

LYK loves QAQ(qaq)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK 有一个骰子，但是不幸的是他并不知道骰子的英文名字，于是他打算用“QAQ”来作为题目名字。这颗骰子共有 n 面，第 i 面骰子对应的数字是 i 。LYK 会抛下无穷多次骰子，每次骰到的数字将会累加进和里。

现在 LYK 想知道出现过的和中存在数字 m 的概率。

输入格式(qaq.in)

第一行一个数 T ，表示有 T 组数据。

接下来 T 行，每行两个数表示 n 和 m 。

输出格式(qaq.out)

T 行，每行一个数表示概率，保留 5 位小数。

输入样例

```
2
3 5
6 10
```

输出样例

```
0.49794
0.28929
```

数据范围

对于 20% 的数据 $n, m \leq 10$ 。

对于 50% 的数据 $n, m \leq 100000$ 。

对于另外 30% 的数据 $n \leq 10, m \leq 10^9$ 。

对于 100% 的数据 $n, m \leq 10^9, T \leq 10$ 。

LYK loves string(string)

Time Limit:1000ms Memory Limit:128MB

题目描述

LYK 喜欢字符串, 它认为, 一个字符串 S 可以用 $[S']k$ 来表示, 其中 k 表示 S' 出现的次数, S' 为 S 的某个前缀。例如字符串 **ababab** 可以用 $[ab]3$ 来表示, 当然也可以用 $[ababab]1$ 来表示。它认为一个字符串的价值就是能找到的 k 的最大值, 那么 **ababab** 的价值即为 3。

它现在拥有一个字符串 S , 它想找到 S 的一个子串, 使得该子串的价值最大。当有许多这样的子串时, 它想知道字典序最小的子串。

输入格式(string.in)

第一行一个数 T , 表示有 T 组数据。

对于每组数据读入一个仅包含小写字母的字符串 S 。

输出格式(string.out)

对于每组数据, 输出一行表示答案。

输入样例

```
2
cabababcaa
cabababcaaaa
```

输出样例

```
ababab
aaa
```

数据范围

令 sum 表示所有数据读入的字符串的长度之和, $|S_i|$ 表示第 i 组数据的字符串 S 的长度。

对于 20% 的数据: $sum \leq 50$ 。

对于 40% 的数据: $sum \leq 500$ 。

对于 60% 的数据: $sum \leq 2000$ 。

对于 80% 的数据: $sum \leq 50000$ 。

对于 100% 的数据: $T \leq 10$, $sum \leq 200000$, $|S_i| \leq 100000$ 。

LYK loves tree(tree)

Time Limit:1000ms Memory Limit:512MB

题目描述

LYK 找到了一棵红黑树，所谓红黑树，就是所有节点要么是黑色的要么是红色的。它觉得这棵树不怎么好看，于是它想通过交换两个节点的颜色使得这棵树好看起来。

LYK 认为，一棵树是好看的当且仅当对于所有点来说，存在一个黑点与它的距离不超过 x 。它想通过最少的交换次数使得这棵树变得好看。

输入格式(tree.in)

第一行两个数 n, x 表示树的节点个数与参数 x 。

接下来一行 n 个数 a_i 表示第 i 个节点的颜色(0 表示红色，1 表示黑色)。

接下来 $n-1$ 行，每行两个数 u,v,w 表示存在长为 w 的边连接 u,v 这两个点。

输出格式(tree.out)

输出一行表示最少操作次数，如果无解输出-1。

输入样例

3 2

1 0 0

1 2 2

2 3 2

输出样例

1

数据范围

对于 20%的数据 $n \leq 10$ 。

对于 40%的数据 $n \leq 100$ 。

对于另外 20%的数据 $w=1, x=1$ 。

对于再另外 20%的数据 $w=1, x=2$ 。

对于 100%的数据 $n \leq 500, w, x \leq 10^9$ 。

其中均匀分布着一半的数据树是随机的。

其中随机分布着 10%的数据无解。