

全国青少年信息学奥林匹克竞赛

CCF NOI 2018

第二试

时间：2018 年 7 月 20 日 08:00 ~ 13:00

题目名称	屠龙勇士	情报中心	多边形
题目类型	传统型	传统型	传统型
目录	dragon	center	polygon
可执行文件名	dragon	center	polygon
输入文件名	dragon.in	center.in	polygon.in
输出文件名	dragon.out	center.out	polygon.out
每个测试点时限	2.0 秒	8.0 秒	10.0 秒
内存限制	512 MB	512 MB	512 MB
测试点/包数目	20	20	20
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	dragon.cpp	center.cpp	polygon.cpp
对于 C 语言	dragon.c	center.c	polygon.c
对于 Pascal 语言	dragon.pas	center.pas	polygon.pas

编译选项

对于 C++ 语言	-O2 -lm
对于 C 语言	-O2 -lm
对于 Pascal 语言	-O2

注意事项：

- 1、提交的源文件必须存放在已建立好的下发样例的文件夹中（该文件夹与试题同名）。
- 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。

屠龙勇士 (dragon)

【题目描述】

小 D 最近在网上发现了一款小游戏。游戏的规则如下：

- 游戏的目标是按照编号 $1\sim n$ 顺序杀掉 n 条巨龙，每条巨龙拥有一个初始的生命值 a_i 。同时每条巨龙拥有恢复能力，当其使用恢复能力时，它的生命值就会每次增加 p_i ，直至生命值非负。只有在攻击结束后且当生命值恰好为 0 时它才会死去。
- 游戏开始时玩家拥有 m 把攻击力已知的剑，每次面对巨龙时，玩家只能选择一把剑，当杀死巨龙后这把剑就会消失，但作为奖励，玩家会获得全新的一把剑。

小 D 觉得这款游戏十分无聊，但最快通关的玩家可以获得 ION2018 的参赛资格，于是小 D 决定写一个笨笨的机器人帮她通关这款游戏，她写的机器人遵循以下规则：

- 每次面对巨龙时，机器人会选择当前拥有的，攻击力不高于巨龙初始生命值中攻击力最大的一把剑作为武器。如果没有这样的剑，则选择攻击力最低的一把剑作为武器。
- 机器人面对每条巨龙，它都会使用上一步中选择的剑攻击巨龙固定的 x 次，使巨龙的生命值减少 $x \times ATK$ 。
- 之后，巨龙会不断使用恢复能力，每次恢复 p_i 生命值。若在使用恢复能力前或某一次恢复后其生命值为 0，则巨龙死亡，玩家通过本关。

那么显然机器人的攻击次数是决定能否最快通关这款游戏的关键。小 D 现在得知了每条巨龙的所有属性，她想考考你，你知道应该将机器人的攻击次数 x 设置为多少，才能用最少的攻击次数通关游戏吗？

当然如果无论设置成多少都无法通关游戏，输出 -1 即可。

【输入格式】

从文件 *dragon.in* 中读入数据。

第一行一个整数 T ，代表数据组数。

接下来 T 组数据，每组数据包含 5 行。

- 每组数据的第一行包含两个整数， n 和 m ，代表巨龙的数量和初始剑的数量；
- 接下来一行包含 n 个正整数，第 i 个数表示第 i 条巨龙的初始生命值 a_i ；
- 接下来一行包含 n 个正整数，第 i 个数表示第 i 条巨龙的恢复能力 p_i ；
- 接下来一行包含 n 个正整数，第 i 个数表示杀死第 i 条巨龙后奖励的剑的攻击力；
- 接下来一行包含 m 个正整数，表示初始拥有的 m 把剑的攻击力。

【输出格式】

输出到文件 *dragon.out* 中。

一共 T 行。

第 i 行一个整数，表示对于第 i 组数据，能够使得机器人通关游戏的最小攻击次数 x ，如果答案不存在，输出 -1。

【样例 1 输入】

```
2
3 3
3 5 7
4 6 10
7 3 9
1 9 1000
3 2
3 5 6
4 8 7
1 1 1
1 1
```

【样例 1 输出】

```
59
-1
```

【样例 1 解释】

第一组数据：

- 开始时拥有的剑的攻击力为 $\{1,9,10\}$ ，第 1 条龙生命值为 3，故选择攻击力为 1 的剑，攻击 59 次，造成 59 点伤害，此时龙的生命值为 -56，恢复 14 次后生命值恰好为 0，死亡。
- 攻击力为 1 的剑消失，拾取一把攻击力为 7 的剑，此时拥有的剑的攻击力为 $\{7,9,10\}$ ，第 2 条龙生命值为 5，故选择攻击力为 7 的剑，攻击 59 次，造成 413 点伤害，此时龙的生命值为 -408，恢复 68 次后生命值恰好为 0，死亡。
- 此时拥有的剑的攻击力为 $\{3,9,10\}$ ，第 3 条龙生命值为 7，故选择攻击力为 3 的剑，攻击 59 次，造成 177 点伤害，此时龙的生命值为 -170，恢复 17 次后生命值恰好为 0，死亡。
- 没有比 59 次更少的通关方法，故答案为 59。

第二组数据:

- 不存在既能杀死第一条龙又能杀死第二条龙的方法, 故无法通关, 输出 -1。

【样例 2】

见选手目录下的 *dragon/dragon2.in* 与 *dragon/dragon2.ans*。

【子任务】

测试点编号	n	m	p_i	a_i	攻击力	其他限制
1	$\leq 10^5$	$= 1$	$= 1$	$\leq 10^5$	$= 1$	无
2						
3						
4						
5	$\leq 10^3$	$\leq 10^3$	$\leq 10^5$	$\leq 10^5$	$\leq 10^5$	特性 1、特性 2
6						
7						
8	$= 1$	$= 1$	$\leq 10^8$	$\leq 10^8$	$\leq 10^6$	特性 1
9						
10						
11						
12						
13						
14	$= 10^5$	$= 10^5$	$= 1$	$\leq 10^{12}$	$\leq 10^6$	无特殊限制
15						
16	$\leq 10^5$	$\leq 10^5$	所有 p_i 是质数	$\leq 10^{12}$	$\leq 10^6$	特性 1
17						
18			无特殊限制			
19						
20						

特性 1 是指: 对于任意的 i , $a_i \leq p_i$ 。

特性 2 是指: $LCM(p_i) \leq 10^6$ 即所有 p_i 的最小公倍数不大于 10^6 。

对于所有的测试点, $T \leq 5$, 所有武器的攻击力 $\leq 10^6$, 所有 p_i 的最小公倍数 $\leq 10^{12}$ 。

【提示】

你所用到的中间结果可能很大, 注意保存中间结果的变量类型。

情报中心 (center)

【题目描述】

C 国和 D 国近年来战火纷飞。

最近, C 国成功地渗透进入了 D 国的一个城市。这个城市可以抽象成一张有 n 个节点, 节点之间由 $n-1$ 条双向的边连接的无向图, 使得任意两个点之间可以互相到达, 也就是说这张无向图实际上是一棵树。

经过侦查, C 国情报部部长 GGB 惊讶地发现, 这座看起来不起眼的城市竟然是 D 国的军事中心。因此 GGB 决定在这个城市内设立情报机构。情报专家 TAC 在侦查后, 安排了 m 种设立情报机构的方案。这些方案中, 第 i 种方案是在节点 x_i 到节点 y_i 的最短路径的所有边上安排情报人员收集情报, 这种方案需要花费 v_i 元的代价。

但是, 由于人手不足, GGB 只能安排上述 m 种方案中的两种进行实施。同时 TAC 指出, 为了让这两个情报机构可以更好的合作, 它们收集情报的范围应至少有一条公共的边。为了评估一种方案的性能, GGB 和 TAC 对所有的边进行了勘察, 给每一条边制定了一个情报价值 c_i , 表示收集这条边上的情报能够带来 c_i 元的收益。注意, 情报是唯一的, 因此当一条边的情报被两个情报机构收集时, 也同样只会有 c_i 的收益。

现在, 请你帮 GGB 选出两种合法的设立情报机构的方案进行实施, 使得这两种方案收集情报的范围至少有一条公共的边, 并且在此基础上总收益减去总代价的差最大。注意, 这个值可能是负的, 但仍然是合法的。如果无法找到这样的两种方案, 请输出 F。

【输入格式】

从文件 *center.in* 中读入数据。

本题包含多组测试数据。

输入文件的第一行包含一个整数 T , 表示数据组数;

每组数据包含 $(n+m+1)$ 行:

第 1 行包含一个整数 n , 表示城市的点数;

第 2 到第 n 行中, 第 $(i+1)$ 行包含三个整数 a_i, b_i, c_i , 表示城市中一条连接节点 a_i 和 b_i 、情报价值为 c_i 的双向边, 保证 $a_i < b_i$ 且 b_i 互不相同;

第 $(n+1)$ 行包含一个整数 m , 表示 TAC 设立的 m 种设立情报机构的方案;

第 $(n+2)$ 到 $(n+m+1)$ 行中, 第 $(n+i+1)$ 行包含三个整数 x_i, y_i, v_i , 表示第 i 种设立情报机构的方案是在节点 x_i 到节点 y_i 的最短路径上的所有边上安排情报人员收集情报, 并且需要花费 v_i 元的代价。

【输出格式】

输出到文件 *center.out* 中。

输出文件包含 T 行;

对于每组数据, 输出一行: 如果存在合法的方案, 则输出一个整数表示最大的总收益减去总代价的差; 否则输出 F 。

【样例 1 输入】

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

【样例 1 输出】

```
1
F
```

【样例 1 解释】

这个样例中包含两组数据。这两组数据的城市相同, 只是在情报的价值和情报机构的方案上有所不同。城市地图如下:



- 对于第一组数据, 方案一中的节点 1 到节点 4 的最短路径为 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$, 方案二中的节点 3 到节点 5 的最短路径为 $3 \rightarrow 2 \rightarrow 1 \rightarrow 5$ 。选择这两种方案

费 $5 + 8 = 13$ 的代价，并且每一条边的情报都被收集从而得到 $1 + 3 + 2 + 8 = 14$ 的收益，因此总收益减去总代价为 $14 - 13 = 1$ 。

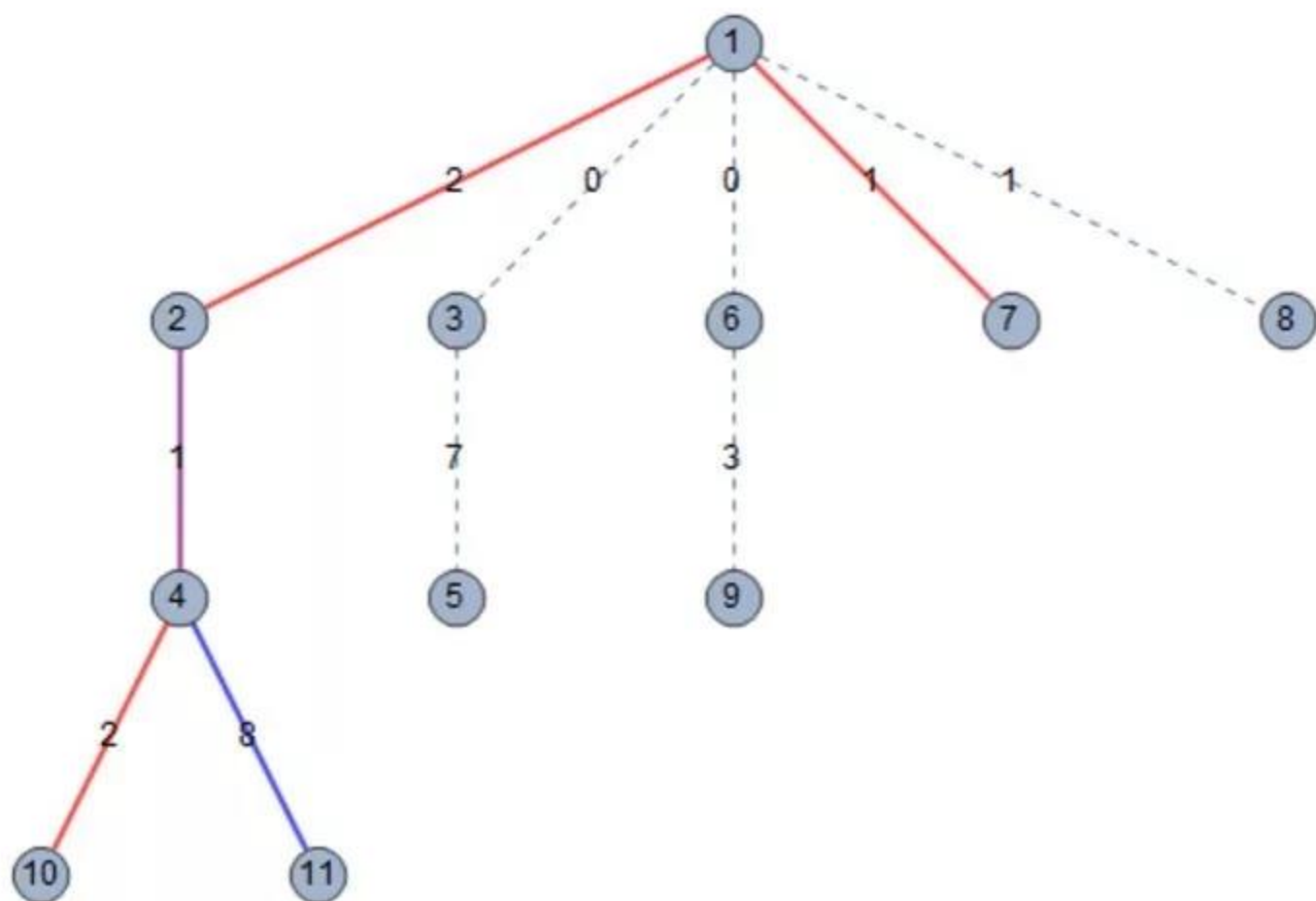
- 对于第二组数据，方案一中的节点 1 到节点 5 的最短路径为 $1 \rightarrow 5$ ，方案二中的节点 2 到节点 3 的最短路径为 $2 \rightarrow 3$ 。这两种方案收集情报的范围没有公共的边，因此非法，所以这组数据不存在合法方案，应输出 F。

【样例 2】

见选手目录下的 *center/center2.in* 与 *center/center2.ans*。

这个样例只包含一组数据。这一数据中，最优方案为选择第 2 种和第 3 种方案。

这组数据的城市地图如下，其中加粗的边表示被情报中心收集情报的边，红色的边表示只被第 2 种方案的情报中心收集情报的边，蓝色的边表示只被第 3 种方案的情报中心收集情报的边，紫色的边表示同时被两个情报中心收集情报的边。



【样例 3】

见选手目录下的 *center/center3.in* 与 *center/center3.ans*。

这个样例和第 4 个测试点的性质相同。每个测试点的性质见下文的表格。

【样例 4】

见选手目录下的 *center/center4.in* 与 *center/center4.ans*。

这个样例包含了经过特殊构造的 $n \leq 100, m \leq 200$ 的测试数据，涵盖了测试点中所有出现性质的组合。你可以合理利用这个测试点，对自己的程序进行全面的检查。

【子任务】

各测试点的数据规模和性质如下表：

测试点	$n \leq$	$m \leq$	$T \leq 50$	特殊性质	
1	2	3	保证	无	
2	10	30			
3	200	300			
4	10^3	2,000		$a_i = b_i - 1$	
5	10^4	3×10^4			
6	5×10^4	10^5			
7	10^4	3×10^4			
8	5×10^4	10^5		不保证	$c_i = 0$
9					
10	10^4	n		不保证	S_1
11	5×10^4				
12					
13	10^4	3×10^4	保证	S_2	
14					
15	5×10^4	10^5	不保证		
16					
17	10^4	3×10^4	保证	无	
18	5×10^4	10^5	不保证		
19					
20					

表格中的特殊性质如下：

- 特殊性质 S_1 ：对于任意 i, j ，保证 x_i 到 y_i 的最短路径所经过的编号最小的节点不同于 x_j 到 y_j 的最短路径所经过的编号最小的节点；
- 特殊性质 S_2 ：对于任意 i ，保证 x_i 到 y_i 的最短路径所经过的编号最小的节点为节点 1。

对于所有的数据， $1 \leq n \leq 5 \times 10^4$ ， $0 \leq m \leq 10^5$ ， $0 \leq c_i \leq 10^9$ ， $0 \leq v_i \leq 10^{10} \times n$ 。每个测试点中，所有 n 的和不会超过 1,000,233，所有 m 的和不会超过 2,000,233。

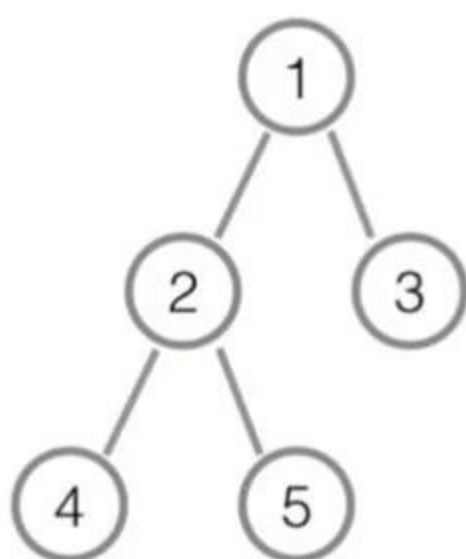
多边形 (polygon)

【题目描述】

久莲是一个喜欢出题的女孩子。

在今年的 World Final 结束以后，久莲特别喜欢计算几何，于是她打算在 NOI 的考场上也出一个计算几何：这是一道只有题目名字和计算几何相关的题目。

首先，久莲给出了一棵 $n(n \geq 2)$ 个节点的有根树 T ，根节点编号为 1。定义叶子节点为除了根以外所有度数恰好为 1 的节点。下图是一个树 T 的例子，其中叶子节点集合为 $\{3, 4, 5\}$ 。



接着通过这棵树，久莲构造了一个序列 A ：

- 从根节点开始深度优先遍历整棵树，遍历时按照编号从小到大的顺序来访问孩子，这样可以得到一个树 T 的 DFS 序。
- 接着按照在 DFS 序中的出现顺序从前往后，久莲把所有叶子节点排成一排得到了一个序列 A 。

更进一步地，通过序列 A ，久莲定义了两个叶子节点 u, v 的距离 $d(u, v)$ ：假设 u 在 A 中是第 i 个元素， v 是第 j 个元素，则 $d(u, v) = \min(|i - j|, |A| - |i - j|)$ 。其中 $|A|$ 为序列的长度，即 T 的叶子个数， i, j 指的是出现的位置，从 1 开始计数。

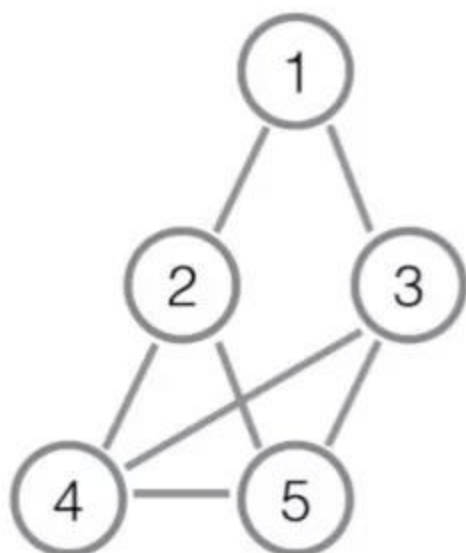
上面的例子中，序列 A 为 $[4, 5, 3]$ ，其中 $d(3, 5) = d(3, 4) = d(4, 5) = 1$ ， $3, 4, 5$ 的出现位置分别为 3, 1, 2。

最后，久莲给出了一个参数 K ，利用这棵有根树 T 和序列 A ，我们可以构造一张 n 个点的无重边无自环的无向图 G ：两个不同的点 u, v 之间有边当且仅当它们满足下列条件中的至少一个：

- 在树 T 中存在连接 u, v 的边。
- 在树 T 中 u, v 都是叶子节点且 $d(u, v) \leq K$ 。

当 $K = 1$ 或 2 时，上面的例子得到的图 G 都如下图所示：

现在久莲想让你来计算一下 G 中不同的哈密尔顿回路数量有多少条。答案可能很大，请对 998244353 取模后输出。



下面是一些补充定义：

- 无重边无自环的无向图 G 的一条哈密顿回路 H 是 G 中边的一个子集，其中每一个点恰好有两条不同的相邻边在 H 中，且任意两个点都可以通过 H 中的边直接或间接到达。
- 无重边无自环的无向图 G 的两条哈密顿回路 H_1, H_2 是不同的当且仅当存在一条边 e 使得 e 在 H_1 中且不在 H_2 中。

【输入格式】

从文件 *polygon.in* 中读入数据。

第一行输入两个整数 n, K ，表示树 T 的点数以及久莲选定的参数 K 。

第二行输入 $n-1$ 个整数 $f_i (1 \leq f_i \leq i)$ ，其中 f_i 表示树 T 上存在边 $(f_i, i+1)$ 。

【输出格式】

输出到文件 *polygon.out* 中。

输出一行一个整数，表示哈密顿回路数量对 998244353 取模后的结果。

【样例 1 输入】

```
5 1
1 1 2 2
```

【样例 1 输出】

```
2
```