

全国青少年信息学奥林匹克竞赛

NOIP2023模拟

时间：8:30-12:00

| 题目名称 | 秀秀的森林 | 签到题 | 简单题 | 不难题 |
|---------|------------|------------|----------|-------------|
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 目录 | forest | signin | easy | nothard |
| 可执行文件名 | forest | signin | easy | nothard |
| 输入文件名 | forest.in | signin.in | easy.in | nothard.in |
| 输出文件名 | forest.out | signin.out | easy.out | nothard.out |
| 每个测试点时限 | 1.0秒 | 1.0秒 | 3.0秒 | 1.0秒 |
| 内存限制 | 512 MB | 256MB | 256MB | 256MB |
| 子任务数目 | 10 | 20 | 50 | 50 |
| 测试点是否等分 | 是 | 是 | 是 | 是 |

提交源程序文件名

| | | | | |
|---------|----------|------------|----------|-------------|
| 对于C++语言 | bomb.cpp | signin.cpp | easy.cpp | nothard.cpp |
|---------|----------|------------|----------|-------------|

编译选项

| | |
|---------|--------------------|
| 对于C++语言 | -lm -std=c++14 -O2 |
|---------|--------------------|

注意事项与提醒（请选手务必仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C++ 中主函数的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参照各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 程序可使用的栈内存空间限制与题目的内存限制一致。
7. 全国统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz，内存 32GB。上述时限以此配置为准。
8. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。
9. 终评测时所用的编译命令中不含编译选项之外的任何优化开关。

秀秀的森林 (forest)

【问题描述】

秀秀有一棵带 n 个顶点的树 T ，每个节点有一个点权 a_i 。

有一天，她想拥有两棵树，于是她从 T 删去了一条边。

第二天，她认为三棵树或许会更好一些。因此，她又从她拥有的某一棵树中删去了一条边。

如此往复，每一天秀秀都会删去一条尚未被删去的边，直到她得到由 n 棵只有一个点的树构成的森林。

秀秀定义一条简单路径（节点不重复出现的路径）的权值为路径上所有点的权值之和，一棵树的直径为树上权值最大的简单路径。秀秀认为树最重要的特征就是它的直径。所以她想请你算出任一时刻她拥有的所有树的直径的乘积。因为这个数可能很大，你只需输出这个数对 $10^9 + 7$ 取模之后的结果即可。

【输入格式】

从文件 *forest.in* 中读入数据。

输入的第一行包含一个整数 n ，表示树 T 顶点的数量。

下一行包含 n 个空格分隔的整数 a_i ，表示顶点的权值。

之后的 $n - 1$ 行中，每一行包含两个用空格分隔的整数 u_i 和 v_i ，表示节点 u_i 和 v_i 之间连有一条边，编号为 i 。

再之后 $n - 1$ 行中，每一行包含一个整数 k_j ，表示在第 j 天里会被删除的边的编号。

【输出格式】

输出文件到 *forest.out* 中。

共 n 行，在第 i 行，输出删除 $i - 1$ 条边之后，所有树直径的乘积对 $10^9 + 7$ 取模的结果。

【样例输入1】

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

【样例输出1】

```
6
9
6
```

【样例1解释】

初始时，树的直径为 6（由节点 2、1 和 3 构成的路径）。在第一天之后，得到了两棵直径都为 3 的树。第二天之后，得到了三棵直径分别为 1，2，3 的树，乘积为 6。

【样例2】

见下发文件中。

【数据范围及约定】

对于 40% 的数据： $n \leq 100$;

另有 20% 的数据： $n \leq 1000$;

另有 20% 的数据： $n \leq 10000$;

对于 100% 的数据： $n \leq 100000, a_i \leq 10000$ 。

签到题 (signin)

【问题描述】

小 P 做了一个梦，梦到自己来到了 3202 年的 CSP-S 的现场，发现需要完成一个官方提出的问题才能完成签到，问题如下：

给定一棵 N 个点的无根树，需要你选定一个起点 S ，从这个点出发对树进行深度优先遍历，得到一个点的经过顺序（也就是 DFS 序）： $v_1 = S, v_2, \dots, v_N$ ，使得代价和 $\sum_{i=1}^N i * A_{v_i}$ 最小。

小 P 觉得这个签到题太简单了，随手就切掉了。但是他觉得这个题很有趣，所以希望你也来做一做。

【输入格式】

第一行一个整数 N ，表示树的节点数。

接下来 $N - 1$ 行，每行两个整数 u_i, v_i ，表示树上一条边。

接下来一行 N 个数，表示 A_i 。

【输出格式】

输出一行，表示最小的代价和。

【样例输入1】

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

【样例输出1】

```
11
```

【样例输入2】

```
5
1 2
1 3
3 4
3 5
5 4 3 2 1
```

【样例输出2】

```
35
```

【样例3】

见下发文件中。

【数据范围及约定】

对于 30% 的数据, $N \leq 3000$

对于 100% 的数据, $1 \leq N, A_i \leq 2 * 10^5$

简单题 (easy)

【问题描述】

小 P 完成了签到之后就开始了比赛，他看了眼第一题，第一题的题意是这样的：

KCZ 王国有 N 个城市，被 M 条双向道路连接。每条双向道路都有一个维修费用。国王 KCZ 想要去掉尽可能多的边，使得留下来的边依旧能使王国的任意两个城市之间都有经过一条或多条边的路径。在所有的方案中，国王 KCZ 希望留下来的边的维修费用之和最小。

现在，国王 KCZ 准备调整一些边的维修费用来影响最后留下的边的结果。对于每条边，他希望你告诉他，在保证这条边有希望被留下来的同时，这条边的维修费用最大能是多少。

小 P 觉得这是个简单题，随手就切掉了，但是他觉得这个题很有趣，所以希望你也来做一做。

【输入格式】

第一行两个整数 N, M ，表示城市的数量和双向道路的数量。

接下来 M 行，每行 3 个整数 u, v, w ，表示在城市 u 和 v 之间有一条双向道路，维修费用为 w 。

保证不存在重边和自环。

【输出格式】

输出共 M 行，按照输入给出的边的顺序分别给出每条边最大的维修费用，如果最大维修费用大于 10^9 ，则输出 10^9 。

【样例输入1】

```
6 7
1 2 2
2 3 1
3 4 0
1 4 3
3 5 20
4 5 8
3 6 14
```

【样例输出1】

```
3
3
3
2
8
20
1000000000
```

【样例2】

见下发文件中。

【数据范围及约定】

对于 30% 的数据, $N, M \leq 3000$

对于 60% 的数据, $N, M \leq 8000$

对于 100% 的数据, $1 \leq N \leq 10^5$, $N - 1 \leq M \leq 10^6$, $0 \leq w_i \leq 10^9$

不难题 (nothard)

【问题描述】

小 P 飞速切完第一题之后, 觉得题目太简单了, 于是直接翻到了最后一题, 最后一题的题意是这样的:

在 KCZ 王国有 K 口恶魔之井, 传说国王 KCZ 曾经斩杀了 N 只恶魔, 并将他们的身体都分成了 K 块, 并分别投入了 K 口恶魔之井中。由于封印力量的流失, 恶魔的力量开始蠢蠢欲动, 国王 KCZ 下令铸造了一口新的巨大的恶魔之井来容纳所有的碎块。

为了将碎块转移到新的恶魔之井中, 国王 KCZ 每一次可以选择一个原来的恶魔之井, 并将井口的碎块取出, 然后立即放入新的恶魔之井中。为了不发生意外, 国王 KCZ 不能将连续的 K 个属于同一只恶魔的碎块放入新的恶魔之井。

但是, 由于能力有限, KCZ 王国可能无法建造如此巨大的恶魔之井, 所以, 国王 KCZ 对恶魔之井进行了 1 到 K 的编号, 并对于任意一组连续的恶魔之井 $[L, R]$, 他想知道自己有多少种不同的取出碎块的方案。对于两个方案视为不同, 当且仅当存在一个 i , 取出的第 i 个碎块来自不同的恶魔之井。注意, 现在的限制条件为, 不能有连续的 $R - L + 1$ 个属于同一只恶魔的碎块放入新的恶魔之井中。

恶魔之井的信息通过 K 个 N 的排列给出, 第 i 个排列代表第 i 个恶魔之井按照从井口到井底的顺序给出的碎块所属恶魔的编号。

小 P 觉得这个题不是很难, 很快就切掉了, 但是他觉得这个题很有趣, 所以希望你也来做一做。

为了降低难度, 小 P 帮你打乱了每一个排列, 也就是说, K 个排列均为一个随机的排列, 每个排列均互相独立地, 等概率地从 $N!$ 种排列中选出。同时, 你只需要输出所有方案的总和, 即设 $F(L, R)$ 为区间 $[L, R]$ 的方案数, 你只需要回答: $\sum_{L=1}^K \sum_{R=L+1}^K f(L, R)$ 。

【输入格式】

第一行两个整数 K 和 N 。

接下来 K 行, 每行一个 N 的排列。

【输出格式】

输出一个整数, 表示答案, 模 $(10^9 + 7)$ 。

【样例输入1】

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

【样例输出1】

```
1464
```


【样例输入2】

```
4 2
1 2
2 1
1 2
2 1
```

【样例输出2】

```
2466
```

【样例3】

见下发文件中。

【数据范围及约定】

对于 20% 的数据, $N \leq 100$, $K \leq 20$;

对于 50% 的数据, $N, K \leq 100$;

对于 100% 的数据, $2 \leq N, K \leq 300$ 。