

膜你赛

yxh-12243

December 20, 2020

题目名称	冠状病毒	传播者	医院
题目类型	传统型	传统型	传统型
目录	rna	quarantine	tree
可执行文件名	rna	quarantine	tree
输入文件名	rna.in	quarantine.in	tree.in
输出文件名	rna.out	quarantine.out	tree.out
每个测试点时限	1 秒	2.5 秒	1 秒
内存限制	512 MiB	1024 MiB	512 MiB
子任务数目	28	25	30
测试点是否等分	否	否	否

提交源程序文件名

对于 C++ 语言	rna.cpp	quarantine.cpp	tree.cpp
对于 C 语言	rna.c	quarantine.c	tree.c
对于 Pascal 语言	rna.pas	quarantine.pas	tree.pas

编译选项

对于 C++ 语言	-O2 -lm -std=c++11
对于 C 语言	-O2
对于 Pascal 语言	-O2

注意事项

1. 需要建立子文件夹。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. 结果比较方式为忽略行末空格、文末回车后的全文比较。
4. C/C++ 中函数 main() 的返回值类型必须是 int，值为 0。

冠状病毒（rna）

【题目背景】

本题的故事发生在 S 星球，在这里我们将为你介绍一些必要的设定。

众所周知，地球上生物体中常见的含 N 碱基有 5 种：腺嘌呤 (用 A 表示)、鸟嘌呤 (用 G 表示)、胞嘧啶 (用 C 表示)、胸腺嘧啶 (用 T 表示) 以及尿嘧啶 (用 U 表示)。

其中，在 DNA/RNA 复制以及转录等相关过程中，它们会遵循**碱基互补配对原则**：其中 A 和 T (U) 配对， G 和 C 配对。

不过，在 S 星球上，一共有 $2n$ 种含 N 碱基，分别用 $b_0, b_1, b_2, \dots, b_{2n-2}, b_{2n-1}$ 表示，在 S 星球上，也有对应的**碱基互补配对原则**：碱基 b_0 和 b_1 配对，碱基 b_2 和 b_3 配对， \dots ，碱基 b_{2n-2} 和 b_{2n-1} 配对。

特别地，对于地球上的生物体，可以看作是 $n = 2$ 的特殊情形：如，规定 $b_0 = A, b_1 = T(U), b_2 = G, b_3 = C$ ，下列**关于地球上碱基序列的描述**均按照此规则。

在下面，如果不加特殊说明，我们用一个 $0 \sim 2n - 1$ 的数字 i ($0 \leq i \leq 2n - 1$) 来表示碱基 b_i 。

【题目描述】

公元 8224 年， S 星球的 Θ 国爆发了一种前所未有的冠状病毒，该病毒具有潜伏期长、致病率低的特性。

8224 年 2 月 29 日，该病毒已造成 12243 人死亡，998244353 人确诊。小 ω 作为 Θ 国的总统，觉得事情已刻不容缓。

经过 Θ 国珂学院的不懈努力，已经将该冠状病毒的基因组成功分离出来，并将其命名为 SARS-CoV-233。

已知 **SARS-CoV-233 病毒第一代** 的碱基序列 (核苷酸序列) 的长度为 L ，且每个碱基为 $b_0, b_1, \dots, b_{2n-1}$ 之一。

Θ 国珂学院还得出，目前有一种细菌 ν 可以产生一种能对 SARS-CoV-233 的繁殖有着强烈的抑制作用的抗体。所以，小 ω 希望尽可能利用细菌 ν 来制造解药。

细菌 ν 是一种非常特殊的生物，它**每繁殖一代，后代的 RNA 序列恰好为在母体的 RNA 序列末尾增加一个碱基 b_{new} 组成**，也就是说，后代的 RNA 序列的长度恰好比母体多 1，且母体的 RNA 序列是后代的 RNA 序列的一个前缀。

目前 Θ 国的科技还无法达到能控制 b_{new} 的能力，因此该种细菌在繁殖时， b_{new} 可以看成在 $b_0, b_1, \dots, b_{2n-1}$ 中**等概率随机**。

如果某一代细菌 ν 的 RNA 序列中，**存在某一个连续的段**，与某一代 SARS-CoV-233 病毒完整的碱基序列遵循**碱基互补配对原则**，则它会产生**这一代**的 SARS-CoV-233 病毒的抗体，从而对该代 SARS-CoV-233 病毒产生抑制作用，减弱它对 Θ 国人类的影响。

但是，SARS-CoV-233 病毒并不是一成不变的，它也会随着外界的环境而进行进化。具体地，每一次进化也是从**原来的某一代病毒**的碱基序列末尾**增加一个碱基 b_{new}** 得到。

小 ω 想知道用细菌 ν 来制造解药的方案是否现实, 于是想请你计算一些问题:

从 2 月 29 日起, Θ 国珂学院已基本预测出将来 q 天的病毒进化情况。

截止 2 月 29 日, SARS-CoV-233 只有一代 (我们下面称它为第 0 代), 它的碱基序列的长度为 L , 序列为 r_0, r_1, \dots, r_{L-1} 。

在接下来的 q 天, 每天会发生如下两种事情之一:

1. SARS-CoV-233 病毒的某一代发生了进化。具体地, SARS-CoV-233 病毒第 f 代的碱基序列末尾增加了一个碱基 b_r , 作为 SARS-CoV-233 病毒第 tot 代, 其中 tot 为当前病毒的总代数 (包含第 0 代, 不包含新产生的一代)。
2. SARS-CoV-233 没有进行进化, Θ 国珂学院对其中的第 f 代病毒进行了研究, 想知道细菌 ν 对该代病毒的抑制情况。

具体地, 目前她们已经得到一个细菌 ν 的样本, 这个样本的 RNA 序列的长度为 l , 且恰好与该代病毒的前 l 个碱基满足碱基互补配对原则。

她们想知道, 期望再繁殖多少代后, 才能产生对该代病毒的抗体。

注意: 某一代病毒可能被研究多次, 也可能一次都未被研究。

细菌 ν 的繁殖遵循链式原则, 即第 k 代是由第 $k-1$ 代的 RNA 序列末尾增加一个碱基, 而 SARS-CoV-233 病毒的繁殖是树状的, 一次进化是由前面某一代病毒末尾补充一个碱基而得。

【输入格式】

从文件 `rna.in` 中读入数据。

第一行包含一个正整数 id , 表示子任务编号。

第二行包含四个非负整数 n, L, q, enc , 分别表示碱基对的种数, SARS-CoV-233 病毒第 0 代的碱基序列长度, 需要处理的天数, 以及数据是否加密。

第三行包含 L 个非负整数 $r_0, r_1, r_2, \dots, r_{L-1}$, 描述 SARS-CoV-233 病毒第 0 代的碱基序列 (核苷酸序列)。

接下来 q 行, 每行描述一天的事件, 格式如下:

1. $\boxed{A f r}$ 表示这一天 SARS-CoV-233 病毒发生了进化, 即第 f 代的碱基序列末尾增加了碱基 b_r , 作为病毒的最新一代。
2. $\boxed{Q f l}$ 表示 Θ 国珂学院在研究 SARS-CoV-233 病毒第 f 代, 且已知某个细菌 ν 的样本的 RNA 序列长度为 l 且与该代病毒的前 l 个碱基遵循碱基互补配对原则, 求期望再繁殖多少代之后能产生对该代病毒的抗体。

注意 r, l (没有 $f!$) 是经过加密的, 记输入的值为 r_0, l_0 , 且上次输出的答案为 ans (初始时为 0), 则:

$$r = (r_0 + enc \cdot ans) \bmod (2n)$$

$$l = (l_0 + enc \cdot ans) \bmod (L_f + 1)$$

其中 L_f 表示 SARS-CoV-233 病毒第 f 代的碱基序列长度。

【输出格式】

输出到文件 *rna.out* 中。

对于每次 Q 事件，输出一行一个整数，表示期望再繁殖的代数在模 998244353 意义下的结果。可以证明，这个数值一定是一个有理数，且分母不为 998244353 的倍数。

【样例 1 输入】

```
5
1 1 5 0
0
Q 0 0
A 0 0
Q 1 0
A 0 1
Q 2 0
```

【样例 1 输出】

```
2
6
4
```

【样例 1 解释】

S 星球只有两种碱基，不妨设为 A 和 U。

SARS-CoV-233 病毒第 0 代的碱基序列为 A。

第一天，Θ 国珂学院在研究 SARS-CoV-233 病毒第 0 代，它的碱基序列为 A。

而目前，它们得到 ν 样本的 RNA 序列的长度为 0 (可以为 0)，即还没有任何碱基对。

然后细菌 ν 进行随机繁殖，每次在末端等概率添加 A 或 U。

当细菌 ν 的 RNA 序列中出现 U 时，它和第 0 代病毒的 A 满足碱基互补配对原则，从而产生第 0 代病毒的抗体。

不难证明，在一个空序列中等概率添加 A 或 U，期望 2 次后就能出现 U。

于是第一个问题的答案就是 2。

第二天，SARS-CoV-233 病毒第 0 代末尾增加了碱基 A，作为病毒第 1 代，即病毒第 1 代的碱基序列为 AA。

第三天，Θ 国珂学院在研究 SARS-CoV-233 病毒第 1 代，它的碱基序列为 AA。

而目前 ν 样本仍为空，需要进行随机繁殖，即每一次在母体后等概率添加 A 或 U。

而当 RNA 序列中出现连续的 UU 时，才能产生对第 1 代病毒的抗体。

不难证明，需要期望添加 6 次。

第四天, SARS-CoV-233 病毒第 0 代末尾增加了碱基 U, 作为病毒第 2 代, 即病毒第 2 代的碱基序列为 AU。

第五天, Θ 国珂学院在研究 SARS-CoV-233 病毒第 2 代, 它的碱基序列为 AU。

而目前 ν 样本仍为空, 需要进行随机繁殖, 即每一次在母体后等概率添加 A 或 U。

而当 RNA 序列中出现连续的 UA 时, 才能产生对第 1 代病毒的抗体。

我们把整个过程分为两步:

第一步, 随机添加, 直到出现U, 这个过程期望 2 代。

第二步, 随机添加, 直到出现A。因为在出现 A 之前, 前面的所有碱基均为 U, 故不影响第一步的成果, 这个过程期望还是 2 代。

由期望的线性性知, 总共期望 $2 + 2 = 4$ 代。

【样例 2 输入】

```
6
1 1 5 1
0
Q 0 0
A 0 0
Q 1 1
A 0 1
Q 2 0
```

【样例 2 输出】

```
2
6
4
```

【样例 2 解释】

该组样例为样例一的加密版本。

【样例 3 输入】

```
5
2 16 8 0
0 1 2 3 0 1 0 1 2 3 0 1 0 1 2 3
Q 0 0
Q 0 8
Q 0 16
```

```
A 0 0
A 1 1
Q 2 0
Q 2 9
Q 2 18
```

【样例 3 输出】

```
303038716
302973164
0
855642060
855379852
0
```

【样例 3 解释】

注意 l 可以等于 L_f ，说明此时细菌 ν 已产生 SARS-CoV-233 病毒第 f 代的抗体，故期望再繁殖 0 代。

【样例 4 输入】

```
6
2 16 8 1
0 1 2 3 0 1 0 1 2 3 0 1 0 1 2 3
Q 0 0
Q 0 11
Q 0 2
A 0 0
A 1 1
Q 2 0
Q 2 16
Q 2 14
```

【样例 4 输出】

```
303038716
302973164
0
855642060
```

855379852

0

【样例 4 解释】

该组样例为样例三的加密版本。

【样例 5】

见选手目录下的 *rna/rna5.in* 与 *rna/rna5.ans*。

该组样例保证出现的碱基序列来源于 *Severe acute respiratory syndrome-related* 病毒的 RNA 序列。

【样例 6】

见选手目录下的 *rna/rna6.in* 与 *rna/rna6.ans*。

该组样例为样例五的加密版本。

【数据范围】

对于所有的测试点, 均满足 $1 \leq n, L, q \leq 3 \times 10^5; enc \in \{0, 1\}; 1 \leq id \leq 28; 0 \leq r, r_i \leq 2n - 1; 0 \leq l \leq L_f; 0 \leq f < tot$, 其中 *tot* 为当前病毒的总代数, 初始时为 1 (包含第 0 代), 但不包含将要产生的那一代。此外, 无论是解密前还是解密后, 所有变量均满足上述限制。

具体的子任务的数据规模见下表:

子任务 (id =)	分值	n	L	q	enc	其它性质		
1	1	$\leq 3 \times 10^5$	$\leq 3 \times 10^5$	$\leq 3 \times 10^5$	= 0	保证没有 Q 事件		
2	1				≤ 1			
3	4	≤ 2	≤ 2	≤ 4	= 0	无		
4	4				≤ 1			
5	3	≤ 100	≤ 100	≤ 100	= 0			
6	3				≤ 1			
7	3	= 1	≤ 2000	≤ 2000	= 0			
8	3				≤ 1			
9	3	≤ 2000			= 0			
10	2				≤ 1			
11	5	= 2	≤ 30000	≤ 30000	= 0	保证出现的碱基序列均来源于果蝇染色体		
12	4				≤ 1	无		
13	3				= 0			
14	3				≤ 1	保证所有 Q 事件在 A 事件之后		
15	5	≤ 30000			= 0			
16	4				≤ 1	无		
17	4				= 0			
18	3				≤ 1	保证出现的碱基序列均来源于果蝇染色体或人类染色体		
19	6	= 2			= 0			
20	5				≤ 1			
21	4				= 0	无		
22	3				≤ 1			
23	4	$\leq 3 \times 10^5$	$\leq 3 \times 10^5$	$\leq 3 \times 10^5$	= 0	保证没有 A 事件		
24	4				≤ 1			
25	4				= 0	保证所有 Q 事件在 A 事件之后		
26	4				≤ 1			
27	4				= 0	无		
28	4				≤ 1			

同时，保证编号为 $2i$ 的子任务和 $2i - 1$ 的子任务的数据完全一致，除了 $enc = 1$ ，以及对应的输入加密。换句话说，子任务 $2i$ 为子任务 $2i - 1$ 的加密版本 ($1 \leq i \leq 14$)。

子任务的依赖规则满足：子任务 a 直接或间接依赖子任务 b ，当且仅当 $a \neq b$ 且一切满足子任务 b 限制的数据均满足子任务 a 的限制。

传播者 (quarantine)

【题目描述】

研究完 SARS-CoV-233 病毒的性状后, S 星球开始转而处理因 SARS-CoV-233 而得病的人。

SHO (S-planet Health Organization) 规定, 将 SARS-CoV-233 病毒感染的肺炎命名为 COVID-12243。

在 S 星球的这一时期, 有众多珂技和卫生会议需要召开。S 星球的会议召开是逐级进行的:

先由 S 星球的联合国在会议中提出若干问题及方法, 然后各国的总统将这些精神传达到各省, 这样逐级传下去, 最后落实到每个人, 再汇总起来。

但是, 在 Θ 国的各级人民代表大会召开过程中, 由于某些乡村的医疗水平不够发达, 导致有些村民已经不知不觉地患上了 COVID-12243, 然而试剂并不能检验出来。

小 ω 作为 Θ 国的总统, 觉得这件事非常严重。于是, 她将亲自下访整条路线, 带上高超的试剂, 并隔离这些患 COVID-12243 的病人。

形式化地, S 星球的行政区划分为 n 类 (相当于中国的国 - 省 - 市 - 县/区 - 乡等), 从小到大分别称为 1 级行政单位, 2 级行政单位, \dots , n 级行政单位 (Θ 国)。

现在, 我们考察一条特定的路线 (即某个乡 $\rightarrow \dots \rightarrow \Theta$ 国), 其中每一级行政单位中有 k 个人大代表, 我们用 (i, j) 表示 i 级人大代表中的第 j 个。

已知, 相邻两级的人大代表会有相互见面的机会, 而非相邻两级的人大代表 (即使是同级) 没有相互见面的机会。

(ps: 同级人大代表在开会的时候有某些特殊的措施, 导致即使相互见面也不会感染病毒)

对于相邻两级的人大代表, 小 ω 已经调查清楚了: 对于 $\forall 1 \leq i \leq n-1, 1 \leq u, v \leq k, (i, u)$ 和 $(i+1, v)$ 是否有相互见面的机会。

现在在 Θ 国需要召开若干次人民代表大会, 每次会议的要求如下:

首先, 第 l 级行政单位召开人民代表大会, 然后第 l 级人大代表和第 $l+1$ 级人大代表依次见面, 然后第 $l+1$ 级行政单位开始开会, 然后再与 $l+2$ 级人大代表依次见面, \dots , 最终到第 r 级会议开完为止, 最后, r 级人大代表需要向小 ω 汇报消息。

特别地, 我们会给定两个集合 P_l, P_r , 表示 l 级行政单位中, 只有 P_l 集合中的人参与整场会议, 在 r 级行政单位中, 只有 P_r 集合中的人参与整场会议。而对于 $\forall l < i < r, i$ 级行政单位的所有人 大代表都必须参加。

但是, 目前已知第 l 级的人大代表具有潜在的患 COVID-12243 可能性, 而其他人并没有。当两个人见面时, 病毒会从下级人大代表传播到上级人大代表。这将导致小 ω 有一定的几率感染 SARS-CoV-233。

于是, 她会选择所有人大代表中的若干个将其隔离, 尤其是一些超级传播者。被隔离的人与任何人都不能见面, 可以通过特殊的方式传递信息。

但是, 将一个人隔离的代价是很大的, 所以, 小 ω 希望隔离尽可能少的人, 从而确保她不会被感染 (即使会议不能开成功)。

而且, 由于某些原因, 不同人之间是否有相互见面的机会, 是在不断改变的, 但始终保持只有相邻两级的人才相互见面的机会。

注意: 在两次不同的会议中, 「第 l 级的人大代表具有潜在的患 COVID-12243 可能性」是相互独立的, 互不影响。

【输入格式】

从文件 `quarantine.in` 中读入数据。

第一行包含三个正整数 n, k, q , 表示行政单位的种数, 每一级人大代表的个数和事件的个数。

接下来 $k(n-1)$ 行, 分为 $n-1$ 段, 每段 k 行。

对于第 i ($1 \leq i \leq n-1$) 段, 第 u ($1 \leq u \leq k$) 行包含一个长度为 k 的 0/1 串, 其中第 v ($1 \leq v \leq k$) 个字符为 1 表示 (i, u) 和 $(i+1, v)$ 有相互见面的机会, 否则表示没有相互见面的机会。

接下来 q 行, 每行描述一个事件, 格式如下:

1. `T i u v` 表示 (i, u) 和 $(i+1, v)$ 能否相互见面关系发生改变, 即如果原先不能相互见面, 则现在能相互见面; 如果原先能相互见面, 则现在不能相互见面。
2. `M l r Pl Pr` 表示第 $l \sim r$ 级行政单位开了一次人民代表大会, 具体会议的形式见题目描述, 其中 P_l, P_r 为 0/1 串, P_l 的第 j 个字符为 1 当且仅当 (l, j) 参加这次会议。对于 r 的情况同理。你需要求出被隔离的人数的最小值。

【输出格式】

输出到文件 `quarantine.out` 中。

对于每次 Q 事件, 输出一行一个整数, 表示需要被隔离的人数的最小值。

【样例 1 输入】

```
2 5 13
11000
00100
00100
00100
00011
M 1 2 11111 11111
M 1 2 01110 11011
M 1 2 01010 01110
T 1 2 2
T 1 4 4
M 1 2 11111 11111
M 1 2 01110 11011
```

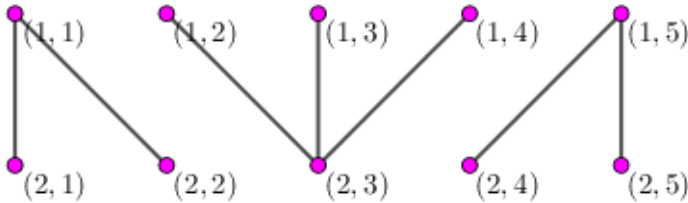
```
M 1 2 01010 01110
T 1 2 2
T 1 4 4
M 1 2 11111 11111
M 1 2 01110 11011
M 1 2 01010 01110
```

【样例 1 输出】

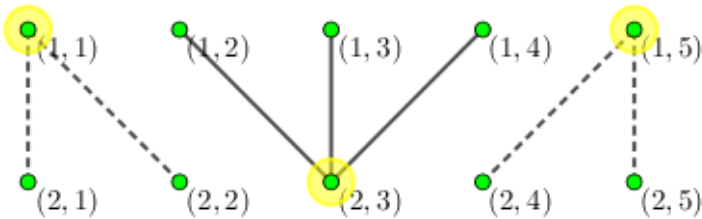
```
3
0
1
5
2
2
3
0
1
```

【样例 1 解释】

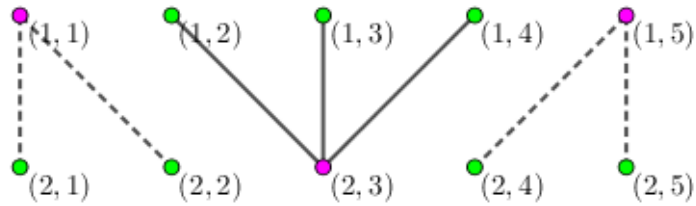
用第一行的点表 1 级人大代表，用第二行的点表示 2 级人大代表，如果两个人大代表能见面，则用一条线段相连，则所得的图形如下：



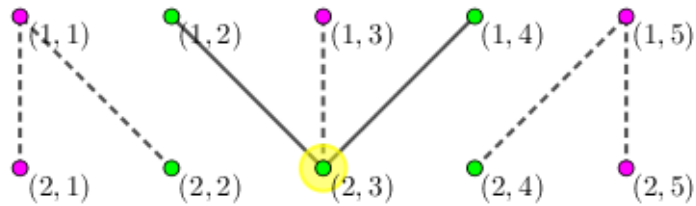
对于第 1 次人民代表大会，所有的人大代表都要参加，于是小 ω 为了防止自己被感染，需要至少隔离三个人（隔离用黄圈表示）：



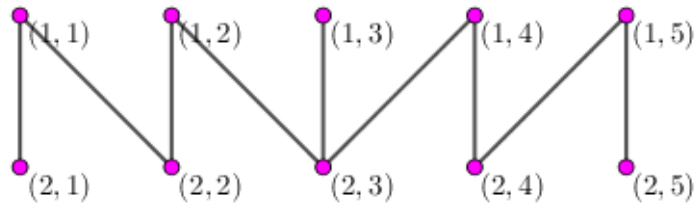
对于第 2 次人民代表大会，只有如下 7 个人大代表需要参加会议，于是这个会议本身就无法成功，小 ω 不需要隔离任何人：



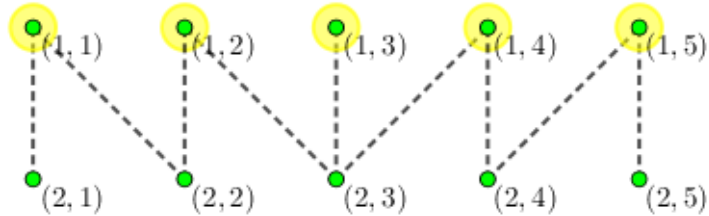
对于第 3 次人民代表大会，只有如下 5 个人大代表需要参加会议，于是为了防止感染小 ω ，只需隔离 (2,3) 即可：



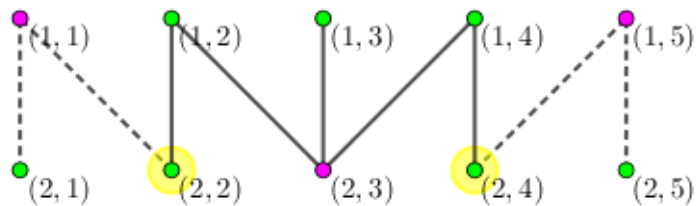
然后，(1,2) 和 (2,2) 的见面关系发生改变，(1,4) 和 (2,4) 的见面关系发生改变，即新的关系图如下：



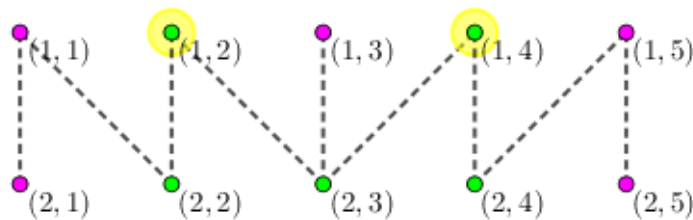
接下来对于第 4 次人民代表大会，所有人大代表都要参加，此时就需要隔离至少 5 个人了：



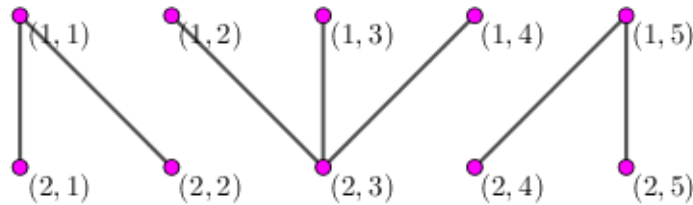
对于第 5 次人民代表大会，还是那时的 7 人参加，不过这回需要隔离 2 个人：



对于第 6 次人民代表大会，有 5 个人参加，这次也隔离 2 个人：



然后，(1,2) 和 (2,2)，(1,4) 和 (2,4) 的见面关系又发生改变，于是她们又无法见面了，从而见面关系图又恢复为最初的形态：



于是接下来的 3 次人民代表大会，所需要隔离的人数和最开始的 3 次相同，为 3,0,1。

【样例 2 输入】

```
3 2 10
01
10
01
10
M 1 3 10 10
M 1 3 10 01
M 1 3 01 10
M 1 3 01 01
M 1 3 11 11
M 1 2 10 10
M 1 2 10 01
M 1 2 01 10
M 1 2 01 01
M 1 2 11 11
```

【样例 1 输出】

```
1
0
0
```

1
2
0
1
1
0
2

【样例 2 解释】

注意中间级的所有人大代表都要参加会议。

【样例 3 输入】

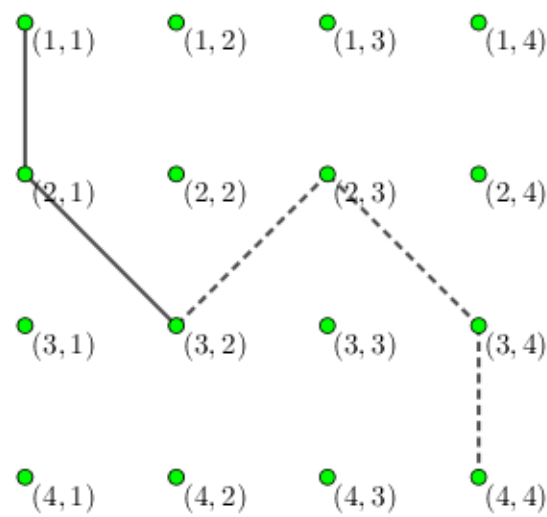
4 4 1
1000
0000
0000
0000
0100
0000
0101
0000
0000
0000
0000
0001
M 1 4 1111 1111

【样例 3 输出】

0

【样例 3 解释】

注意病毒只会从下级人大代表传播到上级人大代表，如下图：



【样例 4】

见选手目录下的 `quarantine/quarantine4.in` 与 `quarantine/quarantine4.ans`。

【数据范围】

对于所有的测试点，均满足 $2 \leq n \leq 8192; 1 \leq k \leq 24; 1 \leq q \leq 8192; 1 \leq i \leq n - 1; 1 \leq u, v \leq K; 1 \leq l < r \leq n$ 。

具体的子任务的数据规模见下表：

子任务	分值	k	n	q	其它性质
1	3	$= 1$	≤ 8192	≤ 8192	无
2	4	≤ 2			
3	4	≤ 3			
4	3	≤ 5			
5	4	≤ 7	≤ 10	≤ 10	
6	3		≤ 100	≤ 100	
7	3		≤ 1000	≤ 1000	
8	2		≤ 8192	≤ 8192	
9	6	≤ 9	≤ 100	≤ 100	
10	5		≤ 1000	≤ 1000	
11	4		≤ 8192	≤ 8192	保证对所有会议，有 $l = 1, r = n$
12	4				保证所有人大代表的见面关系不改变，即没有 T 事件
13	3				
14	6	≤ 16	≤ 100	≤ 100	无
15	5		≤ 1000	≤ 1000	
16	4		≤ 8192	≤ 8192	保证对所有会议，有 $r = l + 1$
17	4				保证对所有会议，有 $l = 1, r = n$
18	4				保证所有人大代表的见面关系不改变，即没有 T 事件
19	3				
20	6	≤ 24	≤ 100	≤ 100	无
21	5		≤ 1000	≤ 1000	
22	4		≤ 8192	≤ 8192	保证对所有会议，有 $r = l + 1$
23	4				保证对所有会议，有 $l = 1, r = n$
24	4				保证所有人大代表的见面关系不改变，即没有 T 事件
25	3				

医院（tree）

【题目描述】

S 星球的人们为了尽快让患 COVID-12243 的病人进入医院救治，准备建造一座「水神山医院」。

在设计图中，「水神山医院」分为 n 个房间 (包括病房和门诊室)，标号为 $1, 2, \dots, n$ ，某些房间之间可以通过一条通道相连。定义一条路径由若干个不同的通道首尾相接而组成，由于某些特殊的原因，每两个房间之间~~有且仅有一条路径~~使它们可以互相到达。

S 星球有 $10^{10^{10}}$ 个工人，她们被安排到「水神山医院」的建设中，每个人必须恰好负责一个房间的施工，每个房间至少需要派遣一个人来施工。

设 i 号房间分配了 w_i 个工人来施工。显然有 $\sum_{i=1}^n w_i = 10^{10^{10}}$ 。

小 ω 作为「水神山医院」工程的经理，为了尽速建完「水神山医院」，需要对每个房间分配合适数量的工人。具体地，一个分配方案的好坏由它的合作指数与矛盾指数来衡量。

对于医院中的每一个通道，它所连接的两个房间 u, v 的工人们可以促进交流，于是会产生 $c \cdot w_u \cdot w_v$ 的合作指数，其中 c 是依赖于这条通道的一个正常数。

而对于医院中的每一个房间，如果里面的工人太多，则会造成拥挤、竞争、咕咕咕等不利因素，反而办不成事 (三个和尚没水喝)，于是一个有 w_i 人的房间会产生 $c \cdot w_i^2$ 的矛盾指数，其中 c 是依赖于房间 i 的一个正常数。

定义一个分配方案的合作指数为所有通道的合作指数之和，矛盾指数为所有房间的矛盾指数之和。

已知，如果 $\frac{\text{合作指数}}{\text{矛盾指数}}$ 的值越大，施工效率就会越高，人民就会越满意。

所以小 ω 希望让 $\frac{\text{合作指数}}{\text{矛盾指数}}$ 的数值尽可能大，然而人太多了，她的暴力程序无法在给定时间内得到结果，而上面又有规定，在 10 天内必须完成医院的所有施工，于是她找到了你，希望你帮她进行合理规划。

【输入格式】

从文件 `tree.in` 中读入数据。

第一行包含一个正整数 n ，表示医院的房间个数。

第二行包含 n 个正整数 a_1, a_2, \dots, a_n ，表示每个房间的矛盾指数的系数。

接下来 $n - 1$ 行，每行三个正整数 u_i, v_i, b_i ，分别表示房间 u_i 和房间 v_i 之间有一条通道相连，且它的合作指数的系数为 b_i 。

【输出格式】

输出到文件 `tree.out` 中。

为了方便，你只需要输出 $\frac{\text{合作指数}}{\text{矛盾指数}}$ 的最大值即可，显然这个最大值存在。

输出一行一个实数，表示 $\frac{\text{合作指数}}{\text{矛盾指数}}$ 的最大值。答案被认为正确当且仅当相对或绝对误差不超过 10^{-5} 。

我们保证给出的参考答案与真实答案的相对或绝对误差不超过 10^{-11} 。

【样例 1 输入】

```
2
1 1
1 2 1
```

【样例 1 输出】

```
0.5
```

【样例 1 解释】

在两个房间分别分配 5×10^{999999999} 个工人，则合作指数为 $2.5 \times 10^{19999999999}$ ，矛盾指数为 $5 \times 10^{19999999999}$ ，于是 $\frac{\text{合作指数}}{\text{矛盾指数}}$ 的值为 $\frac{1}{2} = 0.5$ 。

可以证明，这是使得 $\frac{\text{合作指数}}{\text{矛盾指数}}$ 的值最大的方案。

【样例 2 输入】

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

【样例 2 输出】

```
1
```

【样例 2 解释】

在 1 号房间分配 $3.33333332 \times 10^{999999999}$ 个工人，其余每个房间分配 $1.66666667 \times 10^{999999999}$ 个工人，计算可知 $\frac{\text{合作指数}}{\text{矛盾指数}}$ 的值约为 $1 - 2 \times 10^{-17}$ 。

而另一方面，可以证明在这个例子下，合作指数不会超过矛盾指数，因而最大值不会超过 1。于是 $\frac{\text{合作指数}}{\text{矛盾指数}}$ 的最大值介于 $(1 - 2 \times 10^{-17}, 1)$ 之间，于是只需输出 1 即可。

【样例 3 输入】

```
5
4 2 7 5 6
1 2 3
1 3 8
2 4 1
2 5 9
```

【样例 3 输出】

```
1.44877936767
```

【样例 4】

见选手目录下的 *tree/tree4.in* 与 *tree/tree4.ans*。
该组样例满足 $a_i = 1$ ，通道结构为 P。

【样例 5】

见选手目录下的 *tree/tree5.in* 与 *tree/tree5.ans*。
该组样例满足 $b_i = 1$ ，通道结构为 S。

【样例 6】

见选手目录下的 *tree/tree6.in* 与 *tree/tree6.ans*。
该组样例满足 $a_i = b_i = 1$ 。

【数据范围】

对于所有的测试点，均满足 $2 \leq n \leq 10^5; 1 \leq a_i, b_i \leq 100; 1 \leq u_i, v_i \leq n; u_i \neq v_i$ ，且如果将 (u_i, v_i) 看成边，则所有 $n - 1$ 条边恰好构成一棵树。

具体的子任务的数据规模见下表：

子任务	分值	n	a_i	b_i	通道结构
1	4	$= 2$	≤ 100	≤ 100	无
2	3	$= 3$			
3	3	$= 4$			
4	3	$= 6$			
5	3	$= 9$			
6	2	$= 12$			
7	2	$= 16$			
8	4	≤ 50	$= 1$	≤ 100	P
9	4		≤ 100	$= 1$	S
10	2			≤ 100	无
11	4	≤ 400		$= 1$	P
12	4		≤ 100	≤ 100	S
13	2			无	
14	4	≤ 3000	$= 1$	$= 1$	P
15	4				S
16	2				无
17	3		≤ 100	≤ 100	P
18	3				S
19	4	≤ 20000	$= 1$	$= 1$	P
20	4				S
21	3				无
22	4		≤ 100	≤ 100	P
23	4				S
24	3				无
25	4	$\leq 10^5$	$= 1$	$= 1$	P
26	4				S
27	3				无
28	4		≤ 100	≤ 100	P
29	4				S
30	3				无

表中“通道结构”一栏，变量的含义如下：

- P (path)：所有通道满足 $u_i = i, v_i = i + 1$ 。
- S (star)：所有通道满足 $u_i = 1, v_i = i + 1$ 。