



郑州轻工业大学
ZHENGZHOU UNIVERSITY OF LIGHT INDUSTRY

2024 年中国大学生程序设计竞赛全国邀请赛（郑州）

暨第六届 CCPC 河南省大学生程序设计竞赛

正式赛

2024 年 5 月 12 日

题目概况

| 题号 | 题目名 | 时间限制 | 空间限制 |
|----|-----------------|-------|------|
| A | Once In My Life | 1 s | 1 GB |
| B | 扫雷 1 | 1 s | 1 GB |
| C | 中二病也要打比赛 | 2 s | 1 GB |
| D | 距离之比 | 1 s | 1 GB |
| E | 保卫城邦 | 4.5 s | 1 GB |
| F | 优秀字符串 | 1 s | 1 GB |
| G | 扫雷 2 | 1 s | 1 GB |
| H | 随机栈 | 1 s | 1 GB |
| I | 378QAQ 和字符串 | 4 s | 1 GB |
| J | 排列与合数 | 1 s | 1 GB |
| K | 树上问题 | 1 s | 1 GB |
| L | Toxel 与 PCPC II | 1 s | 1 GB |
| M | 有效算法 | 2 s | 1 GB |



Problem A. Once In My Life

对于小 A 而言，数位包含 $1 \sim 9$ ，并且至少两个数位是 d ($1 \leq d \leq 9$) 的十进制正整数都是幸运数。

当 $d = 3$ 时，显然 1234567890123 是小 A 的幸运数，但 987654321 因为数位 3 仅出现了一次而不是幸运数，998244353 因为缺少数位 1, 6, 7 而不是幸运数。

现在小 A 有一个正整数 n ，并给出正整数 d 。他想找到正整数 k 使得二者的乘积 $n \cdot k$ 是幸运数。你能用计算机辅助他的计算吗？

输入格式

本题测试点包含多组数据。

第一行，一个正整数 T ($1 \leq T \leq 3 \times 10^5$)，表示数据组数。

对于每组数据：

一行，两个正整数 n, d ($1 \leq n \leq 10^8, 1 \leq d \leq 9$)。

输出格式

对于每组数据：

输出一行，一个正整数 k ，满足 $n \cdot k$ 是幸运数。你需要保证 $k \leq 2 \times 10^{10}$ 。

样例

| standard input | standard output |
|----------------|-----------------|
| 3 | 1234567896 |
| 1 6 | 404 |
| 12345678 9 | 9217006 |
| 233 2 | |

提示

对于 $n = 1, d = 6$ ，可以取 $k = 1234567896$ ，有 $n \cdot k = 1234567896$ 为幸运数。

对于 $n = 12345678, d = 9$ ，可以取 $k = 404$ ，有 $n \cdot k = 4987653912$ 为幸运数。

对于 $n = 233, d = 2$ ，可以取 $k = 9217006$ ，有 $n \cdot k = 2147562398$ 为幸运数。

Problem B. 扫雷 1

T0xel 喜欢玩扫雷，但是他玩的扫雷游戏有名为“地雷探测器”的特殊道具。

具体来说，T0xel 会进行 n 轮扫雷。每轮扫雷开始之前，T0xel 会获得 1 枚扫雷币。扫雷币在每轮扫雷结束后不会回收，可以保留至下一轮扫雷。T0xel 知道，在第 i 轮 ($1 \leq i \leq n$) 扫雷中，花费 c_i 枚扫雷币可以购买一个地雷探测器，清除地图中的一个雷。地雷探测器在一轮扫雷中可以购买任意次。

现在 T0xel 想知道，在这 n 轮扫雷中最多能购买多少个地雷探测器呢？

输入格式

第一行，一个正整数 n ($1 \leq n \leq 2 \times 10^5$)，表示扫雷轮数。

第二行， n 个正整数 c_1, c_2, \dots, c_n ($1 \leq c_i \leq 10^9$)。

输出格式

一行，一个非负整数，表示答案。

样例

| standard input | standard output |
|------------------|-----------------|
| 6 3 2 5 3 4 3 | 2 |
| 5 6 3 3 4 2 | 2 |
| 5 7 6 5 9 8 | 0 |

提示

对于第一个样例，T0xel 可以选择在第 2 轮与第 6 轮扫雷中各购买一个地雷探测器。具体过程如下：

- 获得 1 枚扫雷币，目前有 1 枚扫雷币。第 1 轮扫雷开始，不购买地雷探测器。
- 获得 1 枚扫雷币，目前有 2 枚扫雷币。第 2 轮扫雷开始，购买一个地雷探测器，目前有 0 枚扫雷币。
- 获得 1 枚扫雷币，目前有 1 枚扫雷币。第 3 轮扫雷开始，不购买地雷探测器。
- 获得 1 枚扫雷币，目前有 2 枚扫雷币。第 4 轮扫雷开始，不购买地雷探测器。
- 获得 1 枚扫雷币，目前有 3 枚扫雷币。第 5 轮扫雷开始，不购买地雷探测器。
- 获得 1 枚扫雷币，目前有 4 枚扫雷币。第 6 轮扫雷开始，购买一个地雷探测器，目前有 1 枚扫雷币。

对于第二个样例，T0xel 可以选择在第 5 轮扫雷中购买两个地雷探测器。

对于第三个样例，T0xel 无法在这 5 轮扫雷中购买地雷探测器。

Problem C. 中二病也要打比赛

在被中二病彻底占领的世界中，存在着一个被称为“现实”的神秘领域。在这个领域中，小鸟游六花，一位坚信自己拥有着非凡力量的中二病少女，发现了一串神秘的数字序列 A 。这个序列包含了 n 个元素，每个元素 A_i 是 1 到 n 之间的整数。据说，只有当这个序列满足单调不降的性质时，隐藏在其中的超自然力量才会觉醒。

六花相信，通过解开这个序列的秘密，她可以进一步证明自己的“邪王真眼”的力量。然而，她很快就意识到，要驯服这个序列，需要一种特殊的魔法——一个能将 A 转化为另一个序列的函数 f ，其定义域与值域均为 $[1, n] \cap \mathbb{Z}$ 。使用这个魔法后， A 会变成 B ，其中 $B_i = f(A_i)$ 。但是，这个魔法的使用是有代价的，其成本由 $[1, n] \cap \mathbb{Z}$ 中 $f(x) \neq x$ 的 x 数量决定。在这个充斥着中二病的世界中，六花必须以最小的代价激发序列中隐藏的力量。

现在，作为六花的冒险伙伴，你的任务是帮助她找到那个神奇的函数 f ，将 A 转化为单调不降序列，并以最小的代价揭示序列中隐藏的超自然力量。

输入格式

第一行包含一个整数 n ($1 \leq n \leq 2 \times 10^5$)，代表序列长度。

第二行包含 n 个整数 A_i ($1 \leq A_i \leq n$)，代表序列。

输出格式

一行，一个整数，表示最小代价。

样例

| standard input | standard output |
|-----------------------------|-----------------|
| 10 1 10 2 6 10 8 9 4 4 5 | 4 |

提示

在样例中，可以选择函数 f 为 $[1, 2, 3, 4, 5, 2, 7, 2, 2, 2]$ ，其中第 i 个数字 c_i 表示 $f(i) = c_i$ ，由该函数得到的序列 B 为 $1, 2, 2, 2, 2, 2, 2, 4, 4, 5$ ，符合题目条件。由于 $f(6) = 2 \neq 6$, $f(8) = 2 \neq 8$, $f(9) = 2 \neq 9$, $f(10) = 2 \neq 10$ ，所以该函数的代价是 4。可以证明没有比 4 更小的代价。

Problem D. 距离之比

对于 \mathbb{R}^2 平面上的两个点 $P(x_P, y_P)$ 与 $Q(x_Q, y_Q)$, PQ 之间的曼哈顿距离定义为

$$\|PQ\|_1 = |x_P - x_Q| + |y_P - y_Q|$$

而 PQ 之间的欧几里得距离定义为

$$\|PQ\|_2 = \sqrt{(x_P - x_Q)^2 + (y_P - y_Q)^2}$$

现在给出平面上互不重合的 n 个点 P_1, P_2, \dots, P_n , 请求出

$$\max_{1 \leq i < j \leq n} \frac{\|P_i P_j\|_1}{\|P_i P_j\|_2}$$

输入格式

本题测试点包含多组数据。

第一行, 一个正整数 T ($1 \leq T \leq 10^5$), 表示数据组数。

对于每组数据:

第一行, 一个正整数 n ($2 \leq n \leq 2 \times 10^5$), 表示平面上的点数。

接下来 n 行, 每行两个整数 x_i, y_i ($-10^9 \leq x_i \leq 10^9, -10^9 \leq y_i \leq 10^9$), 表示点 $P_i(x_i, y_i)$ 。

保证对于单个测试点有 $\sum n \leq 2 \times 10^5$ 。

输出格式

对于每组数据: 输出一行, 一个实数, 表示点对之间曼哈顿距离与欧几里得距离之比的最大值。当你的答案与标准答案的相对误差或绝对误差不超过 10^{-9} 时将视为正确答案。

样例

| standard input | standard output |
|----------------|-----------------|
| 2 | 1.000000000000 |
| 2 | 1.371988681140 |
| 0 0 | |
| 0 1 | |
| 3 | |
| 1 1 | |
| 2 3 | |
| 5 8 | |

提示

对于第一个样例, 给定的点为 $P_1(0, 0), P_2(0, 1)$, 有 $\frac{\|P_1 P_2\|_1}{\|P_1 P_2\|_2} = 1$ 。

对于第二个样例, 给定的点为 $P_1(1, 1), P_2(2, 3), P_3(5, 8)$, 有

- $\frac{\|P_1 P_2\|_1}{\|P_1 P_2\|_2} = \frac{3}{\sqrt{5}} \approx 1.34164079$
- $\frac{\|P_1 P_3\|_1}{\|P_1 P_3\|_2} = \frac{11}{\sqrt{65}} \approx 1.36438208$
- $\frac{\|P_2 P_3\|_1}{\|P_2 P_3\|_2} = \frac{8}{\sqrt{34}} \approx 1.37198868$

因此答案为 $\frac{8}{\sqrt{34}}$ 。

Problem E. 保卫城邦

小团子是一个精明的君主，他掌管着一个拥有 n 个城邦的国家，城邦间有 $n - 1$ 条高速路进行连接，使得从任意一个城邦出发可以经由高速路到达任意其他城市，高速路使得其相连的两座城市之间可以快速进行支援。

天有不测风云，现在小团子的国家遭受了攻击，小团子需要在城邦内驻扎军队进行守卫，一个城邦内可以驻扎多支军队，也可以不驻扎军队。其中对于不驻扎军队的城邦，需要与其通过高速路直接相连的所有城邦共驻扎至少 2 支军队来确保其安全。

然而，在炮火的袭扰中，会出现一些高速路被损毁的情况。每当出现一条高速路损毁，小团子会立刻指挥强大的后勤修一条新的高速路，保证每时每刻所有城邦都可以通过 $n - 1$ 条高速路相互到达。

在战火中，共有 m 次高速路损毁发生，在快速变化的战局之下，纵使是小团子也难以计算如何对守军进行调整，于是他求助身为军师的你，希望你在每次道路出现损毁时，计算出再次修建好高速路后，至少需要多少支军队才能保证王国的安全。

输入格式

第一行两个整数 n ($2 \leq n \leq 2 \times 10^5$) 和 m ($1 \leq m \leq 2 \times 10^5$)，表示城邦数和高速路损毁数量。

接下来 $n - 1$ 行，每行两个整数 u 和 v ($1 \leq u, v \leq n, u \neq v$)，表示最初王国在城邦 u 与 v 之间存在一条高速路。

接下来 m 行，每行四个整数 u, v, a, b ($1 \leq u, v, a, b \leq n, u \neq v, a \neq b$)，表示城邦 u 与 v 之间的高速路损毁，小团子立刻指挥后勤在城邦 a 与 b 之间修建了一条新的高速路。保证修建新的高速路后，城邦间可通过高速路相互到达。

输出格式

输出共 m 行，每行一个整数，表示每次修建新的高速路后，王国内至少需要驻扎的军队数量。

样例

| standard input | standard output |
|---|-----------------------|
| 5 4 2 4 2 5 2 3 1 4 2 5 2 5 1 4 1 2 2 3 1 3 1 3 4 3 | 3 2 3 3 |
| 8 5 3 2 2 6 6 8 4 3 7 6 7 5 4 1 4 1 7 1 7 1 4 1 6 8 2 8 2 8 4 8 4 8 4 8 | 5 5 5 5 5 |

提示

- 在第一个样例中，高速路有 4 次损毁：
- 城邦 2 与城邦 5 之间的高速路损毁后，小团子在这两座城邦间重新修建了一条高速路，此时可以在城邦 2 驻扎 2 支军队，在城邦 1 驻扎 1 支军队来确保安全；
 - 城邦 1 与城邦 4 之间的高速路损毁后，小团子在城邦 1 与城邦 2 之间修建了一条高速路，此时仅需在城邦 2 驻扎 2 支军队即可确保安全；
 - 城邦 2 与城邦 3 之间的高速路损毁后，小团子在城邦 1 与城邦 3 之间修建了一条高速路，此时可以在城邦 2 驻扎 2 支军队，城邦 3 驻扎 1 支军队来确保安全；
 - 城邦 1 与城邦 3 之间的高速路损毁后，小团子在城邦 4 与城邦 3 之间修建了一条高速路，此时仍然可以通过在城邦 2 驻扎 2 支军队，在城邦 3 驻扎 1 支军队来确保安全。

Problem F. 优秀字符串

小 A 认为，一个字符串 S 是优秀字符串，当且仅当：

- S 的长度 $|S|$ 恰好为 5；
- S 的第三个字符与第五个字符相同；
- S 的前四个字符互不相同。

例如 `henan` 是优秀字符串，但 `query`、`problem`、`queue` 不是，因为：

- `query` 的第三个字符为 `e`，而第五个字符为 `y`；
- `problem` 的长度不为 5；
- `queue` 的前四个字符中 `u` 出现了两次。

现在，小 A 有 n 个仅包含英文字母与数字的字符串 S_1, S_2, \dots, S_n ，请你帮小 A 求出这些字符串中优秀字符串的数量。

输入格式

第一行，一个正整数 n ($1 \leq n \leq 10^5$)，表示字符串的数量。

接下来 n 行，每行一个仅包含英文字母与数字的字符串 S_i 。保证 $\sum |S_i| \leq 2 \times 10^5$ 。

输出格式

一行，一个整数，表示给定字符串中优秀字符串的数量。

样例

| standard input | standard output |
|---|-----------------|
| 4 henan query problem queue | 1 |

Problem G. 扫雷 2

T0xel 喜欢玩扫雷，但是他不喜欢数字 2。

他想构造一个 $n \times n$ 的扫雷的地图，其中有 m 个雷，并且没有一个空地周围恰有 2 个雷。

也就是说，他想构造一个 01 方阵，使得不存在一个 0 周围 8 格中恰有 2 个 1。特别地，边上的 0 周围 5 个格不能恰有 2 个 1，角落上的 0 周围 3 个格不能恰有 2 个 1。

输入格式

本题测试点包含多组数据。

第一行，一个正整数 T ($1 \leq T \leq 100$)，表示数据组数。

每组数据包含一行，两个整数 n, m ($5 \leq n \leq 1000, 1 \leq m \leq n \times n$)，表示地图大小和雷数。

保证单个测试点所有数据的 $\sum n^2 \leq 10^6$ 。

输出格式

对于每组数据：

如果有解，先输出一行 **Yes**，然后输出一个 $n \times n$ 的 01 矩阵，其中 0 表示空地，1 表示雷。

如果无解，输出一行 **No**。

样例

| standard input | standard output |
|----------------|-----------------|
| 2 | Yes |
| 5 2 | 10000 |
| 5 17 | 00000 |
| | 00000 |
| | 10000 |
| | 00000 |
| | Yes |
| | 11111 |
| | 10001 |
| | 10101 |
| | 10001 |
| | 11111 |

Problem H. 随机栈

Toxel 获得了一个随机的“栈”。这个栈可被视为一个多重集 S ，从一个非空的随机栈 S 中取出一个元素时，有可能从中取出任何一个元素，其中每个元素被取出的概率是相等的。取出该元素后，该元素会从集合中删除。以 $\{1, 2, 2\}$ 为例，有 $\frac{1}{3}$ 的概率取出 1，使得集合变为 $\{2, 2\}$ ，有 $\frac{2}{3}$ 的概率取出 2，使得集合变为 $\{1, 2\}$ 。每次取出元素的事件相互独立。

Toxel 正在对这个集合做一些操作。集合初始时空，它总共进行了 $2n$ 次操作，其中 n 次操作为插入， n 次操作为取出。现在，Toxel 告诉了你它操作的顺序以及每次插入的数，且保证每次取出时，集合非空。Toxel 想知道，如果把每次取出的数排成一个序列，那么这个序列递增的概率是多少？这里，递增的严格定义是：取出数列的每一项（除最后一项）小于等于它的后一项。

由于答案可能不是整数，为了方便计算，你只需要求出这个值对 998 244 353 取模的结果。

输入格式

第一行包含一个整数 n ($1 \leq n \leq 2 \times 10^5$)。

第二行包含 $2n$ 个整数 a_1, a_2, \dots, a_{2n} ($-1 \leq a_i \leq n$)，表示 Toxel 操作的序列。其中，若 $0 \leq a_i \leq n$ ，表示 Toxel 向集合中插入了 a_i ；否则 $a_i = -1$ ，表示 Toxel 从集合中取出了一个元素。数据保证取出元素时，集合非空；保证插入和取出操作的次数分别为 n 。

输出格式

输出一行一个整数，表示答案对 998 244 353 取模的结果。

样例

| standard input | standard output |
|--------------------------|-----------------|
| 2 1 2 -1 -1 | 499122177 |
| 3 1 2 -1 -1 1 -1 | 0 |
| 4 1 -1 2 -1 3 -1 4 -1 | 1 |

提示

正式地说，答案对 998 244 353 取模表达了如下含义。令 $M = 998\,244\,353$ ，可以证明答案可表示为既约分数 $\frac{p}{q}$ ，其中 p 和 q 均为整数，且 $q \not\equiv 0 \pmod{M}$ 。你需要输出 $p \cdot q^{-1} \pmod{M}$ 。换句话说，你需要输出满足 $0 \leq x < M$ 且 $x \cdot q \equiv p \pmod{M}$ 的整数 x 。

对于样例一，可能有以下两种情况：

- 加入 1 后，集合变成 $\{1\}$ ，加入 2 后，集合变成 $\{1, 2\}$ 。接下来先取出 1，这里有 $\frac{1}{2}$ 的概率，接下来再取出 2。这种情况下，取出的序列为 1, 2，是递增的，概率为 $\frac{1}{2}$ 。
- 加入 1 后，集合变成 $\{1\}$ ，加入 2 后，集合变成 $\{1, 2\}$ 。接下来先取出 2，这里有 $\frac{1}{2}$ 的概率，接下来再取出 1。这种情况下，取出的序列为 2, 1，不是递增的，概率为 $\frac{1}{2}$ 。

有序的概率为 $\frac{1}{2}$, 而 $2 \cdot 499\,122\,177 \equiv 1 \pmod{998\,244\,353}$, 故答案为 499 122 177。

对于样例二, 2 无论如何都会在第二个 1 前被取出, 递增的概率为 0。

对于样例三, 取出的序列只有 1, 2, 3, 4 一种情况, 递增的概率为 1。

Problem I. 378QAQ 和字符串

378QAQ 有一个长度为 n 的仅包含小写字母的字符串 s 。他认为一个字符串是美丽的，当且仅当存在 p ($1 \leq p \leq \frac{n}{2}$)，使得对于所有的 i ($0 \leq i \leq n - p - 1$) 有 $s_i = s_{i+p}$ 。其中， s_i 表示字符串 s 的第 i 个字符，下标从 0 开始编号。

378QAQ 想知道能否修改字符串 s 中的至多 k 个字符，使得修改后的字符串 s' 是美丽的。

输入格式

本题测试点包含多组数据。

第一行包含一个正整数 t ($1 \leq t \leq 1000$)，表示数据组数。

对于每组数据：

第一行包含两个正整数 n, k ($2 \leq n \leq 3 \times 10^5, 1 \leq k \leq 100$)，表示字符串 s 的长度，以及至多修改的字符个数。

第二行包含一个由 n 个小写字母组成的字符串 s 。

保证单个测试点中各组数据 n 的总和不超过 3×10^5 。

输出格式

对于每组数据，如果能够修改字符串 s 中的至多 k 个字符使得修改后的字符串 s' 是美丽的，输出 “Yes”（不含引号），否则输出 “No”（不含引号）。

样例

| standard input | standard output |
|----------------|-----------------|
| 5 | Yes |
| 6 1 | No |
| yesyrs | Yes |
| 6 2 | Yes |
| bazoka | No |
| 9 2 | |
| zzaazazzc | |
| 11 5 | |
| bazokamocha | |
| 11 4 | |
| bazokamocha | |

提示

对于第一组样例，可以将 s_4 从 “r” 修改为 “e”，此时 $s' = \text{yesyes}$ ，存在 $p = 3$ 使得对于所有的 i ($0 \leq i \leq n - p - 1$)， $s'_i = s'_{i+p}$ 。

Problem J. 排列与合数

小 A 在 2023 年河南省 CCPC 大学生程序设计竞赛的赛场上遇到了一道名为“排列与质数”的题目。与大多数选手一样，小 A 并没能在赛场上解决这个棘手的题目。比赛结束后，小 A 想到了一个与之相关的题目：排列与合数，可是小 A 仍然没有能力解决。这个名为“排列与合数”的题目是这样的：

给定一个有且仅有 5 位，且各个数位互不相同的十进制正整数 n 。你可以重新排列 n 的各个数位，但需要保证重新排列得到的整数 n' 没有前导零。请问重新排列数位得到的 n' 能否为合数？若能为合数，请求出一个满足条件的 n' 。

例如，当 $n = 12345$ 时，任意排列得到的 n' 均是合数，因此可以任意取 n' 。当 $n = 13579$ 时，可以重新排列数位得到合数 $n' = 97531 = 7 \times 13933$ 。

一个正整数是合数，当且仅当它可以分解为两个不小于 2 的整数的乘积。

现在，小 A 带着他的题目来到赛场上求助。你能帮助小 A 解决这个问题吗？

输入格式

本题测试点包含多组数据。

第一行，一个正整数 T ($1 \leq T \leq 10^5$)，表示数据组数。

对于每组数据：

一行，一个正整数 n ($10^4 \leq n < 10^5$)，保证 n 的各个数位互不相同。

输出格式

对于每组数据：

输出一行，一个整数。若能重新排列 n 的数位得到合数 n' 则输出 n' ，否则输出 -1 。

样例

| standard input | standard output |
|----------------|-----------------|
| 5 | 12345 |
| 12345 | 54321 |
| 12345 | 13524 |
| 12345 | 45123 |
| 12345 | 97531 |
| 13579 | |

提示

样例即是题目描述中给出的例子。

Problem K. 树上问题

378QAQ 有一棵由 n 个节点组成的无根树，节点编号从 1 到 n ，每个节点有一个正整数点权。

378QAQ 认为一个节点是美丽节点，当且仅当该节点作为根时，对于除根节点以外的所有节点，其点权都不小于其父亲节点的点权的 $\frac{1}{2}$ 。

请你计算出有多少个节点是美丽节点。

输入格式

本题测试点包含多组数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^4$)，表示数据组数。

对于每组数据：

第一行包含一个正整数 n ($1 \leq n \leq 10^5$)，表示节点数量。

第二行包含 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$)，表示编号为 i 的节点点权。

之后 $n - 1$ 行，每行包含两个正整数 u, v ($1 \leq u, v \leq n, u \neq v$)，表示无根树中存在一条连接节点 u 和节点 v 的边。

保证单个测试点中所有数据的 $\sum n \leq 10^5$ 。

输出格式

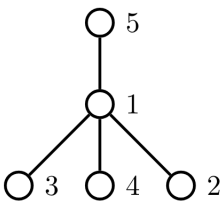
对于每组数据，输出一个非负整数，代表美丽节点的数量。

样例

| standard input | standard output |
|---------------------------------|-----------------|
| 3 | 3 |
| 3 | 1 |
| 1 2 3 | 7 |
| 1 2 | |
| 2 3 | |
| 5 | |
| 3 2 2 2 1 | |
| 1 2 | |
| 3 1 | |
| 4 1 | |
| 1 5 | |
| 8 | |
| 699 673 592 276 600 343 369 374 | |
| 7 6 | |
| 8 5 | |
| 4 6 | |
| 7 1 | |
| 7 2 | |
| 1 8 | |
| 4 3 | |

提示

对于第二组数据，树的形态如下：



只有节点 5 是美丽节点。当节点 5 作为根时，除根节点以外的各个节点与其父亲节点的点权关系如下：

- 节点 1 的父亲节点为节点 5，二者点权有 $3 \geq \frac{1}{2}$ ，满足要求。
- 节点 2,3,4 的父亲节点为节点 1，节点 2,3,4 的点权均有 $2 \geq \frac{3}{2}$ ，满足要求。

因此节点 5 是美丽节点。当其他任意节点作为根时，从上图可以看出节点 5 的父亲节点始终为节点 1，由于 $1 < \frac{3}{2}$ ，不满足要求，所以其余节点均不是美丽节点。

Problem L. Toxel 与 PCPC II

Toxel 正在参加 PCPC（Pokémon Center Programming Contest）比赛。它写的一段代码中有不少 bug，正在调试。这份代码总共有 n 行，而且经验丰富的 Toxel 已经知道了其中 m 行代码有 bug，并锁定了这 m 行的具体位置。但是 Toxel 还需要进行一些调试以了解错误的具体细节并修复它们。

Toxel 会进行多次调试。每次调试时，Toxel 可以任选一个 i ，使得程序从第 1 行开始，顺序运行完第 i 行后退出。Toxel 可以通过这 i 行代码运行的一些输出结果来进行 debug。运行这 i 行代码总共需要 i 秒。接下来，Toxel 会一次性地 debug 这 i 行代码，并修复所有这 i 行中的所有 bug。bug 数量越多，修复所需的时间也越多。设这 i 行代码中现存的 bug 数量为 x ，那么 Toxel 需要 x^4 秒来 debug 并完成修复。修复后，这 i 行代码中将不再存在任何 bug。

PCPC 的赛场争分夺秒。请你帮 Toxel 计算一下，它最短需要多少秒才能完成 debug，修复整个代码中的所有漏洞？

输入格式

第一行包含两个整数 n, m ($1 \leq m \leq n \leq 2 \times 10^5$)。

第二行包含 m 个整数 a_1, a_2, \dots, a_m ($1 \leq a_1 < a_2 < \dots < a_m \leq n$)，表示代码中所有有 bug 的行编号。

输出格式

输出一行一个整数，表示答案。

样例

| standard input | standard output |
|---|-----------------|
| 3 2 1 3 | 6 |
| 1 1 1 | 2 |
| 20 20 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 | 221 |

提示

对于第一个样例，Toxel 应该选择先运行前 1 行代码，运行消耗 1 秒，debug 消耗 $1^4 = 1$ 秒。接下来 Toxel 应该选择运行前 3 行代码，运行消耗 3 秒，debug 消耗 $1^4 = 1$ 秒。总计消耗 $1 + 1 + 3 + 1 = 6$ 秒。

对于第三个样例，Toxel 可以分别运行前 1, 2, 3, ..., 13, 14, 16, 18, 20 行来得到最优解。

Problem M. 有效算法

给出长度为 n 的正整数序列 $\{a_n\}$ 和 $\{b_n\}$ 。对于每个 a_i ($1 \leq i \leq n$)，进行恰好一次以下操作：

- 将 a_i 变成满足 $|a_i - x| \leq k \times b_i$ 的任意整数 x 。

请你求出最小的非负整数 k ，使得存在至少一种方法使得操作后序列 $\{a_n\}$ 所有数都相等。

输入格式

本题测试点包含多组数据。

第一行包含一个正整数 T ($1 \leq T \leq 1.5 \times 10^5$)，表示数据组数。

对于每组数据：

第一行包含一个正整数 n ($2 \leq n \leq 3 \times 10^5$)。

第二行包含 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$)。

第三行包含 n 个正整数 b_1, b_2, \dots, b_n ($1 \leq b_i \leq 10^9$)。

保证单个测试点中所有数据的 $\sum n \leq 3 \times 10^5$ 。

输出格式

对于每组数据：

输出一行一个整数，表示答案。

样例

| standard input | standard output |
|----------------|-----------------|
| 2 | 2 |
| 4 | 2 |
| 8 3 3 5 | |
| 1 2 3 2 | |
| 5 | |
| 4 3 4 5 6 | |
| 3 1 3 1 1 | |

提示

对于样例一，可以令 a_i 全变为 6。

对于样例二，可以令 a_i 全变为 5。