

NOIP 模拟赛

题目名称	Fat_Tiger的迷宫	Fat_Tiger与怪兽	Fat_Tiger的王国
目录	maze	monster	kingdom
可执行文件名	maze	monster	kingdom
输入文件名	maze.in	monster.in	kingdom.in
输出文件名	maze.out	monster.out	kingdom.out
每个测试点时限	1秒	2秒	2秒
测试点数目	10	10	10
每个测试点分值	10	10	10
结果比较方式	全文比较（过滤行末空格及文末回车）		
题目类型	传统	传统	传统
运行内存上限	512M	512M	512M

对于 Pascal 语言	maze.pas	monster.pas	kingdom.pas
对于 C 语言	maze.c	monster.c	kingdom.c
对于 C++ 语言	maze.cpp	monster.cpp	kingdom.cpp

注意：最终测试时，所有编译命令均不打开任何优化开关

Fat_Tiger 的迷宫

【问题描述】

作为 Fat 王国有史以来最 Tiger 的一位国王，Fat_Tiger 有很多敌人。为了防止被潜在的敌人暗算，Fat_Tiger 准备建造一个由 N 个点组成的迷宫。

为了迷惑敌人，迷宫中的任意两个点都有连边。但是 Fat_Tiger 非常讨厌环，于是他希望这个迷宫里不要出现环，不然 Fat_Tiger 会很愤怒。为了让迷宫中不要出现环，Fat_Tiger 将所有边定成有向边，但是这样并不能保证迷宫中不出现环，于是 Fat_Tiger 想要知道有多少种给每条边确定一个方向的方案使迷宫中不出现环。

现在 Fat_Tiger 要求你帮他设计迷宫，并且希望知道有多少种方案使迷宫中不出现环。

你当然不想也不敢让 Fat_Tiger 愤怒，所以你必须及时计算出有多少种建造的方案使迷宫中不出现环。

【输入格式】

输入文件 `maze.in`。

第一行一个正整数 T ，表示输入包含 T 组测试数据，对于每组测试数据，第一行一个正整数 N ，表示迷宫的节点数。

【输出格式】

输出文件 `maze.out`。

输出包含 T 行，第 i 行表示第 i 组测试数据的方案数。由于方案数可能较大，所以请输出对 1000000007 取模的结果。

【样例】

样例输入	样例输出
2	2
2	6
3	

【数据规模和约定】

对于 40% 的测试数据， $T \leq 5$ ， $N \leq 7$

对于 60% 的测试数据， $T \leq 10$ ， $N \leq 100000$

对于 80% 的测试数据， $T \leq 100$ ， $N \leq 10000000$

对于 100% 的测试数据， $T \leq 100$ ， $N \leq 1000000000$

Fat_Tiger 与怪兽

【题目描述】

Fat_Tiger 遇上了 N 只怪兽, N 只怪兽排成了一排, 从左到右编号依次为 $0 \sim N - 1$ 。

作为 Fat 王国的国王, Fat_Tiger 当然不会孤立无援, 他叫来了他的手下 Fat_Fat_Tiger, Fat_Fat_Fat_Tiger 以及

有了这么多帮手, Fat_Tiger 于是向怪兽们发动了 M 次攻击。

已知第 i 只怪兽的初始血量为 H_i , 当怪兽的血量小于等于 0 时, 这只怪兽就挂了。在这 M 次攻击中, Fat_Tiger 每次都会选择两个整数 X_i 和 Y_i , 并对下标 k 满足 $k \& X_i = k$ 的怪兽发动攻击 (此处 $\&$ 代表按位与操作), 被攻击的怪兽会掉 Y_i 点血。

作为战斗的指挥者, Fat_Tiger 需要时刻了解战场上的情况, 于是在每次攻击完成之后, 你都需要告诉他还有多少怪兽还活着。

【输入格式】

输入文件 monster.in。

第一行包含一个整数 N , 代表怪兽的数量。

第二行包含 N 个整数 H_0, H_1, \dots, H_{N-1} , 代表怪兽的初始血量。

第三行包含一个整数 M , 代表 Fat_Tiger 发动攻击的次数。

接下来 M 行, 第 i 行包含两个整数 X_i 和 Y_i , 代表 Fat_Tiger 在第 i 次攻击中对下标 k 满足 $k \& X_i = k$ 的怪兽发动攻击, 被攻击的怪兽掉了 Y_i 点血。

【输出格式】

输出文件 `monster.out`。

输出包含 M 行，第 i 行表示第 i 次攻击完成后还活着的怪兽数量。

【样例】

样例输入	样例输出
5	4
1 2 3 4 5	4
5	2
1 1	2
2 2	1
3 3	
4 4	
5 5	

【数据规模和约定】

测试点	N	M	约定
1	≤ 1000	≤ 1000	无
2	≤ 1000	≤ 1000	无
3	≤ 1000	≤ 1000	无
4	≤ 15000	≤ 100000	无
5	≤ 15000	≤ 100000	无
6	≤ 15000	≤ 100000	无
7	≤ 100000	≤ 100000	$H_i = Y_i = 1$
8	≤ 100000	≤ 100000	$H_i = Y_i = 1$
9	≤ 100000	≤ 100000	无
10	≤ 100000	≤ 100000	无

对于所有测试点均满足：

$1 \leq N, M \leq 100000, 1 \leq H_i, X_i, Y_i \leq 1000000000$

Fat_Tiger 的王国

【题目描述】

在 Fat_Tiger 的 Fat 王国里，有 N 座城镇依次编号为 $1 \sim N$ 。这 N 座城镇由 N 条道路连接，道路是双向的。其中第 i 条道路连接 x_i 和 y_i ，长度均为 1。

王国初始情况是连通的。由于有 N 条道路，不免出现了环，而 Fat_Tiger 是讨厌环的。

于是 Fat_Tiger 希望通过删边使王国中不出现环，但是他必须保证删边后王国仍保持连通，以保持王国的稳定。

在所有的删边方案中，Fat_Tiger 希望找到一种方案，使王国的不方便度最小。

用 $\text{dist}(i, j)$ 表示 i 和 j 之间的最短路径，定义一种方案的不方便度为 $\sum_{i=1}^n \sum_{j=i+1}^n \text{dist}(i, j)$ 。

现在 Fat_Tiger 希望你告诉他最优方案的不方便度。

【输入格式】

输入文件 `kingdom.in`。

第一行包含一个整数 N ，代表王国中的城镇数。

接下来 $N - 1$ 行，第 $i + 1$ 行表示城镇 x_i 与城镇 y_i 有一条双向路径。

【输出格式】

输出文件 `kingdom.out`。

输出一行，包含一个整数表示最小的不方便度。

【样例】

样例输入	样例输出	样例解释
3 1 2 2 3 1 3	4	删去第 1 条道路，不方便度为 $2 + 1 + 1 = 4$ 。
5 2 3 3 1 4 1 3 5 4 5	18	删去第 3 条道路，不方便度为 $1 + 2 + 2 + 3 + 1 + 1 + 2 + 1 + 2 + 3 = 18$ 。 同时可以发现不存在更小的不方便度。

【数据规模和约定】

测试点	N	约定
1	≤ 100	无
2	≤ 100	无
3	≤ 5000	无
4	≤ 5000	无
5	≤ 200000	环的大小不超过 30
6	≤ 200000	环的大小不超过 30
7	≤ 200000	无
8	≤ 200000	无
9	≤ 1000000	无
10	≤ 1000000	无

对于所有测试点均满足：

$1 \leq N \leq 100000$ ， $1 \leq x_i \leq N$ ， $1 \leq y_i \leq N$ ，

保证王国的各城镇初始连通。