

Day 2

题目名称	坏蛋信使	路由器	进化	青蛙
可执行文件名	poster	router	up	frog
输入文件名	poster.in	router.in	up.in	frog.in
**输出文件名	poster.out	router.out	up.out	frog.out
每个测试点时限	1 秒	1 秒	1 秒	2 秒
内存限制	256MB	256MB	256MB	256MB
测试点数目	20	20	20	20
每个测试点分值	5	5	5	5
是否有部分分	否	否	否	否
题目类型	传统型	传统型	传统型	传统型
是否有样例文件	否	否	否	否
是否有附加文件	否	否	否	否

提交源程序须加后缀

对于 C++ 语言	poster.cpp	drink.cpp	up.cpp	frog.cpp
-----------	------------	-----------	--------	----------

编译开关

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

坏蛋信使(Poster)

【问题描述】

小 N 曾经当过坏蛋信使。有一天，小 N 需要送 n 封信，他小心翼翼、不留痕迹地把每封信从信封中拆出偷看。但这样还不过瘾！小 N 每次会在所有信中选择两个（可能相同），交换其信封，若选到两个相同的信则相当于没有交换。小 N 想知道，这样做 m 次后，期望有多少封信不在原来的信封中，结果对998244353取余数。

坏蛋出题人觉得这个问题太简单了，给题目加了多测。

【输入格式】

输入文件为 *poster.in*。

输入文件的第一行包含一个正整数 T ，表示测试数据组数。

接下来 T 行，每行包含用空格隔开的两个整数 n 和 m ，表示信的个数和交换次数。

【输出格式】

输出文件为 *poster.out*。

输出共 T 行，每行一个整数，表示期望有多少封信在原来的信封中，结果对998244353取余数。

【样例 1 输入】

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

【样例 1 输出】

```
0
0
1
```

【子任务】

所有测试数据的范围和特点如下：

对于40%的数据有 $1 \leq T \leq 10, 1 \leq n \leq 5, 1 \leq m \leq 5$

对于70%的数据有 $1 \leq n \leq 1000, 1 \leq m \leq 1000$

对于100%的数据有 $1 \leq T \leq 10^5, 1 \leq n \leq 10^{18}, 1 \leq m \leq 10^{18}$

路由器(Router)

【问题描述】

小 N 曾经是一名顶级黑客，他曾经利用病毒，瘫痪了整个 D 市的网络。

善良的出题人打算恢复 D 市的网络。据了解，D 市的网络是一个 n 个节点的树形结构。如果在其中一个节点上放上路由器，则这个点和他的所有邻居都将恢复网络。

但凡事都有代价，修网络亦是如此。在第 i 个点上放上路由器需要付出 a_i 。善良的出题人想要知道，最少需要付出多少才能恢复整个网络。

【输入格式】

输入文件为 *router.in*。

输入文件的第一行包含一个正整数 n 。

接下来一行包含 n 个整数 a_1, \dots, a_n ，表示各个节点上放上路由器所需要付出的代价。

接下来 $n - 1$ 行，每行包含两个正整数 u_i, v_i ，表示树上的一条边。

【输出格式与部分分】

输出文件为 *router.out*。

输出共一个整数，表示修复整个网络需要付出的最小代价

【样例 1 输入】

```
4
1 17 13 4
1 2
1 3
3 4
```

【样例 1 输出】

```
5
```

【样例 2 输入】

```
7
13 20 1 20 6 9 8
1 2
```

1 3
2 4
2 5
3 6
5 7

【样例 2 输出】

27

【子任务】

所有测试数据的范围和特点如下：

对于30%的数据有 $1 \leq n \leq 10$

对于50%的数据有 $1 \leq n \leq 100$

对于70%的数据有 $1 \leq n \leq 1000$

对于100%的数据有 $1 \leq n \leq 10^6, 1 \leq a_i \leq 10^9$

对于50%的数据有 树随机

进化(Up)

【问题背景】

就算是聪明的出题人也是从猩猩进化而来的。

一个物种拥有的基因可以简化地用两个数 (A, B) 表示（是否太过简化了？）每次进化，这个物种的基因会变为 $(A, A + B)$ 或者 $(A + B, B)$ 。

现在科学家已经知道了 n 个物种的基因 $(a_1, b_1), \dots, (a_n, b_n)$ 。他们新测出来了 q 个物种的基因 $(c_1, d_1), \dots, (c_q, d_q)$ ，聪明的出题人想要知道，对于每个新测出来基因的物种，他们可能是多少种已知物种的祖先。

一个物种可能是另一个物种的祖先，当且仅当这个物种的基因可以经过若干次进化得到另一个物种的基因（也可能 0 次进化）。

【输入格式】

输入文件为 *up.in*。

输入文件的第一行包含两个正整数 n, q ，用空格隔开，表示已知的基因的物种数和新测的物种数。

接下来 n 行，每行包含用空格隔开的两个整数 a_i 和 b_i 。

接下来 q 行，每行包含用空格隔开的两个整数 c_i 和 d_i 。

【输出格式】

输出文件为 *up.out*。

输出文件有 q 行，每行包含一个正整数，即新测物种可能是多少种已知物种的祖先。

【样例 1 输入】

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

【样例 1 输出】

1
0
1
1

【样例 2 输入】

2 2
7 14
7 14
7 7
7 14

【样例 2 输出】

2
2

【子任务】

所有测试数据的范围和特点如下：

对于20%的数据有 $1 \leq n \leq 10, 1 \leq q \leq 10$

对于40%的数据有 $1 \leq n \leq 100, 1 \leq q \leq 100$

对于70%的数据有 $1 \leq n \leq 1000, 1 \leq q \leq 1000$

对于100%的数据有 $1 \leq n \leq 5 \times 10^4, 1 \leq q \leq 5 \times 10^4, 1 \leq a_i, b_i, c_i, d_i \leq 10^{18}$

青蛙(Frog)

【问题背景】

倒霉的出题人在玩一款 VR 游戏的时候找不到退出按钮了。

在这款 VR 游戏中，出题人变成了一只五维空间中的青蛙，从河的一岸跳到另一岸。他本可以直接跳过去的，但是河中间有编号为 $1 \sim n$ 的 n 块石头，石头坐标为 $(a_i, b_i, c_i, d_i, e_i)$ ，每块石头上有奖励 f_i 。

由于河水湍急，如果青蛙跳上了石头，那么他只能往岸上，或者往编号比当前石头大，且五个坐标均不小于当前石头的石头上跳。

倒霉的出题人想知道，如果他最后从第 k 块石头往对岸跳，他最多能获得多少奖励。

【输入格式】

输入文件为 *frog.in*。

输入文件的第一行包含个正整数 n ，表示石头数量。

接下来 n 行，每行6个整数，其中包含 $a_i, b_i, c_i, d_i, e_i, f_i$ ，表示石头的坐标和奖励。

【输出格式】

输出文件为 *frog.out*。

输出文件有 n 行，每行一个整数，表示从第 i 块石头上岸所能获得最大获得的奖励。

【样例 1 输入】

```
4
1 2 1 2 1 30
2 1 2 1 2 40
3 3 3 3 3 50
2 3 3 2 4 100
```

【样例 1 输出】

```
30
40
90
```


140

【子任务】

所有测试数据的范围和特点如下：

对于30%的数据有 $1 \leq n \leq 10$

对于60%的数据有 $1 \leq n \leq 5000$

对于100%的数据有 $1 \leq n \leq 5 \times 10^4, 1 \leq a_i, b_i, c_i, d_i, e_i \leq n, 1 \leq f_i \leq 10^4$