  之间用一个空格隔开。接下来的  $M$  行每行包含两个不同的整数，用一个空格隔开，表示一个区域的起始点和终止点的坐标。

**输出数据：**

输入文件名为 `tree.out`。

一个整数，表示马路上剩余的树的数目。

**输入输出样例：**

<code>tree.in</code>	<code>tree.out</code>
500 3	298
150 300	
100 200	
470 471	

**数据规模：**

$1 \leq L \leq 10000$ ， $1 \leq M \leq 100$

对于 20% 的数据，区域之间没有重合的部分；

对于其它的数据，区域之间有重合的情况。

### 参考题解:

本题可采用模拟法。

用一个一维数组来存放马路各位置点当前的状态:设 1 表示当前位置的状态为树, 0 来表示当前位置的状态为地铁, 根据题目所给的地铁区域, 可以将对应的状态区域置 0, 当处理完所有的地铁区间后, 再重新对整条公路的状态遍历一遍, 用一个记数变量记录状态为 1 的个数, 即为问题所求。

---

# 五级标准

## 5.1 定义

掌握简单数据结构知识，并结合已学算法和数学知识编写程序，解决问题。

## 5.2 知识要求

1. 指针类型。
2. 一般线性表，队列，堆栈，二叉树的存储和遍历。
3. 排列与组合，高精度数值处理。
4. 二分算法，快速排序，深度优先搜索，宽度优先搜索，简单动态规划。
5. 圆排列，可重集排列，鸽笼原理，素因数分解，幂函数，指数函数，对数函数，三角函数，模运算，不等式基础知识。

## 5.3 能力要求

1. 能运用常用算法和简单数据结构解决实际问题。
2. 能从算法本质出发，分析相关算法之间的本质联系。
3. 具备初步的数学建模能力。

## 5.4 评价方法

1. 与 NOIP 普及组复赛相衔接。采用上机编程考核方式，共 4 个试题，考核时间：3 小时。
2. 在 NOIP 普及组复赛中成绩列全国前 20%。

## 5.5 题例

**试题名：**摆花

**文件名：**flower

**试题描述：**

小明的花店新开张，为了吸引顾客，他想在花店的门口摆上一排花，共  $m$  盆。通过调查顾客的喜好，小明列出了顾客最喜欢的  $n$  种花，从 1 到  $n$  标号。为了在门口展出更多种花，规定第  $i$  种花不能超过  $a_i$  盆，摆花时同一种花放在一起，且不同种类的花需按标号从小到大的顺序依次摆列。

试编程计算，一共有多少种不同的摆花方案。

### 输入数据:

输入文件 `flower.in`，共 2 行。

第一行包含两个正整数  $n$  和  $m$ ，中间用一个空格隔开。

第二行有  $n$  个整数，每两个整数之间用一个空格隔开，依次表示  $a_1$ 、 $a_2$ 、..... $a_n$ 。

### 输出数据:

输出文件名为 `flower.out`。

输出只有一行，一个整数，表示有多少种方案。注意：因为方案数可能很多，请输出方案数对 1000007 取模的结果。

### 输入输出样例:

<code>flower.in</code>	<code>flower.out</code>
2 4	2
3 2	

样例说明：有 2 种摆花的方案，分别是(1, 1, 1, 2)，(1, 1, 2, 2)。括号里的 1 和 2 表示两种花，比如第一个方案是前三个位置摆第一种花，第四个位置摆第二种花。

### 数据范围:

对于 20% 数据，有  $0 < n \leq 8$ ， $0 < m \leq 8$ ， $0 \leq a_i \leq 8$ ；

对于 50% 数据，有  $0 < n \leq 20$ ， $0 < m \leq 20$ ， $0 \leq a_i \leq 20$ ；

对于 100% 数据，有  $0 < n \leq 100$ ， $0 < m \leq 100$ ， $0 \leq a_i \leq 100$ 。

**参考题解：**

本题可采用简单动态规划。设  $f[i,j]$  表示前  $i$  种花放在前  $j$  个花盆里的最大方案数，那么，

$$f[i,j]=\max_{0\leq x\leq \min(a[i],j)}(f[i-1,j-x]+1)$$

状态转移方程的含义为：前  $j$  个花盆里，第  $i$  种花最少放 0 个，最多放  $\min\{a[i],j\}$  个。

其中  $\min\{a[i],j\}$  表示第  $i$  种花的数目和花盆的数目中较少的那一个。

边界条件：  $f[1,i]=1 (i\leq a[1])$ 。

---

# 六级标准

## 6.1 定义

掌握基本的数据结构知识,能够根据实际需求设计算法编写程序,解决问题。

## 6.2 知识要求

1. 树、图的存储。
2. 哈希表、集合数据结构。
3. 图的最短路、生成树算法,有向图的拓扑排序算法。
4. 动态规划常见模型,分治策略,各种排序算法。
5. 可重集组合,二项式定理,数列与级数,归纳与递推,容斥原理,函数的连续性、函数的单调性和极值。

## 6.3 能力要求

1. 能对一些算法和数据结构估算时间复杂度和空间复杂

- 度。
2. 能根据实际问题的模型选择合适的算法和数据结构来解决问题。
  3. 具备知识收集和知识管理的能力。

## 6.4 评价方法

1. 与 NOIP 提高组复赛相衔接。采用上机编程考核方式，分两次测试，每次 3 个试题，考 4 小时。共 6 道题，考 8 小时。
2. 在 NOIP 提高组复赛中成绩列全国前 20%。

## 6.5 题例

**试题名：**最优贸易

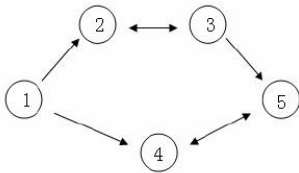
**文件名：**trade

**试题描述：**

C 国有  $n$  个大城市和  $m$  条道路，每条道路连接这  $n$  个城市中的某两个城市。任意两个城市之间最多只有一条道路直接相连。这  $m$  条道路中有一部分为单向通行的道路，一部分为双向通行的道路，双向通行的道路在统计条数时也计为 1 条。C 国幅员辽阔，各地的资源分布情况各不相同，这就导致了同一种商

品在不同城市的价格不一定相同。但是，同一种商品在同一个城市的买入价和卖出价始终是相同的。

商人阿龙来到 C 国旅游。当他得知同一种商品在不同城市的价格可能会不同这一信息之后，便决定在旅游的同时，利用商品在不同城市中的差价赚回一点旅费。设 C 国  $n$  个城市的标号从  $1 \sim n$ ，阿龙决定从 1 号城市出发，并最终在  $n$  号城市结束自己的旅行。在旅游的过程中，任何城市可以重复经过多次，但不要求经过所有  $n$  个城市。阿龙通过这样的贸易方式赚取旅费：他会选择一个经过的城市买入他最喜欢的商品——水晶球，并在之后经过的另一个城市卖出这个水晶球，用赚取的差价当做旅费。由于阿龙主要是来 C 国旅游，他决定这个贸易只进行最多一次，当然，在赚不到差价的情况下他就无需进行贸易。假设 C 国有 5 个大城市，城市的编号和道路连接情况如下图，单向箭头表示这条道路为单向通行，双向箭头表示这条道路为双向通行。



假设  $1 \sim n$  号城市的水晶球价格分别为 4, 3, 5, 6, 1。阿龙可以选择如下一条线路： $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$ ，并在 2 号城市以 3 的价格买入水晶球，在 3 号城市以 5 的价格卖出水晶球，赚取的旅费数为 2。阿龙也可以选择如下一条线路  $1 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 5$ ，并在第 1 次到达 5 号城市时以 1 的价格买入水晶球，在第 2 次到

达 4 号城市时以 6 的价格卖出水晶球，赚取的旅费数为 5。现在给出  $n$  个城市的水晶球价格， $m$  条道路的信息（每条道路所连接的两个城市的编号以及该条道路的通行情况）。

请你告诉阿龙，他最多能赚取多少旅费。

### 输入数据：

输入文件为 `trade.in`。

第一行包含 2 个正整数  $n$  和  $m$ ，中间用一个空格隔开，分别表示城市的数目和道路的数目。第二行  $n$  个正整数，每两个整数之间用一个空格隔开，按标号顺序分别表示这  $n$  个城市的商品价格。接下来  $m$  行，每行有 3 个正整数， $x$ ， $y$ ， $z$ ，每两个整数之间用一个空格隔开。如果  $z=1$ ，表示这条道路是城市  $x$  到城市  $y$  之间的单向道路；如果  $z=2$ ，表示这条道路为城市  $x$  和城市  $y$  之间的双向道路。

### 输出数据：

输出文件 `trade.out`。

共 1 行，包含 1 个整数，表示最多能赚取的旅费。如果没有进行贸易，则输出 0。

### 输入输出样例：

<code>trade.in</code>	<code>trade.out</code>
5 5	5
4 3 5 6 1	

1 2 1  
1 4 1  
2 3 2  
3 5 1  
4 5 2

### 数据范围:

输入数据保证 1 号城市可以到达 n 号城市。

对于 10%的数据,  $1 \leq n \leq 6$ 。

对于 30%的数据,  $1 \leq n \leq 100$ 。

对于 50%的数据, 不存在一条旅游路线, 可以从一个城市出发, 再回到这个城市。

对于 100%的数据,  $1 \leq n \leq 100000$ ,  $1 \leq m \leq 500000$ ,  $1 \leq x, y \leq n$ ,  $1 \leq z \leq 2$ ,  $1 \leq$ 各城市水晶球价格 $\leq 100$ 。

### 参考题解:

此题方法有很多, 下面介绍两种方法。

方法 1:

将图中圈缩成点, 变成一个有向无环图。

设  $f(i)$  表示  $i$  点的最低买入价,  $g(i)$  表示  $i$  点的最高卖出价。

从左至右做简单动态规划,  $f(i) = \min\{f[j], a[i]\}$ , 其中  $j$  到  $i$  有弧, 意思是当前的最低买入价, 要么是前面城市的最低买入价, 要么就是当前城市的价格。

然后从右到左再进行简单动态规划。  $f(i) = \max\{f[j], a[i]\}$ , 其

中  $i$  到  $j$  有弧，意思是当前的最高卖出价，要么是后面城市的最低卖出价，要么就是当前城市的价格。

答案就是对所有城市求出最大差价，即  $\text{ans}=\max\{g(i)-f(i)\}$ 。

方法 2:

从起始城市开始，到目标城市结束，沿着正向弧用 spfa 求出每个城市的最大买入价。

从目标城市开始，到起始城市结束，沿着逆向弧用 spfa 求出每个城市的最大卖出价。

答案就是对所有城市求出最大差价，即  $\text{ans}=\max\{g(i)-f(i)\}$ 。

---

# 七级标准

## 7.1 定义

综合运用算法和数据结构编写程序，解决问题。

## 7.2 知识要求

1. 并查集、线段树、哈夫曼树、二叉排序树、二叉堆。
2. 图的连通性算法，最短路、最小生成树的优化算法，二分图的构造、判定及匹配，搜索算法的优化，扩展欧几里德算法。
3. 中国剩余定理，剩余类，概率基础知识，解析几何基础知识。

## 7.3 能力要求

1. 能根据时间和空间复杂度的要求灵活构造算法和数据

结构解决实际问题。

2. 具备较强的程序代码实现能力。
3. 具备较强的归纳、总结和表达能力。

## 7.4 评价方法

1. 与 NOIP 提高组复赛相衔接。采用上机编程考核方式，分两次测试，每次 3 个试题，考 4 小时。共 6 道题，考 8 小时。
2. 在 NOIP 提高组复赛中成绩列全国前 10%。

## 7.5 题例

**试题名：**关押罪犯

**文件名：**prison

**试题描述：**

S 城现有两座监狱，一共关押着  $N$  名罪犯，编号分别为  $1\sim N$ 。他们之间的关系自然也极不和谐。很多罪犯之间甚至积怨已久，如果客观条件具备则随时可能爆发冲突。我们用“怨气值”（一个正整数）来表示某两名罪犯之间的仇恨程度，怨气值越大，则这两名罪犯之间的积怨越多。如果两名怨气值为  $c$  的罪犯被关押在同一监狱，他们俩之间会发生摩擦，并造成影响力为  $c$  的冲突

事件。

每年年末,警察局会将本年内监狱中的所有冲突事件按影响力从大到小排成一个列表,然后上报到 S 城 Z 市长那里。公务繁忙的 Z 市长只会去看列表中的第一个事件的影响力,如果影响很坏,他就会考虑撤换警察局长。

在详细考察了 N 名罪犯间的矛盾关系后,警察局长觉得压力巨大。他准备将罪犯们在两座监狱内重新分配,以求产生的冲突事件影响力都较小,从而保住自己的乌纱帽。假设只要处于同一监狱内的某两个罪犯间有仇恨,那么他们一定会在每年的某个时候发生摩擦。那么,应如何分配罪犯,才能使 Z 市长看到的那个冲突事件的影响力最小?这个最小值是多

#### 输入数据:

输入文件名为 `prison.in`。

输入文件的每行中两个数之间用一个空格隔开。

第一行为两个正整数 N 和 M,分别表示罪犯的数目以及存在仇恨的罪犯对数。

接下来的 M 行每行为三个正整数  $a_j, b_j, c_j$ ,表示  $a_j$  号和  $b_j$  号罪犯之间存在仇恨,其怨气值为  $c_j$ 。数据保证  $a_j \leq b_j \leq N$ ,  $0 < c_j \leq 1,000,000,000$ ,且每对罪犯组合只出现一次

#### 输出数据:

输出文件 `prison.out`。

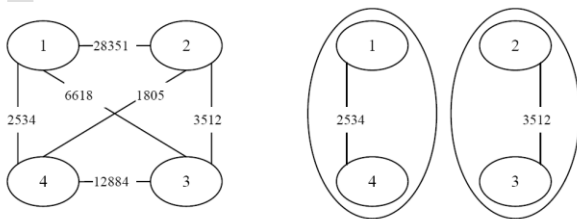
共 1 行,为 Z 市长看到的那个冲突事件的影响力。如果本年内监狱中未发生任何冲突事件,请输出 0。

**输入输出样例:**

prison.in	prison.out
4 6	3512
1 4 2534	
2 3 3512	
1 2 28351	
1 3 6618	
2 4 1805	
3 4 12884	

**样例说明:**

罪犯之间的怨气值如下图所示，下图所示为罪犯的分配方法，市长看到的冲突事件影响力是 3512（由 2 号和 3 号罪犯引发）。其他任何分法都不会比这个分法更优。

**数据范围:**

对于 30% 的数据有  $N \leq 15$ 。

对于 70% 的数据有  $N \leq 2000$ ， $M \leq 50000$ 。

对于 100% 的数据有  $N \leq 20000$ ,  $M \leq 100000$ 。

### 参考题解:

此题方法有很多, 下面介绍两种方法。

方法 1: 贪心分配法。

假设最后的答案是其中一条边  $g[i,j]$ , 则小于  $g[i,j]$  的所有边都可以当作不存在, 所以我们先把所有的边按长度排序。然后从大到小往下找, 每条边的两个点分到两个集合, 直到找到一条边的两个点都不能分到同一个集合为止, 则输出这条边的权值, 就是要求的答案。

现在关键是怎样方便快捷的来存储和分配这些点, 可使用并查集。每个集合除了父亲节点  $f[i]$ , 还有一个下标  $s[i]$ , 表示第  $i$  个点所在的集合 (0 或 1)。一开始查询一条边  $g[i,j]$  的时候, 如果两个点有相同的父亲, 而且在同一个集合中 (即  $s[i]=s[j]$ ), 则这条边就是要求的答案。如果两个点父亲相同, 而且不在同一个集合中 ( $s[i] \neq s[j]$ ), 则不做修改。如果父亲不相同, 则把这两个点合并, 把  $i$  的父亲指向  $j$ , 然后更新  $s$  的值。

方法 2: 二分法。

二分答案  $ans$ , 然后删除比答案小的边, 看剩下的图能否构成二分图, 能构成二分图, 表示这些比  $ans$  大的冲突都能分成两个监狱关押, 可以继续二分, 缩小  $ans$ ; 否则表示不能关押, 只能扩大  $ans$ , 继续二分判断。

二分图的判定, 可以采用黑白染色法。

---

# 八年级标准

## 8.1 定义

掌握高级数据结构知识，能运用恰当算法编写程序，解决较复杂问题。

## 8.2 知识要求

1. 树状数组，字典树，优先队列，平衡树。
2. 网络流算法，复杂的分治思想，树形动态规划，状态压缩动态规划，二分图的匹配，启发式搜索。
3. 矩阵概念及其基本运算，线性方程组的解法，迭代法，费马小定理和欧拉定理，母函数。

## 8.3 能力要求

1. 能针对复杂问题建立清晰的数学模型。

2. 能运用数学知识、高级数据结构和算法解决复杂的问题。
3. 能根据需要，开展基于协作的学习和研究。

## 8.4 评价方法

1. 与 NOI、亚洲与太平洋信息学奥林匹克竞赛（APIO）、信息学奥林匹克国家队选拔赛（CTSC）、全国信息学奥林匹克冬令营（WC）等竞赛相衔接。采用上机编程考核方式。NOI、CTSC 两试，每试 3 个试题，5 小时；APIO、WC 一试，3 个试题，5 小时。
2. NOI 铜牌或累计积分 250 分以上的选手可获得八级资格。积分折算详见“积分计算表”。

## 8.5 题例

**试题名：**能量采集

**文件名：**energy

**试题描述：**

栋栋有一块长方形的地，他在地上种了一种能量植物，这种植物可以采集太阳光的能量。在这些植物采集能量后，栋栋再使用一个能量汇集机器把这些植物采集到的能量汇集到一起。

栋栋的植物种得非常整齐，一共有  $n$  列，每列有  $m$  棵，植

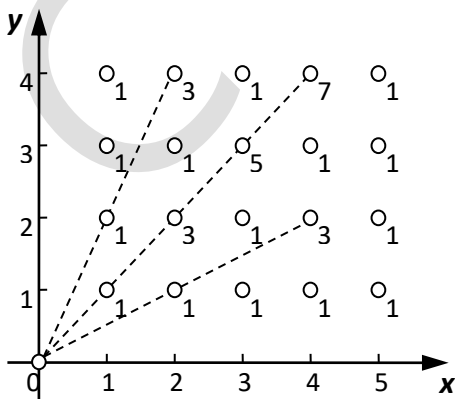
物的横竖间距都一样，因此对于每一棵植物，栋栋可以用一个坐标 $(x, y)$ 来表示，其中  $x$  的范围是 1 至  $n$ ，表示是在第  $x$  列， $y$  的范围是 1 至  $m$ ，表示是在第  $x$  列的第  $y$  棵。

由于能量汇集机器较大，不便移动，栋栋将它放在了一个角上，坐标正好是 $(0, 0)$ 。

能量汇集机器在汇集的过程中有一定的能量损失。如果一棵植物与能量汇集机器连接而成的线段上有  $k$  棵植物，则能量的损失为  $2k + 1$ 。例如，当能量汇集机器收集坐标为 $(2, 4)$ 的植物时，由于连接线段上存在一棵植物 $(1, 2)$ ，会产生 3 的能量损失。注意，如果一棵植物与能量汇集机器连接的线段上没有植物，则能量损失为 1。现在要计算总的能量损失。

下图给出了一个能量采集的例子，其中  $n = 5$ ， $m = 4$ ，一共有 20 棵植物，在每棵植物上标明了能量汇集机器收集它的能量时产生的能量损失。

在这个例子中，总共产生了 36 的能量损失。



**输入数据:**

输入文件 energy.in。

仅包含一行，为两个整数  $n$  和  $m$ 。

**输出数据:**

输出文件 energy.out。

包含一个整数，表示总共产生的能量损失。

**输入输出样例:**

energy.in	energy.out
5 4	36

**数据范围:**

对于 10% 的数据:  $1 \leq n, m \leq 10$ ;

对于 50% 的数据:  $1 \leq n, m \leq 100$ ;

对于 80% 的数据:  $1 \leq n, m \leq 1000$ ;

对于 90% 的数据:  $1 \leq n, m \leq 10,000$ ;

对于 100% 的数据:  $1 \leq n, m \leq 100,000$ 。

**参考题解:**

坐标为  $(i,j)$  的植物的能量损失为  $2 * \gcd(i,j) - 1$ 。由于  $n, m$  最大达到了 100000，不能直接枚举每个坐标  $(i,j)$ 。对于每个点  $(x,y)$ ，它连线上的点的个数为  $\gcd(x,y)$ ，证明如下：

对于一对 $(x,y)$ , 且  $\gcd(x,y)=1$ , 则 $(d*x,d*y)(d*x \leq n, d*y \leq m)$ 与  $x,y$  在同一条连线上。而 $(d*x,d*y)$ 的连线上有  $d$  个点。又因为  $\gcd(d*x,d*y)=d$ , 且每对 $(x',y')$ 都可以表示为 $(d*x,d*y)$ 的形式, 所以上述分析成立。

然后原问题转化为求所有  $\gcd(i,j)$ 的和。

再次观察发现, 所求问题的域较大(为  $n*m$ ), 而答案的域较小(为  $\min(n,m)$ ), 所以一个很重要的转化就是对于与每一个  $d(<\min(m,n))$ , 求出有多少对  $\gcd(x,y)=d$ , 而对于这个问题, 可以利用容斥原理解决。

首先, 令  $f[d]$ 表示最大公约数为  $d$  的 $(x,y)$ 的对数, 公约数里有  $d$  的 $(x,y)$ 对数共有  $(\lfloor n/d \rfloor * \lfloor m/d \rfloor)$ 对, 然后要排出里面公约数有大于  $d$  的 $(x,y)$ 对数, 要排出多少个? 答案是:

$f[d*2]+f[d*3]+...+f[d*k](d*k \leq \min(n,m))$ 个。这样, 我们可以用逆推的方式求出所有的  $\gcd(x,y)$ 之和, 然后代入推出的公式中就可以得到答案, 时间复杂度为  $O(n \lg n)$ 。

具体实现为: 令  $f(x)$ 表示  $\gcd(i,j)=x$  的有序数对 $(i,j)$ 的个数,  $g(x)$ 表示  $x|\gcd(i,j)$ 的有序数对 $(i,j)$ 的个数, 则:

$$g(x) = \left\lfloor \frac{n}{x} \right\rfloor * \left\lfloor \frac{m}{x} \right\rfloor$$

$$f(x) = g(x) - \sum_{k=2}^{\lfloor \frac{\min(n,m)}{x} \rfloor} f(kx)$$

倒序计算  $f(x)$ , 则

$$ans = \sum_{x=1}^{\min(n,m)} f(x) * (2 * x - 1)$$

时间复杂度:

$$O\left(\frac{n}{1} + \frac{n}{2} + \dots + \frac{n}{n}\right) = O(n \log n)$$

CCCF

---

# 九级标准

## 9.1 定义

具有对问题进行抽象和数学建模能力,能选用合适的数据结构 and 算法编写程序, 解决较难问题。

## 9.2 知识要求

1. 块状链表, 后缀数组, 后缀树, 复杂的线段树。
2. 动态规划优化, 模拟退火算法。
3. 计算几何基础知识 (点积、叉积、凸包、半平面等知识及应用), 数学期望。

## 9.3 能力要求

1. 能针对疑难问题建立清晰的数学模型。
2. 能灵活运用数学知识、高级数据结构和算法解决疑难问

题。

3. 具备发现问题、解决问题的探索研究能力。

## 9.4 评价方法

1. 与 NOI、APIO、CTSC、WC 等竞赛相衔接。采用上机编程考核方式。NOI、CTSC 两试，每试 3 个试题，5 小时；APIO、WC 一试，3 个试题，5 小时。
2. NOI 银牌或累计积分 500 分以上的选手可获得九级资格。积分折算详见“积分计算表”。

## 9.5 题例

**试题名：**直线和点

**文件名：**line

**试题描述：**

平面的  $n$  条直线将平面分割成了若干区域，给出  $m$  个点，求每个点所在区域的面积。为了防止出现面积无穷大的情况，有额外的四条直线框定了平面区域的大小，分别是  $x=L$ ， $y=L$ ， $x=-L$ ， $y=-L$ 。其中  $L$  是给定的正实数，所有的点都在这个框定的区域内。

另外为了防止精度问题，任意一个点到任意一条直线的距

离 $>10^{-7}$ 。

**输入数据:**

输入文件名为 `line.in`。

第一行两个正整数和一个正实数,  $n,m,L$ , 意义如上所述。

第  $2\sim n-1$  行每行三个实数  $A,B,C$  表示直线的方程为

$Ax+By+C=0$ 。

第  $n+2\sim n+m+1$  行每行两个实数  $x,y$  表示点的坐标。

**输出数据:**

输出文件名为 `line.out`。

按输入的顺序输出每个点所在的区域面积, 每个一行, 保留 2 位小数。

**输入输出样例:**

<code>line.in</code>	<code>line.out</code>
2 4 3	4.00
1 1 -1	8.50
-1 1 -1	8.50
0 2	15.00
-2 1	
2 1	
0 0	

**数据范围:**

对于 20% 的数据,  $n, m \leq 10$ 。

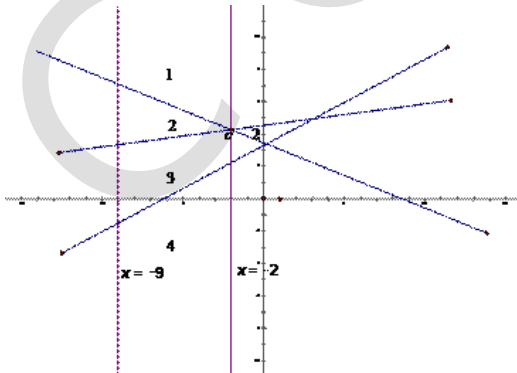
对于 40% 的数据,  $n, m \leq 300$ 。

对于 100% 的数据,  $n \leq 500, m \leq 100000$ 。

对于 100% 的数据, 输入数据的绝对值  $\leq 107$  且最多保留 2 位小数。

### 参考题解:

首先我们要确定每个点所在的区域位置, 我们使用扫描线的方法去处理这个问题。先求出所有直线两两之间的交点, 按它们的  $x$  轴坐标排序。用一条平行于  $y$  轴的扫描线从左向右扫描, 维护所有直线与这条扫描线交点的有序序列。如何维护呢? 只要当扫描线扫到某两条直线的交点时, 交换这两条直线的位置即可。维护了所有直线的有序序列之后, 我们就可以使用二分来比较轻松地计算出每个点所在的区域了。那如何计算面积呢?



我们将区域从上往下编号。我们发现, 当扫描线经过上面两

条直线的交点时，有 3 个区域发生的变化，分别是一号区域、二号区域和三号区域。一号区域的下边界“转弯”了，三号区域的上边界“转弯”了，对于这两个区域，只要用叉积把这条边的面积计入这个区域的总面积就好了。而我们发现，二号区域在经过这个交点后结束了，然后又出现了一个新的二号区域。所以我们要将原来的二号区域面积计算出来，然后更新那些在二号区域内的点的答案。

---

# 十级标准

## 10.1 定义

具有一定的提出问题、解决问题的能力，能构造算法与数据结构，解决开放性问题。

## 10.2 知识要求

1. 最小树形图，自动机，动态树，树套树，一般图的匹配。
2. 双重动态规划，基于连通性的动态规划，线性规划，极大极小搜索算法。
3. 三维计算几何，组合游戏中的 NIM 问题和 SG 函数，群的概念，置换群，Burnside 引理，Polya 原理，莫比乌斯反演定理，FFT。

## 10.3 能力要求

1. 具备创造性地运用数据结构和算法解决开放性问题的能力。
2. 具备很强的代码编写能力。
3. 具备提出问题、并开展相关研究的创新能力。

## 10.4 评价方法

1. 与 NOI、APIO、CTSC、WC 等竞赛相衔接。采用上机编程考核方式。NOI、CTSC 两试，每试 3 个试题，5 小时；APIO、WC 一试，3 个试题，5 小时。
2. NOI 金牌或累计积分 1000 分以上的选手可获得十级资格。积分折算详见“积分计算表”。

## 10.5 题例

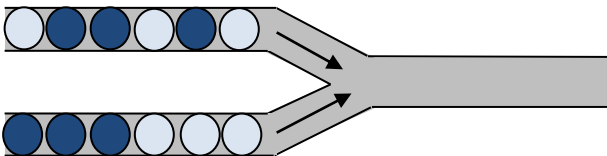
**试题名：**直线和点

**文件名：**ball

**试题描述：**管道取珠

管道取珠是小 X 很喜欢的一款游戏。在本题中，我们将考

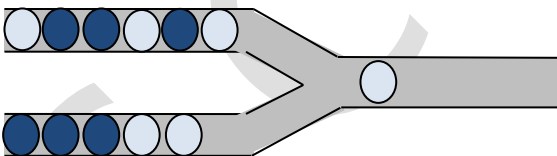
考虑该游戏的一个简单改版。游戏画面如图 1 所示：



(图 1)

游戏初始时，左侧上下两个管道分别有一定数量的小球（有深色球和浅色球两种类型），而右侧输出管道为空。每一次操作，可以从左侧选择一个管道，并将该管道中最右侧的球推入右边输出管道。

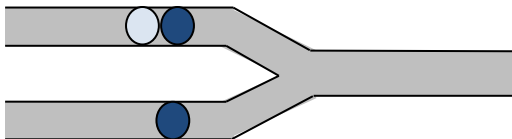
例如，我们首先从下管道中移一个球到输出管道中，将得到图 2 所示的情况。



(图 2)

假设上管道中有  $n$  个球，下管道中有  $m$  个球，则整个游戏过程需要进行  $n + m$  次操作，即将所有左侧管道中的球移入输出管道。最终  $n + m$  个球在输出管道中从右到左形成输出序列。

爱好数学的小 X 知道，他共有  $C(n+m, n)$  种不同的操作方式，而不同的操作方式可能导致相同的输出序列。举个例子，对于图 3 所示的游戏情形：



(图 3)

我们用 A 表示浅色球，B 表示深色球。并设移动上管道右侧球的操作为 U，移动下管道右侧球的操作为 D，则共有  $C(2+1,1)=3$  种不同的操作方式，分别为 UUD, UDU, DUU；最终在输出管道中形成的输出序列（从右到左）分别为 BAB, BBA, BBA。可以发现后两种操作方式将得到同样的输出序列。

假设最终可能产生的不同种类的输出序列共有 K 种，其中第 i 种输出序列的产生方式(即不同的操作方式数目)有  $a_i$  个。聪明的小 X 早已知道，

$$\sum_{i=1}^k a_i = C(n+m, n)$$

因此，小 X 希望计算得到

$$\sum_{i=1}^k a_i^2$$

你能帮助他计算这个值么？由于这个值可能很大，因此只需要输出该值对 1024523 的取模即可(即除以 1024523 的余数)。

说明：文中  $C(n+m, n)$  表示组合数。组合数  $C(a, b)$  等价于在 a 个不同的物品中选取 b 个的选取方案数。

**输入数据:**

输入文件 `ball.in`。

第一行包含两个整数  $n, m$ ，分别表示上下两个管道中球的数目。

第二行为一个 `AB` 字符串，长度为  $n$ ，表示上管道中从左到右球的类型。其中 `A` 表示浅色球，`B` 表示深色球。

第三行为一个 `AB` 字符串，长度为  $m$ ，表示下管道中的情形。

**输出数据:**

输出文件 `ball.out`

仅包含一行，即为  $\sum_{i=1}^k a_i^2$  除以 1024523 的余数。

**输入输出样例:**

<code>ball.in</code>	<code>ball.out</code>
2 1	5
AB	
B	

**样例说明:**

样例即为文中(图 3)。共有两种不同的输出序列形式，序列 `BAB` 有 1 种产生方式，而序列 `BBA` 有 2 种产生方式，因此答案为 5。

**数据范围：**

约 30% 的数据满足  $n, m \leq 12$ ;

约 100% 的数据满足  $n, m \leq 500$ 。

**参考题解：****题意：**

两个管道里面装有 AB 两种珠子，每次可以从任意一个管道末端取一个珠子直到取完，取出的珠子一共有 K 种不同的输出

序列，每种序列有  $A_i$  种取法，求  $\sum_{i=1}^k a_i^2$ 。

**算法：**

动态规划，时间复杂度： $O(N^3)$ 。

这道题比较复杂，初步分析感觉思路也难以一下子理顺。由于数据规模的庞大，K 和  $A_i$  都感觉是不可求不可记录类型的，所以只可能从  $A_i^2$  处下手。

顺向思考出现困难，不妨逆向思考。如果已知输出序列，如何求解。那么很简单，设  $F[i, j]$  表示上管道取 i 颗，下管道取 j 颗有多少种可行方案，然后按照输出序列进行转移即可，最后把答案平方。而现在并不知道输出序列是什么样子的，所以需要确定一个输出序列，所以需要有一个标准。既然是  $A_i$  的 2 次方，那么联想一下两个人分别取珠子，第一个人随意取珠子，第二个人必须以第一个人作为标准，保证自己的输出序列和第一个人相

同，这样，对于每种输出序列，两个人都可以采取任意一种可行的取法，由乘法原理可知方案正好是  $A_i^2$ 。状态自然就可以设计出来， $F[i, j, k, l]$ 表示第一个人在上管道取了  $i$  颗，下管道取了  $j$  颗，第二个人在上管道取了  $k$  颗，下管道取了  $l$  颗，并且两人输出序列相同时有多少种方案。由于  $i+j=k+l$ ，所以  $l$  那一维状态是多余的，可以之间简化成  $F[i, j, k]$ ，转移也十分明显，枚举当前的输出序列是要取  $A$  珠子还是  $B$  珠子，再进行相应的转移即可。

这个算法的时间复杂度是  $O(N^3)$  的，理论上超过了时间限制，但是由于第二个人必须保证和第一个人的输出序列相同，时间上状态的限制非常多，用到的状态非常少，所以可以采用记忆化搜索+HASH 或者宽搜+HASH 搜出所有可能出现的状态后再动态规划的方法加速，就能通过全部数据了。

## 积分计算表

赛事	最低积分	最高积分	积分计算办法
APIO	50	500	第 1 名获奖选手积分 = 最高积分
CTSC	50	800	最后一名获奖选手积分 = 最低积分
			第 i 名获奖选手积分 = 最高积分 - 分差 * (i-1)
冬令营	50	600	分差 = (最高积分 - 最低积分) / (获奖总人数 - 1)
<p><b>说明：</b></p> <ol style="list-style-type: none"> <li>1. 累计积分为上述三项比赛积分之和。</li> <li>2. 一项比赛只累计一次积分，若某项比赛参加多次，可选一次最高积分参与计算。</li> <li>3. 获奖总人数为获得金牌（一等奖）、银牌（二等奖）和铜牌（三等奖）的总人数。</li> <li>4. 十级选手不参与积分运算。</li> </ol>			

---

## 1~10 级评分表

级别	评级要求
一级	NOIP 初赛 60 分以上（此级不颁发等级证书）
二级	NOIP 初赛 80 分以上（此级不颁发等级证书）
三级	进入 NOIP 复赛
四级	NOIP 普及组（或提高组）复赛中成绩列全国前 50%
五级	NOIP 普及组复赛中成绩列全国前 20%
六级	NOIP 提高组复赛中成绩列全国前 20%
七级	NOIP 提高组复赛中成绩列全国前 10%
八级	NOI 铜牌或积分 250 分以上
九级	NOI 银牌或积分 500 分以上
十级	NOI 金牌或积分 1000 分以上



**中国计算机学会**  
China Computer Federation

地址：北京 海淀区科学院南路6号  
通信：北京 2704 信箱，100190  
电话：(86-10) 6256 2503

传真：(86-10) 6256 7485  
邮箱：ccf@ccf.org.cn  
网址：www.ccf.org.cn