

NOIP 模拟赛

时间：2024 年 11 月 31 日 8 : 00 ~ 12 : 30

| | | | | |
|---------|------------|--------------|----------|-----------|
| 题目名称 | 不解之题 | 白日 | 唱 | 海馬成長痛 |
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 目录 | aporia | ceremony | sing | hippo |
| 可执行文件名 | aporia | ceremony | sing | hippo |
| 输入文件名 | aporia.in | ceremony.in | sing.in | hippo.in |
| 输出文件名 | aporia.out | ceremony.out | sing.out | hippo.out |
| 每个测试点时限 | 2.0 秒 | 2.0 秒 | 2.0 秒 | 2.0 秒 |
| 内存限制 | 512 MB | 512 MB | 512 MB | 512 MB |
| 测试点数目 | 10 | 40 | 25 | 20 |
| 测试点是否等分 | 是 | 否 | 是 | 是 |

提交源程序文件名

| | | | | |
|-----------|------------|-----------|----------|------------|
| 对于 C++ 语言 | aporia.cpp | class.cpp | hippo.in | aporia.cpp |
|-----------|------------|-----------|----------|------------|

注意事项与提醒：

- 编译选项: `-O2 -std=c++14 -static`
- 文件名（程序名和输入输出文件名）必须使用英文小写。
- C/C++ 中函数 `main()` 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`。
- 提交的程序代码文件的放置位置请参照具体要求。
- 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。
- 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）。
- 程序可使用的栈内存空间限制与题目的内存限制一致。
- 只提供 Linux 格式附加样例文件。
- 评测在 Linux 下进行，各语言的编译器版本以其为准。
- 禁止在源代码中改变编译器参数（如使用 `#pragma` 命令），禁止使用系统结构相关指令（如内联汇编）和其它可能造成不公平的方法。

不解之题 (aporia)

【题目描述】

对于一张每一个节点入度，出度均为 1 的有向图，定义其周期 r 为：

- 在最开始(时刻 0)，第 i 个节点站有编号为 i 的人。
- 之后的每一个时刻，每一个人沿着节点的唯一出边移动恰好一步。

则 r 为一个最小的时刻 $t(t > 0)$ ，使得对于任意 $i \in [1, n]$ ，此时第 i 个人重新回到节点 i 。

你需要对于全部满足入度，出度均为 1 的有向图，求其周期的 k 次方之和。由于答案可能过大，所以你只需要输出答案对 998244353 取模的值即可。

【输入格式】

从文件 `aporia.in` 中读入数据。

第一行包含两个整数 n, k 。

【输出格式】

输出到文件 `aporia.out` 中。

一行一个整数表示答案。

【样例 1 输入】

```
2 2
```

【样例 1 输出】

```
5
```

【样例 2 输入】

```
4 5
```

【样例 2 输出】

```
8377
```

【样例 3 输入】

```
60 0
```

【样例 3 输出】

```
563101905
```

【样例 4 输入】

50 1000

【样例 4 输出】

77436607

【样例 5 输入】

97 1145

【样例 5 输出】

671375992

【数据范围】

| 测试点编号 | n | k |
|---------|------------|-------------|
| 1 | ≤ 13 | $\leq 10^9$ |
| 2, 3, 4 | ≤ 100 | $= 0$ |
| 5 | ≤ 50 | $\leq 10^9$ |
| 6 | ≤ 60 | $\leq 10^9$ |
| 7 | ≤ 70 | $\leq 10^9$ |
| 8 | ≤ 80 | $\leq 10^9$ |
| 9 | ≤ 90 | ≤ 10 |
| 10 | ≤ 100 | $\leq 10^9$ |

白日 (ceremony)

【题目描述】

有一个 n 个节点， $n - 1$ 条边的无向连通图，每一个点都具有一个点权 a_i 。

你可以选择一个点作为源点，然后整张图变换为一个以此源点作为根的外向树。在此图上运行拓扑排序算法，得到一个拓扑序 $\{p_1, p_2, \dots, p_n\}$ ，认为此拓扑序的权值为

$$\min_{i=1}^n \left\{ \sum_{j=1}^i a_{p_j} \right\}$$

现在希望对于每一个源点 $S = 1, 2, \dots, n$ ，求出所有可能的拓扑序中权值最大值。

【输入格式】

从文件 `ceremony.in` 中读入数据。

本题有多组测试数据。

输入的第一行包含两个正整数 T ，表示数据组数。

接下来包含 T 组数据，每组数据的格式如下：

第一行包含一个正整数 n ，表示节点的数量。

第二行包含 n 个正整数 a_1, a_2, \dots, a_n ，表示每一个节点的点权。

接下来的 $n - 1$ 行，每行包含两个正整数 u, v 表示原图存在一条连接 u, v 两个节点的边。

【输出格式】

输出到文件 `ceremony.out` 中。

对于每一组测试数据，一行 n 个整数表示答案。第 i 个整数表示 $S = i$ 时，所有可能的拓扑序 $\{p\}$ 中的权值最大值。

【样例 1 输入】

```
1
7
-182 -247 -360 148 -122 -116 177
1 2
1 3
3 4
3 5
2 6
2 7
```

【样例 1 输出】

```
-702 -702 -702 -702 -763 -728 -702
```

【数据范围】

本题实际测试不开启子任务捆绑，下表仅供展示测试点梯度与分值。

| 子任务编号 | $\sum n$ | 特殊性质 | 分值 |
|----------|------------------------|--------------------------------|----|
| subtask1 | ≤ 9 | 无 | 10 |
| subtask2 | ≤ 16 | 无 | 10 |
| subtask3 | ≤ 600 | 一条链, 保证 $u_i = i, v_i = i + 1$ | 20 |
| subtask4 | $\leq 2.5 \times 10^3$ | 菊花图 | 10 |
| subtask5 | ≤ 600 | 无 | 25 |
| subtask6 | $\leq 2.5 \times 10^3$ | 无 | 25 |

唱 (sing)

【题目描述】

今天你学习了二分图染色算法，为了考验你是否真正理解了这一有趣的算法，我来考考你。

现在给出了一个具有 n 个点的二分图，为了通过不正经的方式得到二分图的一组合法染色方案，我要执行“伪二分图染色算法”，具体流程如下：

- 第一步：等概率的随机一个 1 到 n 的排列 $\{p_1, p_2, \dots, p_n\}$ 。
- 第二步：从 $1 \sim n$ 遍历每一个 p_i ，接下来分三类讨论：
 1. 如果不存在一个与 p_i 相邻且已经被染色的节点，则将 p_i 染成黑色。
 2. 如果存在一个与 p_i 相邻且已经被染成黑色的节点，则将 p_i 染成白色。
 3. 如果不满足上述两种情况，将 p_i 染成黑色。
- 第三步：检验是否是一个合法的染色。这里认为一个染色方案合法当且仅当不存在一条边两端的点染色相同。如果是一组合法的染色，则结束程序。否则返回第一步。

那么如果想要通过上述“伪二分图染色算法”得到一个合法的染色，期望运行“第一步”几遍？答案对 998244353 取模。

【输入格式】

从 `sing.in` 中读入数据。

第一行包含两个正整数 n, m ，分别表示二分图的节点数量和边的数量。

接下来的 m 行，每行包含两个正整数 u_i, v_i ，表示二分图中存在一条连接 (u_i, v_i) 的双向边。

保证给出的图是一个二分图。

【输出格式】

输出到文件 `sing.out` 中。

输出一行包含一个非负整数，表示期望下“第一步”运行次数。

【样例 1 输入】

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

【样例 1 输出】

```
598946613
```

【数据范围】

样例解释：期望操作次数为 $\frac{6}{5}$ 。

| 测试点编号 | n | m | 特殊性质 | 分值 |
|----------|-----------|-------------------------|------|----|
| subtask1 | ≤ 18 | $\leq \frac{n(n-1)}{2}$ | 无 | 16 |
| subtask2 | ≤ 22 | $\leq \frac{n(n-1)}{2}$ | 无 | 8 |
| subtask3 | ≤ 30 | $\leq \frac{n(n-1)}{2}$ | 无 | 24 |
| subtask4 | ≤ 40 | $\leq \frac{n(n-1)}{2}$ | 有 | 24 |
| subtask5 | ≤ 40 | $\leq \frac{n(n-1)}{2}$ | 无 | 16 |
| subtask6 | ≤ 45 | $\leq \frac{n(n-1)}{2}$ | 无 | 12 |

特殊性质：保证对于任意一条边 u_i, v_i ，总有 $2 \times |u_i - v_i| \geq n$ 。

海馬成長痛 (hippo)

【题目描述】

现在有一个深度为 n 的满 m 叉树，根节点的深度被认为是 0 。每一个叶子节点都有一个权值(初始化为 0)，每一个非叶子节点的权值是其 m 个儿子的权值和。为了方便辨认每一个叶子，将所有叶子按照从左到右的顺序依次编号为 $1, 2, \dots, m^n$ 。

如果一个长度为 m^n 的操作序列 a 满足如下条件：

- 序列是一个 1 到 m^n 的排列。
- 第 i 次操作将编号为 a_i 的叶子权值设为 1 ，并更新整棵树的权值。要求每一次操作完毕之后，任意一个非叶子节点全部儿子权值的极差不可以超过 1 。

那么这个操作序列合法。

求出全部合法的 a 是一件容易的事情，你从来不做容易的事情。具体来说，依次给出了 k 个限制，第 i 个限制是指将序列 a 中编号为 b_i 的位置上的值强制设为 c_i 。

在每一次添加限制之后，你需要求出在满足当前给出的全部限制的前提下，还有多少个 a 序列满足条件？

答案对 998244353 取模。

【输入格式】

从 `hippo.in` 中读入数据。

第一行包含三个正整数 n, m, k ，表示树的深度，儿子数量以及限制的数量。

第二行包含 k 个正整数表示 b_1, b_2, \dots, b_k 。保证 b_i 互不相同。

第三行包含 k 个正整数表示 c_1, c_2, \dots, c_k 。保证 c_i 互不相同。

【输出格式】

输出到文件 `hippo.out` 中。

第一行输出在不添加任何限制的情况下，有多少个合法的序列 a 。

接下来输出 k 行。第 i 行一个整数表示在满足前 i 个限制的前提下，还有多少个合法的序列 a 。

【样例 1 输入】

```
2 3 4
9 1 6 7
1 7 2 6
```

【样例 1 输出】

```
46656
5184
576
96
16
```


【数据范围】

| 测试点编号 | m | m^n | k |
|----------------|-------------|----------------|----------------------|
| 1, 2, 3, 4 | ≤ 16 | ≤ 16 | ≤ 20 |
| 5, 6, 7, 8 | ≤ 20 | $\leq 10^3$ | $\leq 10^3$ |
| 9, 10 | $\leq 10^5$ | $\leq 10^{18}$ | $= 0$ |
| 11, 12, 13, 14 | $\leq 10^5$ | $\leq 10^{18}$ | $\leq 10^3$ |
| 15, 16, 17, 18 | $\leq 10^5$ | $\leq 10^{18}$ | $\leq 2 \times 10^4$ |
| 19, 20 | $\leq 10^5$ | $\leq 10^{18}$ | $\leq 10^5$ |