

2024 “钉耙编程” 中国大学生算法设计超级联赛（5）

2024 年 8 月 2 日

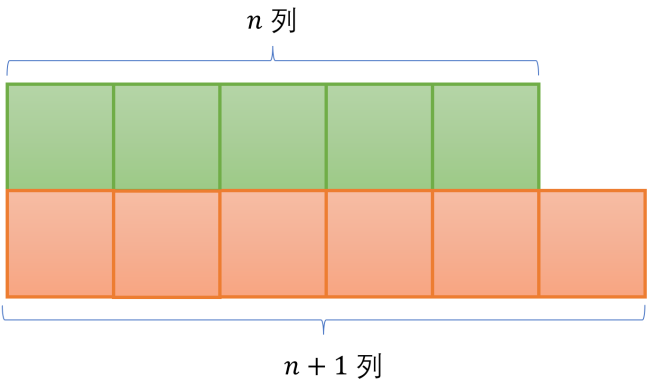
目录

1 数表（二）	2
2 Array Gift	4
3 捆绑魔方	6
4 树论（一）	9
5 树论（二）	11
6 猫罐头游戏	13
7 猫咪军团	14
8 猫咪们狂欢	16
9 用心感受（三）	18
10 世末农庄	19
11 开关灯	22
12 串串	23
13 飞行棋	24

Problem 1 数表（二）

Time limit: 1 seconds
Memory limit: 256 Megabytes

你得到了一张 2 行的表格，其中第一行有 n 列，第二行有 $n + 1$ 列，第一行的第一列和第二行第一列对齐，如图示：



现在需要在这一共 $2n + 1$ 个格子中都填入一个整数，需要满足：

- 1、填的整数在 $[0, 2^k - 1]$ 之间。
- 2、同一行不能有相同的数，同一列不能有相同的数。
- 3、所有 $2n + 1$ 个数的异或和恰好是 0.

现在 Cuking 想问你，有多少种不同的填数方案，由于答案可能很大，你需要输出答案在模 998 244 353 下的结果。

Input

第一行一个整数 $T(1 \leq T \leq 30)$ 表示测试用例数量。
对每个测试用例，输入两个整数 n, k ， $1 \leq n \leq 10^6, 1 \leq k \leq 10^9$ ，和题目中的 n, k 对应。

Output

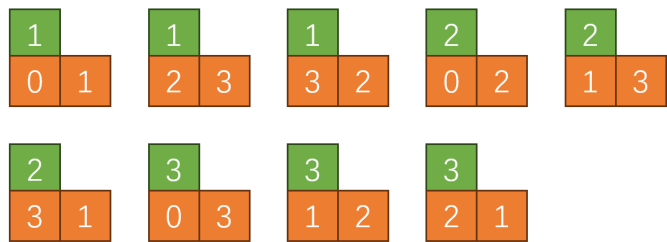
对每组格式用例，输出一行一个整数表示答案。

Example

standard input	standard output
2	9
1 2	410645142
114 514	

Hints

$n = 1, k = 2$ 的 9 种情况如下：



Problem 2 Array Gift

Time limit: 1 seconds
Memory limit: 256 Megabytes

最近开始流行赠送数组，所以 Beijixing 送给了 Liyishui 一个长度为 n 的**正整数**序列 $a = [a_1, \dots, a_n]$ ，并附赠了两种操作：

- 操作 1：选择不同的下标 i, j （即 $1 \leq i, j \leq n, i \neq j$ ），且 $a_j \neq 0$ ，然后将 a_i 修改为 $a_i \bmod a_j$ ¹
- 操作 2：选择某个 i 满足 $1 \leq i \leq n$ 和任意一个**正整数** x ，将 a_i 修改为 $a_i + x$ 。

Liyishui 喜欢倒腾数组，她希望只使用这两种操作让 a 只剩**恰好一个**非 0 数，其他的数都变成 0。两种操作均可使用任意次。不幸的是 Liyishui 最近忙着赶 ddl，你能帮忙求出**最小操作次数**吗？

Input

第一行输入一个正整数 T ，表示一共有 T 组测试用例， $1 \leq T \leq 30$ 。
接下来每组测试用例由两行构成：其中第一行输入一个正整数 $n(1 \leq n \leq 100)$ ，表示序列 a 的长度。第二行输入 n 个正整数 a_1, \dots, a_n ，相邻两数用一个空格隔开，表示序列 a 。对 $1 \leq i \leq n$ ，有 $1 \leq a_i \leq 10^6$ 。

Output

对每个测试用例，输出一行一个整数，表示最小操作次数。

Example

standard input	standard output
2	4
4	3
2 3 5 7	
4	
2 4 6 8	

¹ $a_i \bmod a_j$ 表示 a_i 除 a_j 的余数，形式化地说，是满足『 $a_i = a_j \times p + r$ 且 p 为整数』的最小非负整数 r 。例如 $3 \bmod 2 = 1, 4 \bmod 2 = 0$ 。

Hints

对于第一个样例，一种可能操作如下： $[2, 3, 5, 7] \rightarrow [2, \mathbf{1}, 5, 7] \rightarrow [\mathbf{0}, 1, 5, 7] \rightarrow [0, 1, \mathbf{0}, 7] \rightarrow [0, 1, 0, \mathbf{0}]$ ，一共用了 4 次操作，可以证明这是最少可能的操作次数。

对于第二个样例，一种可能操作如下： $[2, 4, 6, 8] \rightarrow [2, \mathbf{0}, 6, 8] \rightarrow [2, 0, \mathbf{0}, 8] \rightarrow [2, 0, 0, \mathbf{0}]$ ，一共用了 3 次操作，可以证明这是最少可能的操作次数。

Problem 3 捆绑魔方

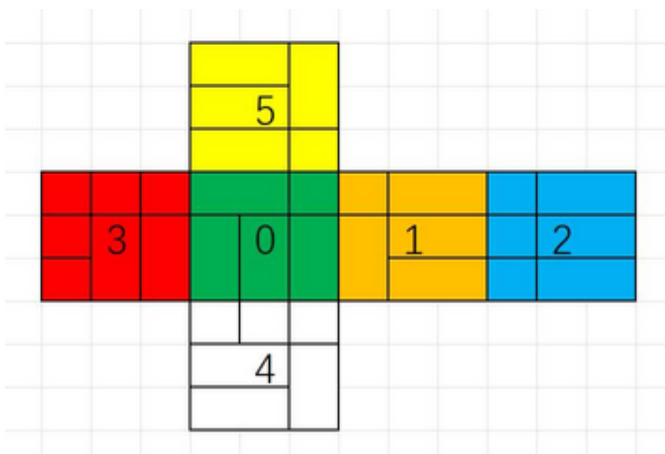
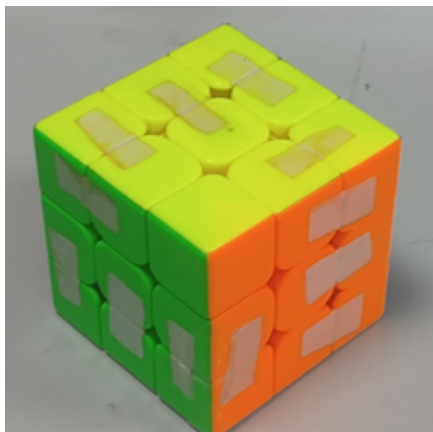
Time limit: 1 seconds

Memory limit: 512 Megabytes

三阶魔方是一款经典玩具，由 26 个小立方体（中心块 6 个、棱块 12 个、角块 8 个）组成，每个面都可以自由转动。

捆绑魔方是由标准的三阶魔方将若干相邻的块捆绑粘合而成，这些捆绑的块无法被分离，使得魔方在旋转时可能会被阻挡，即每个面并非在任意时刻都能转动

下面给出捆绑魔方的还原状态实物图及其外表面平面展开图（捆绑方式是唯一的，所有的询问都使用这种魔方）



给你一个打乱后的捆绑魔方，请求解出还原所需的**最小步数**，并输出步数最少的操作序列中**字典序最小**的那个序列

每次操作，可以选择一个面并将其顺时针旋转 90 度 (请不要问为什么魔方只能顺时针旋转)

Input

第一行包含一个正整数 $T(1 \leq T \leq 20)$ 表示测试用例组数。
对每组询问：以样例的格式，输入捆绑魔方打乱后的平面展开图。保证所有展开图的展开方式是相同的，且所有 6 个中心块在展开图中的对应位置不变。
简单来说，所有的输入为以下格式，其中 $0 \leq x \leq 5$ ，表示该部分的颜色

```

      x x x
      x 5 x
      x x x
x x x x x x x x x x x x
x 3 x x 0 x x 1 x x 2 x
x x x x x x x x x x x x
      x x x
      x 4 x
      x x x
```

保证给定状态合法且有解。

Output

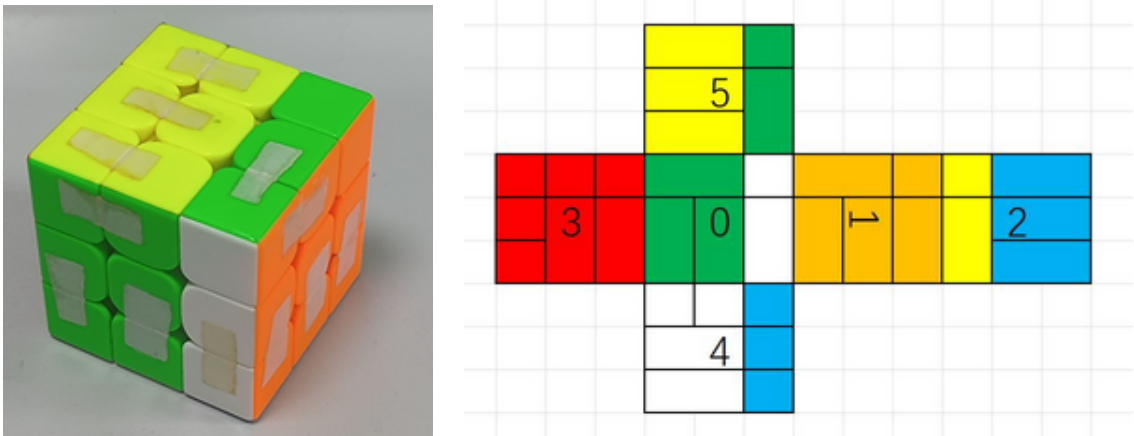
对每个测试用例，输出两行：
第一行输出一个整数 k ，表示还原所需 **最小** 操作步数。
第二行输出一个长度为 k 的整数操作序列 $p_1, p_2, \dots, p_k (0 \leq p_i \leq 5)$ ，表示对应的操作序列，其中 p_i 表示：第 i 步操作为将中心块颜色为 p_i 的面**顺时针旋转 90 度**，两个操作之间以空格隔开，若有多种操作序列满足步数最小，请输出其中**字典序最小**的序列。

Example

standard input	standard output
1 5 5 0 5 5 0 5 5 0 3 3 3 0 0 4 1 1 1 5 2 2 3 3 3 0 0 4 1 1 1 5 2 2 3 3 3 0 0 4 1 1 1 5 2 2 4 4 2 4 4 2 4 4 2	3 1 1 1

Hints

样例 1 对应的展开图及实物图



（此种状态下可行的合法操作为 1，2，5，其余操作会被阻挡）

提示：该捆绑魔方共有 1,108,800 种不同的状态

在线模拟传统三阶魔方：<https://rubiks-cube-solver.com/>

Problem 4 树论 (一)

Time limit: 4 seconds

Memory limit: 256 Megabytes

一天, 学傻了的 Yoshinow 在学数论时, 迷失在了数论的黑暗森林里!

(因为是在数论森林里!) 现在 Yoshinow 得到了一棵树, 现在他想知道: 在某个以 r 为根的子树内, 有多少点对 (i, j) , 满足 i, j 的最小公倍数² 不超过 x (注意是编号的 lcm)

由于 Yoshinow 非常——好奇, 所以他现在一共有 Q 个这样的询问。

形式化地说: 给定 n 个结点的有根树 (以 1 为根), 结点标号从 1 到 n , 共有 Q 次询问, 每次询问给出两个参数 r, x , 表示询问有多少点对 (i, j) , 满足:

- 1、 i, j 是 $[1, n]$ 内的正整数。
- 2、 $\text{lcm}(i, j) \leq x$.
- 3、标号为 i, j 的结点均在以 r 为根的子树内。

Input

第一行输入一个正整数 T , 表示测试用例组数, $1 \leq T \leq 3$.

对每组询问: 第一行输入一个整数 n , 其中 $1 \leq n \leq 10^5$.

接下来 $n - 1$ 行, 每行两个整数 $u, v (1 \leq u, v \leq n)$, 表示 u, v 之间有一条边。保证给出的 n 个点构成树。

接下来一行一个整数 Q , 表示询问组数, $1 \leq Q \leq 10^6$.

接下来 Q 行, 每行两个整数 $r, x (1 \leq r \leq n, 0 \leq x \leq 10^5)$, 表示一个询问。

Output

共 T 行, 对每组测试用例, 输出一行共 Q 个整数, 按顺序给出对应询问的答案, 相邻整数用空格隔开。

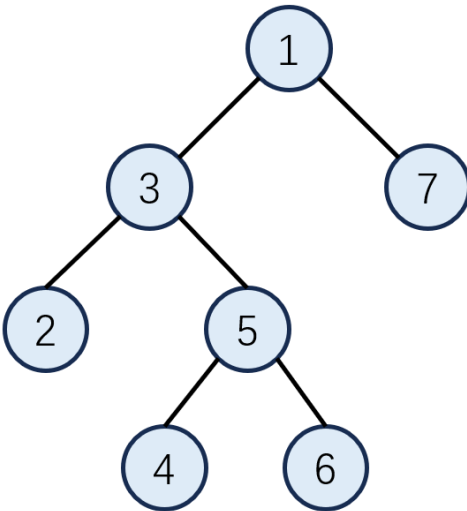
²对于正整数 x, y , $\text{lcm}(x, y)$ 表示 x, y 的最小公倍数, 即同时是 x, y 倍数的最小正整数。例如, 10 和 12 的最小公倍数是 60, 即 $\text{lcm}(10, 12) = 60$.

Example

standard input	standard output
1	23 3 9 15 27
7	
1 3	
1 7	
3 2	
3 5	
5 4	
5 6	
5	
3 23	
5 10	
5 30	
1 5	
1 7	

Hints

样例给出的树如图所示：



对于第一个询问， $r = 3, x = 23$ ， $r = 3$ 的子树内共有 5 个点，有 $5^2 = 25$ 对点对，除了 $(5, 6), (6, 5)$ 的最小公倍数是 $30 > 23$ 外，其余每个点对都满足最小公倍数不超过 23 的条件，因此答案是 $25 - 2 = 23$.

Problem 5 树论（二）

Time limit: 4 seconds

Memory limit: 512 Megabytes

Yoshinow 又得到了一棵 n 个结点的树，结点的标号为 1 到 n 内的整数，Yoshinow 突发奇想：如果把树上的一条边删掉后，一棵树会被划成两棵树 T_1, T_2 ，从 T_1 的节点编号中选一个数 x ，再从 T_2 的节点编号中选一个数 y ， $\gcd(x, y)$ 最大能取到多少？³ 因为 Yoshinow 非常好奇，所以他想对每条边都询问。

形式化地说：有一棵树 $G = (V, E)$ ， $V = \{1, 2, \dots, n\}$ ， $n-1$ 条边 $E = \{(u_1, v_1), \dots, (u_{n-1}, v_{n-1})\}$ 。对每条边 $(u_i, v_i) (i = 1, \dots, n-1)$ ，需要回答：将 (u_i, v_i) 删去后，原树 G 会被分成不连通的两棵树 $G_1 = (V_1, E_1)$ ， $G_2 = (V_2, E_2)$ ，此时 $\max_{x \in V_1, y \in V_2} \gcd(x, y)$ 是多少

Input

第一行一个正整数 T ，表示测试用例组数， $1 \leq T \leq 10$ 。

接下来 T 组数据，对每组数据：先输入一个正整数 n ，表示树的点数， $2 \leq n \leq 10^6$ ，接下来 $n-1$ 行，第 i 行包括两个整数 u_i, v_i ，表示树上的第 i 条边为 (u_i, v_i) 。

保证对所有测试用例， $\sum n \leq 5 \times 10^6$ 。

Output

对每个测试用例，输出一行 $n-1$ 个整数，第 i 个整数表示删除边 (u_i, v_i) 后的答案。

注：在这题中你可能需要用到更大的栈空间，C++ 选手可以在 main 函数中加入如下代码，以防出现栈溢出的错误：

```
int main() {
    int size(256<<20); //256M
    __asm__ ( "movq %0, %%rsp\n"::"r"((char*)malloc(size)+size));
    // YOUR CODE
    //...
    exit(0);
}
```

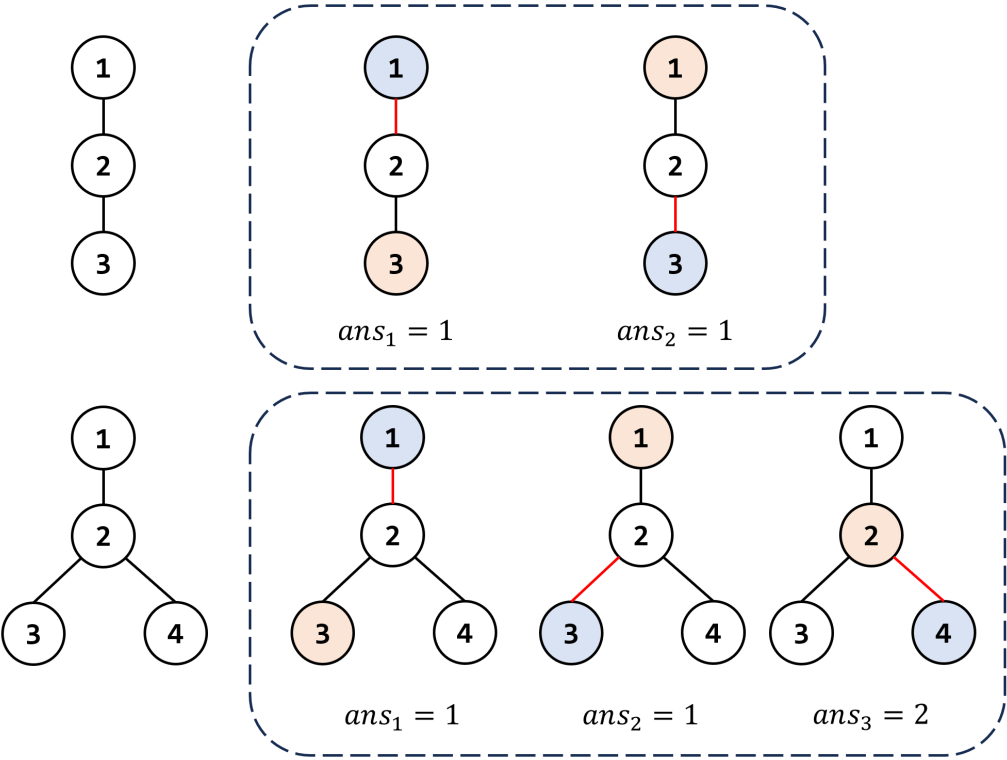
³对于正整数 x, y 来说， $\gcd(x, y)$ 表示同时整除 x, y 的最大整数。

Example

standard input	standard output
2	1 1
3	1 1 2
1 2	
2 3	
4	
1 2	
2 3	
2 4	

Hints

两个样例如下图所示：



红边表示询问的边，选取的点 x, y 对应地用浅蓝色和橙色标注。在样例中，除了第二棵树的 ans_3 以外，其他询问中的无序对 $\{x, y\}$ 的选取方式都不唯一。

Problem 6 猫罐头游戏

Time limit: 1 seconds
Memory limit: 256 Megabytes

小小猫和勇者猫比赛吃猫罐头。

总共有 3 堆猫罐头，两猫轮流吃猫罐头，小小猫先手。每次可以选择 2 堆猫罐头，将其中 1 堆吃光，然后把剩下 1 堆分成 2 堆，且必须保证这 2 堆猫罐头的数量都大于 0。无法操作者输。

注意：必须选择猫罐头数量大于 0 的堆。按照规则，场上永远会存在且仅存在 3 堆猫罐头数量大于 0 的堆。

假设小小猫和勇者猫都足够聪明且都希望获胜，小小猫（先手）想知道它是否有必胜策略。

Input

第一行一个正整数 $T(1 \leq T \leq 100)$ ，表示测试用例组数。

接下来 T 行，每行三个正整数 a, b, c ，分别表示 3 堆猫罐头的初始数量，用空格隔开。其中 $1 \leq a, b, c \leq 10^4$ 。

Output

共 T 行：对每组测试用例，输出一行一个字符串 “YES” 或 “NO”，表示小小猫（先手）是否有必胜策略（是输出 “YES”，否输出 “NO”）

Example

standard input	standard output
4	NO
1 1 1	YES
1 2 1	YES
1 2 3	YES
8 9 10	

Hints

第一组数据， $a = b = 1$ ，先手已经无法操作。

第二组和第三组数据，先手选择 b, c ，将 c 取走，将 $b = 2$ 分成 $b = 1$ 和 $c = 1$ ，则局面变成 $a = b = c = 1$ ，先手胜利。

Problem 7 猫咪军团

Time limit: 3 seconds

Memory limit: 512 Megabytes

【猫咪军团】 侵略狗星!!!

狗星有 n 个地区，有 $n - 1$ 条双向道路使狗星的所有地区连通。猫咪军团要投放一些猫咪在狗星的某些地区上以抓捕狗仔。猫咪们修建了 m 条猫咪通道。猫咪只能走猫咪通道，狗仔只能走狗星道路。

猫咪通道：一条猫咪通道连接两个不同的点 u 、 v ，并且猫咪通道上有一个中转站，猫咪想从 u 走到 v 或从 v 走到 u 时，需要先走到中转站，再从中转站走到另一个点。

当某只猫咪在某个中转站时，若狗星道路上也有一条边连接 u 和 v ，视为这只猫咪在狗星道路的这条路中间。

游戏规则：

- 猫咪军团先投放猫咪，狗仔再选择初始位置。猫咪和狗仔交替行动，由猫咪先手（事实上先后手不影响结果）。
- 猫咪行动：选择一只猫咪，然后让它从一个地区走到一个中转站上，或者从中转站走到一个地区上（只能走一次，经过“半条”猫咪通道）。
- 狗仔行动：狗仔可以通过狗星的道路移动到另一个地区（可以经过任意条道路和地区，只要路径中间及路径经过的地区没有猫咪），也可以原地不动。
- 任何时刻，只要有猫咪和狗仔在同一个地区，就视作抓捕成功。

在保证能抓捕住狗仔的情况下，最小化要投放的猫咪的数量。

补充说明 1：多只猫咪可以待在同一个地区，猫咪们和狗仔都知道彼此的位置。

补充说明 2：猫咪通道不一定将狗星的所有地区连通。

Input

第一行一个正整数 $T(1 \leq T \leq 100)$ ，表示测试用例组数。对每组测试用例：

第一行两个正整数 $n, m(1 \leq n \leq 10^5, 0 \leq m \leq 10^5)$ ，分别表示地区数量（点数）和猫咪通道的数量。

接下来 $n - 1$ 行，每行两个正整数 u, v ，表示狗星地图上的一条边。

接下来 m 行，每行两个正整数 u, v ，表示连接 u 和 v 的一条猫咪通道。

保证对所有测试用例， $\sum n \leq 5 \times 10^5, \sum m \leq 6.5 \times 10^5$ 。

Output

共 T 行，每行一个整数表示至少需要投放的猫咪数量。

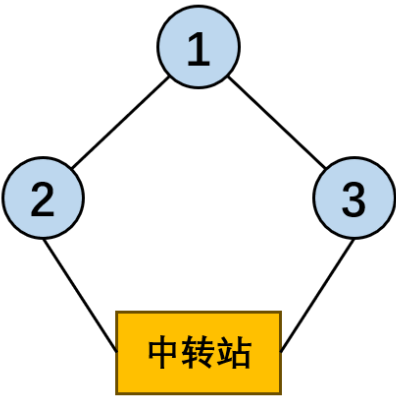
Example

standard input	standard output
3	3
3 0	2
2 1	1
2 3	
3 1	
1 2	
3 1	
2 3	
3 3	
1 2	
3 2	
2 1	
3 2	
1 3	

Hints

第一组数据，没有猫咪通道，所以每个地区都要投放 1 只猫咪。

第二组数据，地区 1 和 2，3 无法通过猫咪通道连通，地区 1 需要投放 1 只猫咪，地区 2 或 3 需要投放且只需要投放一只猫咪。



第三组数据，猫咪通道连成一个完全图，可以证明投放 1 只猫咪即可。

Problem 8 猫咪们狂欢

Time limit: 1 seconds

Memory limit: 256 Megabytes

猫咪们生活在树上。

具体来说，有 n 只猫咪和两棵大小为 n 的树⁴。猫咪编号为 $1 \sim n$ ，每棵树上的节点编号也为 $1 \sim n$ （编号各不相同）。

今晚，每只猫咪要分别选择一棵树，并待在与其编号相同的节点。

在这 n 只猫咪之中，有 k 只猫咪是狂欢猫。狂欢猫晚上不会睡觉，而是会选择开 party。其他猫咪则会选择睡觉。

每条树边都有一个狂欢值，如果这条边连接的两个节点在晚上都有狂欢猫待着，这个狂欢值就会被累加到总狂欢值上。

最大化今晚的总狂欢值，并输出这个值。

Input

第一行包含一个正整数 T ($1 \leq T \leq 50$)，表示测试用例组数。对于每一组测试用例：

- 第一行包含两个正整数 n 和 k ，分别表示猫咪的总数和狂欢猫的数量， $1 \leq k \leq n \leq 1000$ 。
- 接下来一行包含 k 个整数，分别表示狂欢猫的编号，保证这 k 个数互不相等。
- 接下来 $n - 1$ 行，每行包含三个正整数 u, v 和 w ，表示第一棵树的一条连接 u, v 且狂欢值为 w 的边。
- 接下来 $n - 1$ 行，每行包含三个正整数 u, v 和 w ，表示第二棵树的一条连接 u, v 且狂欢值为 w 的边。
- 上述 $w : 1 \leq w \leq 20$ 。
- 对所有测试用例， $\sum n \leq 10000$ 。

Output

共 T 行，每行一个整数分别表示对应测试用例的答案。

⁴由 n 个点， $n - 1$ 条边构成的连通图，树的大小在这里定义为其节点数。

Example

standard input	standard output
2	5
4 3	3
1 2 3	
1 2 1	
2 3 2	
2 4 1	
2 3 5	
2 4 1	
4 1 1	
4 3	
1 2 3	
1 2 2	
2 3 1	
2 4 1	
2 3 2	
2 4 1	
4 1 1	

Hints

- 第一组数据，狂欢猫 2,3 待在第二棵树上，得到狂欢值 5.
- 第二组数据，狂欢猫 1,2,3 待在第一棵树上，得到狂欢值 $2 + 1 = 3$.

Problem 9 用心感受（三）

Time limit: 1 seconds
Memory limit: 256 Megabytes

在去年杭电多校，Stump 用心感受出了 $\sum_{d|n} \mu(\frac{n}{d}) \ln(d)$ 的结果，又震惊了 Yoshinow 一整年。今年 Legend_dy 在复习期末考时，Cuking 掏出了下面的式子：

$$\sum_{k=1}^n \mu(k) \cdot (a^{\varphi(k)+1} \bmod k)$$

其中 μ 和 φ 分别表示莫比乌斯函数和欧拉函数。
可怜的 Legend_dy 快复习不完了，你能帮他用心感受一下吗？

Input

第一行一个正整数 $T(1 \leq T \leq 30)$ ，表示测试用例组数。
接下来 T 行，每行输入两个正整数 $n, a(1 \leq n, a \leq 10^9)$ ，用一个空格隔开。

Output

对每组测试用例，输出一行一个整数表示答案。

Example

standard input	standard output
3	-1
3 3	0
10 10	97
100 100	

Hints

莫比乌斯函数 μ ：若 n 中含有非平凡平方因子（即存在某个正整数 $a > 1, a^2|n$ ）则 $\mu(n) = 0$ ，否则不妨设 n 的唯一分解为 $n = \prod_{i=1}^k p_i$ ，则 $\mu(n) = (-1)^k$ 。
欧拉函数 φ ： $\varphi(n)$ 表示 $\{1, 2, \dots, n\}$ 中和 n 互质的数的个数。
对于第一组测试用例， $n = 3, a = 3$ ，答案为 $1 \times (3^1 \bmod 1) + (-1) \times (3^1 \bmod 2) + (-1) \times (3^2 \bmod 3) = -1$ 。

Problem 10 世末农庄

Time limit: 3 seconds

Memory limit: 256 Megabytes

西元???? 年，星球“泰拉”发生了核战争，整个地上世界成为了一片废土，由于辐射，异变生物已经占领了地上世界。

矮人族先西为了生存，不得不隐入地下进行生存。

幸好！先西的老祖宗早有远见，开发了地下农庄。地下农庄是一个家中地下室为根（记根的编号为 0）的树形结构，由于老祖宗没有留下地图，先西只能进行探索，以找寻农庄中每个农田的具体位置。

先西带着升降滑索和背包进入了农庄，并把锚点打在了家中的地下室。也就是说，它可以在回家时从当前所在位置到根的简单路径上获取农田中的农作物。

先西发现，不同农田的农作物所能提供的营养参差不齐，每单位农作物能得到的粮食不同。简单来说，某一层的粮食产出符合如下公式：

$$\text{粮食重量} = \text{该田农作物的营养值 } p \cdot \text{该田剩余农作物重量 } w$$

并且，越是到达深层，环境越是恶劣，农田的农作物营养值总是不如上一层直接相邻农田的营养值高。

此外，由于极少数异变蝗虫会钻地以获取食物，某些已经发现的农田可能因此被一扫而空。

为了简化问题，每天仅会发生下列三种事情之一：

- 发现新的农田，编号记为 u ，其与某一个已知的编号为 v 的农田相通（注意地下农庄是一个树形结构），该农田农作物的营养值为 p_u ，农作物重量为 w_u ，且由于深处环境更加恶劣，总是有 $p_u < p_v$ ；
- 回家整理行装，此时先西在编号为 k 的农田，打算沿着最短路径回到地下室（根），背包还能装 s 个单位重量的农作物，此时需要输出先西最多能得到多少粮食重量（先西总是优先保证这一天带走的农作物能够生产最多粮食），注意每次获取农作物后，农作物剩余量会减少；
- 异变蝗虫入侵，此时编号为 l 的农田中，农作物被一扫而空。

现在对于接下来 q 天，根据已知的地下农庄结构，背包剩余重量空间，每次回家时先西应当如何决策以获得最多的粮食回到地下室？

简单地说：

初始有一棵只有根节点的树，根编号为 0，根有两个点权 p_0 和 w_0 ；

在 q 个操作中，每个操作是以下三种情况之一：

- 操作 1：新增一个未出现过的节点 u ，令节点 u 与某已经存在的节点 v 相连，有两个点权 p_u 和 w_u ，保证 $p_u < p_v$ ；

- 操作 2: 进行一次询问, 给定节点 k 和背包容量 s , 查询从节点 k 到根的路径点集 X 中, 给每个点 $v \in X$ 分配一个非负整数 m_v , 在约束 $(\sum m_v \leq s) \wedge (m_v \leq w_v), v \in X$ 下, 得到 $Ans = \max \sum_{v \in X} (m_v \times p_v)$, 输出 $\sum m_v$ 和 Ans , 注意每次询问过后, 需要进行更新 $w_v = w_v - m_v$;
- 操作 3: 给定一个节点 l , 将 w_l 置为 0.

Input

第一行输入一个正整数 $T(1 \leq T \leq 10)$, 表示有 T 组测试用例。对每组测试用例:

- 第一行, 给出三个整数 q, p_0, w_0 , 以空格隔开, 分别表示总天数, 根节点农田农作物营养值, 农作物重量。
- 接下来 q 行, 首先给出一个整数 opt , 用于表示三种事情的编号, 紧接一个空格, 然后根据 opt 的不同:
 - 若 $opt = 1$, 则表示发现新的农田, 接下来给出整数 u, v, p_u, w_u , 同样以空格隔开, 保证编号为 u 的农田未被发现, 而为 v 的已被发现, 且 $p_u < p_v$;
 - 若 $opt = 2$, 则表示回家整理行装, 接下来给出整数 k, s , 同样以空格隔开, 保证编号 k 的农田已经被发现, 此时需要进行输出, 具体见“输出格式”节;
 - 若 $opt = 3$, 则表示异变蝗虫入侵, 接下来给出整数 l , 同样保证编号 l 的农田已经被发现。

对所有测试用例, $1 \leq q \leq 2 \times 10^5$, 且 $1 \leq u \leq 2 \times 10^5, 0 \leq v, k, l \leq 2 \times 10^5, 0 \leq p_i, w_i, s \leq 10^6, i \in \mathbb{N}$ 。

Output

对每组测试用例: 若输入 opt 为 2, 则需要输出一行两个整数 x, y (用空格隔开), 分别表示背包中装入的农作物重量, 以及能得到的粮食总量。

Example

standard input	standard output
1	3 18
5 6 4	3 12
2 0 3	2 2
1 2 0 3 2	
2 2 4	
1 4 0 1 3	
2 4 2	

Hints

在样例中，一共有 $q = 5$ 天，初始根节点 $p_0 = 6, w_0 = 4$.

- 第一天，用大小为 $s = 3$ 的背包容量，拿走了 0 号节点 3 个单位重量的农作物， $w_0 \rightarrow 1$ ，得到粮食总量是 $3 \times 6 = 18$.
- 第二天，添加了一个 2 号节点，连向 0 号点， $p_2 = 3 < p_0, w_2 = 2$.
- 第三天，用大小为 $s = 4$ 的背包，在 2 号点询问，拿走 0 号点 1 个单位重量的农作物，以及 2 号点 2 个单位重量的农作物，得到的总重量为 $1 + 2 = 3$ ，粮食总量是 $1 \times 6 + 2 \times 3 = 12$.
- ...

Problem 11 开关灯

Time limit: 1 seconds
Memory limit: 256 Megabytes

Yoshinow 有一排灯，按 1 到 n 的顺序从左到右编号，开始时所有灯全是关闭的。每次操作可以选择一盏灯，同时反转这盏灯及其左右两盏灯（如果存在）的开/关状态（即关变开、开变关）现在可以对这些灯做任意多次（可以是零次）操作，Yoshinow 想问你，这排灯共有多少种不同状态是可以到达的，由于答案可能很大，需要输出其对 998 244 353 取模的结果。称两种状态不同，当且仅当两个状态中，存在至少一盏灯的开/关状态不同。

Input

第一行输入一个正整数 T ，表示一共有 T 组测试用例， $1 \leq T \leq 100$ 。
接下来 T 行，每行输入一个整数 n ，即题目描述的电灯数量， $1 \leq n \leq 10^9$ 。

Output

对每个测试用例，输出一行一个整数表示答案。

Example

standard input	standard output
2	8
3	16
4	

Hints

$n = 3$ 时每种状态的一种可能操作方式如下（其中 r_i 表示在编号为 i 的灯上操作）：
(0, 0, 0)
(0, 0, 0) $\xrightarrow{r_1}$ (1, 1, 0) $\xrightarrow{r_2}$ (0, 0, 1)
(0, 0, 0) $\xrightarrow{r_1}$ (1, 1, 0) $\xrightarrow{r_2}$ (0, 0, 1) $\xrightarrow{r_3}$ (0, 1, 0)
(0, 0, 0) $\xrightarrow{r_3}$ (0, 1, 1)
(0, 0, 0) $\xrightarrow{r_2}$ (1, 1, 1) $\xrightarrow{r_3}$ (1, 0, 0)
(0, 0, 0) $\xrightarrow{r_1}$ (1, 1, 0) $\xrightarrow{r_3}$ (1, 0, 1)
(0, 0, 0) $\xrightarrow{r_1}$ (1, 1, 0)
(0, 0, 0) $\xrightarrow{r_2}$ (1, 1, 1)

Problem 12 串串

Time limit: 2 seconds
Memory limit: 256 Megabytes

给定一个长度为 n 的字符串 S 。现在有一个字符串 T ，初始时 $T = S$ （在接下来的操作中 S 不变）。

定义对 T 的删除操作：每次操作删除 T 开头或结尾恰好一个字符，同时花费『（操作前） T 在 S 中的出现次数』的代价。

现在希望通过 n 次操作将 T 变为空串，求最小的总花费

Input

第一行一个整数 $T(1 \leq T \leq 5 \times 10^4)$ 表示测试用例数量。

对每个测试用例，输入一个仅包含小写字母的字符串 $S(1 \leq \sum |S| \leq 10^6)$ 。

Output

对每个测试用例，输出一行一个整数，表示最小总花费。

Example

standard input	standard output
5	3
abc	4
aaba	6
aaabb	7
legendy	9
ygfuygfu	

Hints

例如对于 $S = T = aaabb$ ，一种可能的操作方式如下： $\underline{a}aabb \xrightarrow{1} \underline{a}abb \xrightarrow{1} \underline{a}bb \xrightarrow{1} \underline{b}b \xrightarrow{1} \underline{b} \xrightarrow{2} \epsilon$ 。（ ϵ 表示空串， \rightarrow 上的数字表示花费）

Problem 13 飞行棋

Time limit: 1 seconds
Memory limit: 256 Megabytes

今晚飞行棋大师 *Legend_dy* 不是很想学习于是邀请 *Jeneveuxpas* 和 *LZVSDY* 一起来玩把飞行棋，赛前放下豪言势必要血虐他们。没想到成了小丑，四连胜就此终结。



Jeneveuxpas 三辆飞机抵达终点，剩下的一辆飞机也已率先进入直道，本以为自己已经胜券在握了，没想到连摇了十来次骰子都没能摇进终点，让 *LZVSDY* 后来居上夺走了胜利的果实，他非常难过百思不得其解提出了这样一个问题：

有 $n + 1$ 个的格子编号 0 到 n ， 0 号格子为起点， n 号格子为终点。
花费 1 的代价等概率随机 $[1, n]$ 中的一个整数 x ，向终点走 x 步，若达到终点还有剩余步数则反向行走。如果随机到了数字 n 并且未抵达终点则 赠送一次等概率随机 $[1, n - 1]$ 中整数 y 的机会，并向终点走 y 步。问期望花费多少代价能恰好抵达终点？

Input

第一行输入一个正整数 T ，表示一共有 T 组测试用例， $1 \leq T \leq 100$ 。
接下来 T 行，每行输入一个整数 n ，即题目描述的 n ， $6 \leq n \leq 10^9$ 。

Output

一行一个整数，表示期望花费多少代价对 $10^9 + 7$ 取模的结果.

Example

standard input	standard output
1 998244353	3168436