

NOIP 2019 模拟赛 Contest 17

diamond_duke

题目名称	红黑树	旅行	数字
可执行文件名	<code>rbt</code>	<code>travel</code>	<code>number</code>
输入文件名	标准输入	标准输入	标准输入
输出文件名	标准输出	标准输出	标准输出
时间限制	1s	1s	1s
内存限制	512MB	512MB	512MB
子任务个数	4	5	8
题目类型	传统型	传统型	传统型

请注意： 评测时开启 O2 优化和 C++11 编译选项，栈空间限制同空间限制。

1 红黑树

1.1 Problem Statement

小 D 正在研究红黑树。

我们称一棵排序二叉树（本题中你可以认为它是一棵普通的有根二叉树，即每个节点的孩子个数都不超过 2 的树）是**红黑树**，当且仅当我们可以给它的每个节点染上红、黑两种颜色之一，且以下两个条件满足：

- 不存在一条边连接了两个红点；
- 如果我们将所有度数不足 2 的点均用空节点补全孩子，则从根节点出发，走到到任意空节点的路径经过的黑点个数都相同。这里，我们认为空节点没有颜色。

小 D 画出了一个 n 个点的二叉排序树，节点被依次编号为了 $1, 2, \dots, n$ 。

小 D 想要知道，这棵二叉树是不是红黑树。如果是，他还想知道一种染色方案。但是他并不会，请你帮帮他。

1.2 Input Format

从标准输入读入数据。

第一行一个整数 n ，表示树的节点个数。

第二行 n 个数 p_1, p_2, \dots, p_n ，表示第 i 个节点的父亲节点编号为 p_i 。特别地，对于 i 是根节点的情况，有 $p_i = 0$ 。

1.3 Output Format

向标准输出输出答案。

如果这棵树不是红黑树，则输出一行一个字符串 `Impossible`。否则，输出一行一个长度为 n 的仅由 R 和 B 组成的字符串表示染色方案：第 i 个字符是 R 表示节点 i 是红色的，否则是黑色的。

1.4 Sample 1

1.4.1 Input

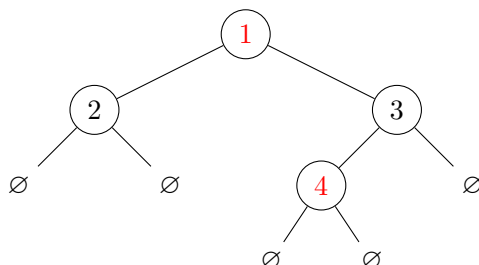
```
4
0 1 1 3
```

1.4.2 Output

```
RBBR
```

1.4.3 Explanation

一种染色方案如下，容易验证其符合红黑树的要求：



1.5 Sample 2

见下发文件 `rbt/rbt2.in` 与 `rbt/rbt2.ans`。

1.6 Sample 3

见下发文件 `rbt/rbt3.in` 与 `rbt/rbt3.ans`。

1.7 Constraints

对于所有测试数据， $2 \leq n \leq 5000$ ， $0 \leq p_i \leq n$ ，保证输入是一棵有根二叉树，即有且仅有一个根节点，每个节点有不超过 2 个孩子，且所有点均连通。

- 子任务 1 (20 分)： $n \leq 20$ ；
- 子任务 2 (30 分)： $n \leq 50$ ；
- 子任务 3 (30 分)： $n \leq 500$ ；
- 子任务 4 (20 分)： 无特殊限制。

2 旅行

2.1 Problem Statement

小 D 正在规划一次旅行。

小 D 共有 n 种方法坐飞机到达目的地，每种方法都可能需要一定次数的中转。

因为天气因素，可能某些航班会停运。对于第 i 种方法的第 j 条航班，它停运的概率是 $p_{i,j}$ 。为了减小难度，这里我们认为这些方法中经过的所有航班互不相同，且航班是否停运是相互独立的。

因为天气无法太早地预知，所以小 D 只能在当天查询每条航班是否停运。

显然，如果一种方法中任何一条航班停运了，那么小 D 就无法通过这种方法到达目的地了。小 D 希望能够找到一种可以到达目的地的方法，或者遗憾地发现所有方法都无法到达目的地。

由于时间紧张，小 D 希望查询的次数最少。小 D 想要确定一个策略，以最小化这一最小查询次数的期望。如果你不知道期望是什么，你可以把它理解为每种情况下最小查询次数与这种情况发生的概率的乘积之和。

值得一提的是，小 D 可以根据之前的查询结果决定之后如何进行查询。

但是他并不会，请你帮帮他。你只需要输出这个期望即可。

2.2 Input Format

从标准输入读入数据。

第一行一个整数 n ，表示到达目的地的方法数。

接下来 n 行，每行若干个数。第一个整数 k_i 表示第 i 种方法经过的航班个数，接下来 k_i 个小数点后不超过三位的浮点数 $p_{i,j}$ ，表示其中第 j 个航班停运的概率。

2.3 Output Format

向标准输出输出答案。

输出一行一个浮点数表示期望查询次数。你的答案被认为正确，当且仅当它与答案的相对误差或绝对误差不超过 10^{-9} 。形式化地，若你的答案为 a 而参考答案为 b ，则你的答案被认为是正确的当且仅当 $\min\{|a - b|, |a - b|/b\} \leq 10^{-9}$ 。

2.4 Sample 1

2.4.1 Input

```
2
3 0.1 0.1 0.1
```

2 0.9 0.9

2.4.2 Output

3.0081

2.4.3 Explanation

显然，第一种方法可行的概率远比第二种来得大，因此小 D 应当先检查第一种方法中的所有航班，直到遇到一个停运的航班后再去检查第二种方法的航班。

2.5 Sample 2

见下发文件 `travel/travel2.in` 与 `travel/travel2.ans`。

2.6 Sample 3

见下发文件 `travel/travel3.in` 与 `travel/travel3.ans`。

2.7 Constraints

对于所有测试数据， $1 \leq n, k_i \leq 500$ ， $0 \leq p_{i,j} \leq 1$ ，输入的浮点数不超过小数点后三位。

- 子任务 1 (20 分): $n \leq 4$, $k_i \leq 5$;
- 子任务 2 (20 分): $n, k_i \leq 10$;
- 子任务 3 (20 分): $n, k_i \leq 70$;
- 子任务 4 (20 分): $n, k_i \leq 200$;
- 子任务 5 (20 分): 无特殊限制。

3 数字

3.1 Problem Statement

小 D 和小 Y 正在玩数字游戏。

在游戏开始前，小 D 和小 Y 会约定一个数字 n ，以及 m 个介于 $[0, 2^n)$ 之间的数字 a_1, a_2, \dots, a_m 作为游戏的参数。

在游戏开始时，小 D 会选择一个数字 $x \in [0, 2^n)$ ，称为**初值**。接下来，数字 x 会依次通过这 m 个数：在通过 a_i 后， x 会变为 $x \oplus a_i$ ，其中 \oplus 为按位异或运算。

而在任意一步操作后（在小 D 选择了数字后，以及与某个数字异或后），小 Y 可以使得 x 变为 $(\lfloor \frac{x}{2^{n-1}} \rfloor + 2x) \bmod 2^n$ 。小 Y 应当一共进行该操作**恰好一次**。

小 D 的目标是最大化游戏结束后 x 的值（称为**终值**），而小 Y 想要最小化它。

小 D 和小 Y 想要知道，如果他们都以最优策略进行游戏，那么游戏结束后终值应该是多少。

在某些时候，小 D 还想要知道，有多少个初值，可以使游戏达到这个终值。

但是他们并不会，请你帮帮他们。

3.2 Input Format

从标准输入读入数据。

第一行三个空格隔开的整数 n, m, T ， n, m 的含义请参考题目描述部分，而 T 表示是否需要输出初值的个数。

第二行 m 个空格隔开的整数 a_1, a_2, \dots, a_m ，含义请参考题目描述部分。

3.3 Output Format

向标准输出输出答案。

第一行输出一行一个整数，表示 x 的终值。

若 $T = 1$ ，则你还需要输出一行一个整数，表示可以取到这个终值的初值个数。如果 $T = 0$ ，则你不需要输出第二行。

3.4 Sample 1

3.4.1 Input

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

3.4.2 Output

1
2

3.4.3 Explanation

下表为小 Y 采取最优策略时，初值与终值的对应关系：

初值	0	1	2	3
终值	0	1	1	0

因此小 D 可以选择初值 1 或 2 来达到最大的终值 1。

3.5 Sample 2

见下发文件 `number/number2.in` 与 `number/number2.ans`。

3.6 Sample 3

见下发文件 `number/number3.in` 与 `number/number3.ans`。

3.7 Constraints

对于所有测试数据， $1 \leq n \leq 30$ ， $1 \leq m \leq 10^6$ ， $0 \leq a_i < 2^n$ ， $T \in \{0, 1\}$ 。

- 子任务 1, 2 (18, 12 分)： $n \leq 10$ ， $m \leq 1000$ ；
- 子任务 3, 4 (12, 8 分)： $m \leq 10$ ；
- 子任务 5, 6 (18, 12 分)： $m \leq 10^5$ ；
- 子任务 7, 8 (12, 8 分)：无特殊限制。

对于编号为奇数的子任务，有 $T = 0$ 成立。