IOI 2017 中国代表队选拔赛
暨全国青少年信息学奥林匹克精英赛

CCF CTSC 2017

第一试

时间：2017 年 5 月 8 日 08:30 ~ 13:30

题目名称	密钥	网络	游戏
题目类型	传统型	传统型	传统型
目录	cipher	tree	game
可执行文件名	cipher	tree	game
输入文件名	cipher.in	tree.in	game.in
输出文件名	cipher.out	tree.out	game.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	10	20	20
每个测试点分值	10	5	5

提交源程序文件名

对于 C++ 语言	cipher.cpp	tree.cpp	game.cpp
对于 C 语言	cipher.c	tree.c	game.c
对于 Pascal 语言	cipher.pas	tree.pas	game.pas

编译选项

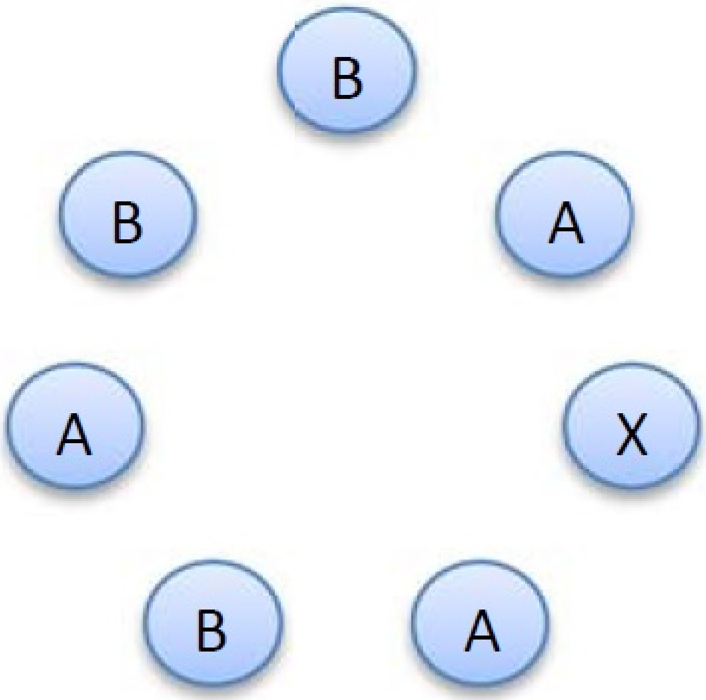
对于 C++ 语言	-lm	-lm	-O2 -lm
对于 C 语言	-lm	-lm	-O2 -lm
对于 Pascal 语言			-O2

密钥（cipher）

【问题描述】

一个密钥是一个长度为 $n = 2k + 1$ 的字符串，它包含 1 个字母 X、k 个字母 A 和 k 个字母 B。例如 $k = 3$ 时，BAXABAB 就是一个密钥。

如下图所示，可以按顺时针顺序把这 2k+1 个字母排成一个圈：



在 k 个字母 A 中，有一部分可以定义为"**强的**"。具体来说，从 X 出发顺时针走到某个 A 时，如果途中 A 的数目**严格多于** B 的数目，则称此字母 A 为强的。

对于上面的例子来说，顺时针方向从字母 X 数起第 1 个和第 2 个字母 A 是强的，而第 3 个字母 A 不是强的。

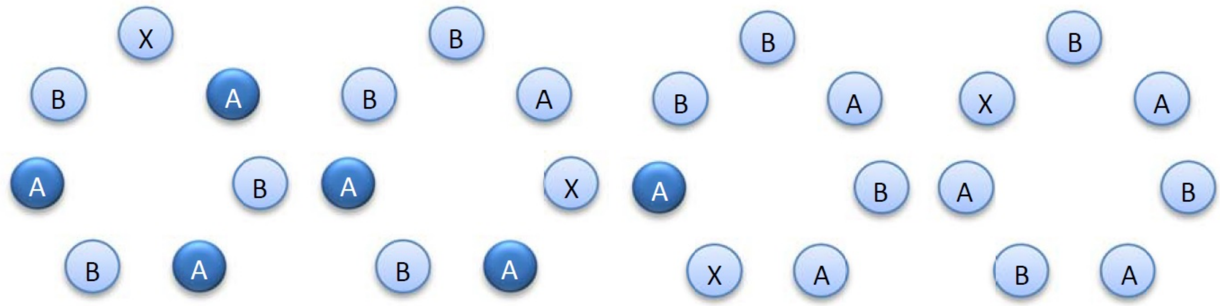
一个密钥的**特征值**就是其中包含的强的字母 A 的个数。

天才小朋友 KT 给出了一个结论：

假设 k 个字母 A 所在的位置已经固定，但是剩下的 k 个 B 和 1 个 X 的位置是未知的。（注意，满足这样要求的密钥一共有 $k + 1$ 个，因为字母 X 还剩下 $k + 1$ 个可能的位置。）

可以证明：所有这 $k + 1$ 个可能的密钥的特征值是各不相同的，它们恰好为 $0, 1, 2, \ldots, k$。

下页的图是一个具体的示例，从左到右的四个子图中分别有 3 个，2 个，1 个，0 个字母 A 是强的。



类似地，如果固定 k 个字母 B 的位置，那满足条件的所有 $k+1$ 个密钥的特征值也各不相同，恰好为 $0,1,\ldots,k$。

现在你需要解决以下三个问题：

1. 给定密钥中所有 A 的位置，当密钥的特征值为 0 时，请问 X 在哪个位置。
2. 给定密钥中所有 A 的位置，当密钥的特征值为 S 时，请问 X 在哪个位置。
3. 给定密钥中所有 B 的位置，当密钥的特征值为 S 时，请问 X 在哪个位置。

注意：字符串的 $2k+1$ 个字母的位置由 1 到 $2k+1$ 编号。

【例子 1】

假定 $k=3, S=2$。那么：

当 A 的位置是 {2,4,6} 且特征值为 0 时，X 的位置在 7；

当 A 的位置是 {2,4,6} 且特征值为 2 时，X 的位置在 3；

当 B 的位置是 {2,4,6} 且特征值为 2 时，X 的位置在 5。

【例子 2】

假定 k=9。S=7。那么：

当 A 的位置是 {3,4,5,9,10,12,13,16,19} 且特征值为 0 时，X 的位置在 14；

当 A 的位置是 {3,4,5,9,10,12,13,16,19} 且特征值为 7 时，X 的位置在 18；

当 B 的位置是 {3,4,5,9,10,12,13,16,19} 且特征值为 7 时，X 的位置在 17。

【输入格式】

从文件 ***cipher.in*** 中读入数据。

只包含一组测试数据。

第一行包含一个整数 k，意义如题所述。

第二行包含一个整数 $seed$，这个数将用于生成一个 k 元集合 P。

第三行包含一个整数 S，意义如题所述。

保证 $0 \le S \le k \le 10^7$。$1 \le seed \le 10000$。

在 `cipher/`下，包含三个用于生成输入数据的文件 `cipher.cpp/c/pas`。其中读入部分已经完成，在数组 $p[]$ 中，若 $p[i] = 0$，表示 i 不属于集合 P，否则，i 属于集合 P。

【输出格式】

输出到文件 ***cipher.out*** 中。

输出三行，每行一个数，依次对应问题描述中的三个子问题的答案。

即：

1. 第一个数表示当 k 元集合 P 代表 A 的位置且特征值为 0 时 X 的位置。
2. 第二个数表示当 k 元集合 P 代表 A 的位置且特征值为 S 时 X 的位置。
3. 第三个数表示当 k 元集合 P 代表 B 的位置且特征值为 S 时 X 的位置。

【样例 1 输入】

```
5
3344
2
```

【样例 1 输出】

```
10
1
2
```

【样例 2 输入】

```
500000
4545
234567
```

【样例 2 输出】

```
999992
246922
753067
```

【样例解释】

第一个样例中, P 数组为 1 的元素的下标分别为 $5, 6, 7, 8, 9$。

【数据范围与约定】

对于 30% 的数据，$k \le 10^3$。

对于 50% 的数据，$k \le 10^5$。

对于 100% 的数据，$k \le 10^7$。

对于每个测试点, 得分为以下三部分得分之和：

1. 如果第一问回答正确，你将获得 3 分。
2. 如果第二问回答正确，你将获得 4 分。
3. 如果第三问回答正确，你将获得 3 分。

如果你仅仅知道部分答案，请也务必按此格式要求输出三个数。否则你可能会因格式错误无法得分。

网络（tree）

【问题描述】

一个一般的网络系统可以被描述成一张无向连通图。图上的每个节点为一个服务器，连接服务器与服务器的数据线则看作图上的一条边，边权为该数据线的长度。两个服务器之间的通讯距离定义为其对应节点之间最短路的长度。

现在，考虑一个当前图结构为树的网络系统。你作为该网络系统的管理员，被要求在这个系统中新加入一条给定长度的数据线。数据线可以连在任意两个服务器上。

你的任务是，求出在所有合法的方案中，通讯距离最远的两个服务器之间的最小距离。

【输入格式】

从文件 *tree.in* 中读入数据。

输入包含多组数据。对于每组数据，输入的第一行包含二个正整数 N, L, 其中 N 表示服务器个数，L 为新加入数据线的长度。

接下来 $n-1$ 行，第 i 行有三个正整数 a_i, b_i, l_i，表示有一条长度为 l_i 的数据线连接服务器 a_i, b_i。服务器的编号为 $1 \sim N$。

输入的末尾以两个 0 作为结束。

【输出格式】

输出到文件 *tree.out* 中。

对于每组数据，输出一行一个整数，描述在所有合法的方案中，通讯距离最远的两个服务器之间的最小距离。

【样例 1 输入】

```
7 1
1 2 1
2 3 1
3 4 1
4 5 1
5 6 1
6 7 1
0 0
```

【样例 1 输出】

```
3
```

【样例 2 输入】

```
6 26
1 2 66
2 3 11
3 4 73
2 5 77
3 6 33
10 47
1 2 86
2 3 69
3 4 41
4 5 26
5 6 41
2 7 73
3 8 77
4 9 2
5 10 65
0 0
```

【样例 2 输出】

```
143
232
```

【样例 3】

见选手目录下的 *tree/tree3.in* 与 *tree/tree3.ans*。

【数据范围与约定】

一共有 20 个测试点。编号为 1 ~ 20。每个测试点为 5 分。

保证在任一测试点中，数据的组数不会超过 15，且所有数据的 N 之和不超过数据范围中标明的 N 的最大值的 5 倍。

保证所有的输入数据均为不超过 $2^{31}-1$ 的非负整数，保证 $N \geq 1$。

保证数据合法。

对于给定的测试点，其限制条件如下表所示。

<table>
<tr><th>测试点</th><th>N</th><th>测试点</th><th>N</th></tr>
<tr><td>1</td><td>≤ 10</td><td>11</td><td rowspan="5">≤ 20000</td></tr>
<tr><td>2</td><td>≤ 50</td><td>12</td></tr>
<tr><td>3</td><td rowspan="2">≤ 100</td><td>13</td></tr>
<tr><td>4</td><td>14</td></tr>
<tr><td>5</td><td>≤ 150</td><td>15</td></tr>
<tr><td>6</td><td rowspan="3">≤ 600</td><td>16</td><td rowspan="5">≤ 100000</td></tr>
<tr><td>7</td><td>17</td></tr>
<tr><td>8</td><td>18</td></tr>
<tr><td>9</td><td rowspan="2">≤ 2000</td><td>19</td></tr>
<tr><td>10</td><td>20</td></tr>
</table>

游戏（game）

【问题描述】

小 R 和室友小 B 在寝室里玩游戏。他们一共玩了 n 局游戏，每局游戏的结果要么是小 R 获胜，要么是小 B 获胜。

第 1 局游戏小 R 获胜的概率是 p_1，小 B 获胜的概率是 $1-p_1$。除了第一局游戏之外，每一局游戏小 R 获胜的概率与上一局游戏小 R 是否获胜有关。

具体来说：

1. 如果第 $i-1$ $(1 < i \le n)$ 局游戏小 R 获胜，那么第 i 局游戏小 R 获胜的概率为 p_i，小 B 获胜的概率为 $1-p_i$。
2. 如果第 $i-1$ $(1 < i \le n)$ 局游戏小 B 获胜，那么第 i 局游戏小 R 获胜的概率为 q_i，小 B 获胜的概率为 $1-q_i$。

小 D 时常过来看小 R 和小 B 玩游戏，因此他知道某几局游戏的结果。他想知道在他已知信息的条件下，小 R 在 n 局游戏中总共获胜的局数的期望是多少。

小 D 记性不太好，有时他会回忆起某局游戏的结果，并把它加入到已知信息中；有时他会忘记之前某局游戏结果，并把它从已知信息中删除。你的任务是：每当小 D 在已知信息中增加或删除一条信息时，根据小 D 记得的已知信息，帮助小 D 计算小 R 在 n 局游戏中总共获胜局数的期望是多少。

需要注意的是：如果小 D 忘了一局游戏的结果，之后又重新记起，两次记忆中的游戏结果不一定是相同的。你不需要关心小 D 的记忆是否与实际情况相符，你只需要根据他的记忆计算相应的答案。

【输入格式】

从文件 *game.in* 中读入数据。

第一行两个正整数 n, m 和一个字符串 $type$。表示小 R 和小 B 一共玩了 n 局游戏，小 D 一共进行了 m 次修改已知信息的操作，该数据的类型为 $type$。$type$ 字符串是为了能让大家更方便地获得部分分，你可能不需要用到这个输入，其具体含义见【限制与约定】。

接下来 n 行，第 1 行包含一个实数 p_1，表示第一局比赛小 R 获胜的概率是 p_1。第 i $(1 < i \le n)$ 行包含两个实数 p_i, q_i。表示在第 $i-1$ 局游戏小 R 获胜的情况下，第 i 局游戏小 R 获胜的概率是 p_i；q_i 表示在第 $i-1$ 局游戏小 B 获胜的情况下，第 i 局游戏小 R 获胜的概率是 q_i。

接下来 m 行，每行描述一个小 D 已知信息的变化，操作分为两类。

1. `add i c` 表示小 D 回忆起了第 i 局比赛的结果，并把它加入到已知信息中。若 $c = 0$ 表示第 i 局比赛小 B 获胜，若 $c = 1$ 表示第 i 局比赛小 R 获胜。数据保证

i, c 均为整数且 $1 \le i \le n, 0 \le c \le 1$，如果这个操作不是第一个操作，保证在上一个操作结束后的已知信息中没有第 i 局比赛的结果。

2. `del i` 表示小 D 忘记了第 i 局比赛的结果，并把它从已知信息中删除。数据保证 i 是整数且 $1 \le i \le n$，保证在上一个操作结束后的已知信息中有第 i 局比赛的结果。

【输出格式】

输出到文件 ***game.out*** 中。

对于每个操作，输出一行实数，表示操作结束后，在当前已知信息的条件下，小 R 在 n 局游戏中总共获胜的局数的期望是多少。

【样例 1 输入】

```
3 3 A
0.3
0.5 0.2
0.9 0.8
add 1 1
add 3 0
del 1
```

【样例 1 输出】

```
2.350000
1.333333
0.432749
```

【样例 2】

见选手目录下的 ***game/game2.in*** 与 ***game/game2.ans***。

【样例 3】

见选手目录下的 ***game/game3.in*** 与 ***game/game3.ans***。

【评分标准】

如果你的答案与正确答案的绝对误差在 10^{-4} 以内，则被判定为正确。

如果你的所有答案均为正确，则得满分，否则得 0 分。

请注意输出格式：每行输出一个答案，答案只能为一个实数。每行的长度不得超过 50。错误输出格式会被判定为 0 分。

【限制与约定】

对于 100% 的数据，$1 \le n \le 200000, 1 \le m \le 200000, 0 < p_i, q_i < 1$ 。

对于 100% 的数据，**输入保留最多四位小数**。

本题共有 20 个数据点，每个数据点 5 分, 每个测试点的具体约定如下表：

测试点	n	m	数据类型
$1-2$	≤ 10	≤ 20	A
$3-4$	≤ 100	≤ 100	B
$5-6$	≤ 1000	≤ 5000	A
$7-9$	≤ 2000	≤ 5000	B
$10-13$	≤ 10000	≤ 200000	B
$14-15$	≤ 200000	≤ 200000	C
$16-17$			D
$18-20$			A

数据类型的含义：

A：无限制

B：$\forall i > 1, |p_i - q_i| > 0.999$

C：同一时刻，小 D 最多只有 1 条已知信息

D：同一时刻，小 D 最多只有 5 条已知信息

【小 R 教你学数学】

你**可能**会用到以下公式

1. 条件概率的计算方法

我们记 $p(A|B)$ 表示在已知事件 B 发生时事件 A 发生的概率，条件概率可以用以下公式计算：

$$p(A|B) = \frac{p(AB)}{p(B)}$$

其中 $p(AB)$ 表示事件 B 和事件 A 同时发生的概率，$p(B)$ 表示事件 B 发生的概率。

2. 贝叶斯公式 (bayes)

由条件概率的计算方法，我们容易得到贝叶斯公式

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

3. 全概率公式

如果随机变量 x 有 k 个取值，分别为 $x_1, x_2, \ldots, x_k$ 那么

$$p(A) = \sum_{i=1}^{k} p(A|x = x_i)p(x = x_i)$$

【样例 1 解释】

运用贝叶斯公式

第一问：

$p(x_2 = 1|x_1 = 1) = 0.5, p(x_3 = 1|x_1 = 1) = 0.5*0.9+0.5*0.8 = 0.85, E(x_1+x_2+x_3|x_1 = 1) = 0.5 + 0.85 + 1 = 2.35$

第二问：

$p(x_2 = 1|x_1 = 1, x_3 = 0) = \frac{p(x_3=0|x_1=1,x_2=1)p(x_2=1|x_3=0)}{p(x_3=0|x_1=1)} \approx 0.333, E(x_1 + x_2 + x_3|x_1 = 1, x_3 = 0) \approx 1.333$

第三问：

$p(x_2 = 1|x_3 = 0) = \frac{p(x_3=0|x_2=1)p(x_2=1)}{p(x_3=0)}$

其中 $p(x_3 = 0|x_2 = 1) = 0.1, p(x_2 = 1) = 0.3 * 0.5 + 0.7 * 0.2 = 0.29, p(x_3 = 0) = 0.29 * 0.1 + 0.71 * 0.2 = 0.171$

所以 $p(x_2 = 1|x_3 = 0) = 0.1 * 0.29/0.171 \approx 0.16959$

$p(x_1 = 1|x_3 = 0) = \frac{p(x_3=0|x_1=1)p(x_1=1)}{p(x_3=0)}$

其中 $p(x_3 = 0|x_1 = 1) = 0.5*0.1+0.5*0.2 = 0.15, p(x_1 = 1) = 0.3, p(x_3 = 0) = 0.171$

所以 $p(x_1 = 1|x_3 = 0) = 0.15 * 0.3/0.171 \approx 0.26316$

$E(x_1 + x_2 + x_3|x_3 = 0) \approx 0.43275$

【温馨提示】

在本题中，如果你**希望获得全部的分数**，你**可能**需要考虑由于浮点数运算引入的误差。只使用加法和乘法运算不会引入太大的误差，但请谨慎使用减法和除法。

1. 两个大小相近的数相减可以引入非常大的相对误差。
2. 如果一个矩阵的行列式值非常小，那么求解该矩阵的逆可以带来相当大的误差。

当然，如果你的算法在数学上是正确的，但没有考虑浮点数运算的误差问题，可能仍然可以获得一部分的分数。