

# 全国青少年信息学奥林匹克竞赛

## NOI2023 模拟

时间：7:30–12:20

题目名称	氦	锂	最短路
题目类型	传统型	传统型	传统型
目录	he	li	shortest
可执行文件名	he	li	shortest
输入文件名	he.in	li.in	shortest.in
输出文件名	he.out	li.out	shortest.out
每个测试点时限	1.0 秒	2.0 秒	--秒
内存限制	1024MB	1024 MB	-- MB
子任务数目	Subtask	Subtask	10
测试点是否等分	否	否	是

### 提交源程序文件名

对于 C++语言	he.cpp	li.cpp	shortest.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. 选手提交的源程序必须存放在已建立好的，且带有样例文件和下发文件的文件夹中，文件夹名称与对应试题英文名一致。
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. C++中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
4. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
5. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 每道题目所提交的代码文件大小限制为 100KB。
9. 若无特殊说明，输入文件与输出文件中同一行的相邻整数均使用一个空格分隔。
10. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。
11. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。
12. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。
13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外不允许在程序中手动开启其他编译选项，一经发现， 本题成绩以 0 分处理。

## 氦 (he)

### 题目描述

很久很久以前，有一个神秘的**有根二叉树**  $T$ ，包含  $n$  个结点，并且以 1 为根。

有  $k$  个球将会以**随机的顺序**被扔到树上，编号为  $i$  的球会被扔到结点  $p_i$  上。

当一个球被扔到结点  $x$  时：

- 如果  $x$  的所有儿子上均有球，则这个球就停在  $x$ ；
- 如果  $x$  恰好有一个儿子上没有球，则递归将这个球扔到这个儿子上；
- 如果  $x$  有两个儿子都没有球，则：
  - 如果球是从  $x$  的父亲下到  $x$  的，那么递归将这个球扔到与  $x$  同方向的儿子上（如  $x$  是父亲的左儿子，则递归到  $x$  的左儿子）；
  - 如果球是一开始就扔在  $x$  的，那么递归将这个球扔到随机的一个儿子上。

显然在所有球扔下后，每个点上至多只有一个球，定义一种可能的最终结果的  $(\circlearrowleft\circlearrowright)$  序列是一个长度为  $n$  的序列  $a_1, \dots, a_n$ ，其中  $a_i$  表示最终在点  $i$  上的球的编号。

求有多少种可能的  $(\circlearrowleft\circlearrowright)$  序列，对  $10^9 + 7$  取模。

### 输入格式

第一行：两个整数  $n, k$ 。

第二行： $k$  个整数  $p_1, \dots, p_k$ 。

接下来  $n$  行：第  $i$  行两个整数  $l_i, r_i$ ，表示  $i$  的左右儿子编号，若不存在则用 0 代替。

### 输出格式

输出一行一个整数，表示答案。

### 样例 1 输入

1	5 2
2	1 3
3	2 3
4	0 0
5	4 5
6	0 0
7	0 0

### 样例 1 输出

1	4
---	---

### 样例 1 解释

先扔编号为 1 的球，再扔编号为 2 的球，可能得到的  $(\circlearrowleft\circlearrowright)$  序列有  $(0, 1, 0, 2, 0), (0, 1, 0, 0, 2), (0, 0, 0, 1, 2), (0, 0, 0, 2, 1)$ 。

先扔编号为 2 的球，再扔编号为 1 的球，可能得到的  $(\circlearrowleft\circlearrowright)$  序列有  $(0, 0, 0, 2, 1), (0, 0, 0, 1, 2)$ 。

总共有 4 种可能的  $(\circlearrowleft\circlearrowright)$  序列。

样例 2 输入

1	4	3
2	1	2 4
3	0	2
4	0	3
5	0	4
6	0	0

样例 2 输出

1	2
---	---

样例 2 解释

共有两种可能的  $(\circ \downarrow \circ)$  序列：

$(0, 1, 2, 3)$ ：先扔编号为 3 的球，再扔编号为 2 的球，最后扔编号为 1 的球。

$(0, 2, 1, 3)$ ：先扔编号为 3 的球，再扔编号为 1 的球，最后扔编号为 2 的球。

数据范围

- 对于 100% 的数据： $1 \leq n, k \leq 4000, 1 \leq p_i \leq n, 0 \leq l_i, r_i \leq n$ 。
- Subtask 1 (20%)：  $n \leq 12, k \leq 6$ 。
- Subtask 2 (30%)：  $l_i = 0$ 。
- Subtask 3 (20%)：  $p_i$  两两不同。
- Subtask 4 (30%)： 无特殊限制。

锂 (li)

题目描述

很久很久以前，有一个神秘的无向完全图  $G$ ，包含  $n$  个结点。

现在给这张图的所有边赋权，边权是  $1, 2, \dots, \frac{n(n-1)}{2}$  的一个排列，在所有可能情况中等概率随机。

称一条边  $(u, v)$  是  $(\bullet \text{---} \bullet)$  的，当且仅当它的权值是以  $u$  为端点的边权的最大值，或者是以  $v$  为端点的边权的最大值。

由全部  $n$  个点和所有  $(\bullet \text{---} \bullet)$  边构成的子图称为  $(\bullet \text{---} \bullet)$  子图。

求  $G$  的  $(\bullet \text{---} \bullet)$  子图恰好有  $k$  个连通块的概率，对素数  $p$  取模。

输入格式

第一行：三个整数  $n, k, p$ 。

输出格式

输出一行一个整数，表示答案。

样例 1 输入

1

5 1 998244353

样例 1 输出

1

427819009

样例 1 解释

所求概率为  $\frac{4}{7}$ 。

样例 2 输入

1

6 3 998244353

样例 2 输出

1

617960790

样例 2 解释

所求概率为  $\frac{1}{21}$ 。

数据范围

对于 100% 的数据： $2 \leq n \leq 10^7, 1 \leq k \leq n, 9 \times 10^8 \leq p \leq 10^9$ ，保证  $p$  为素数。

子任务编号	$n \leq$	$p$	分值
1	5	/	10
2	50	/	20
3	5000	/	20

子任务编号	$n \leq$	$p$	分值
4	$10^5$	= 998244353	10
5	$10^7$	/	40

# 最短路 (shortest)

## 【题目描述】

有一个结点 1 和结点  $n$  连通的正权无向图，求删除不超过  $K$  条边，使得结点 1 和结点  $n$  仍然连通的同时，令这两个结点之间的最短路尽可能长。

## 【输入格式】

第一行包含三个整数  $n, m, K$  表示点数、边数与限制条件。结点从 1 到  $n$  编号，边从 1 到  $m$  编号。

接下来  $m$  行每行三个正整数  $u, v, w$  表示一条连接  $(u,v)$  的权值为  $w$  的无向边。

## 【输出格式】

第一行一个非负整数  $T$  表示需要删除的边数。

接下来  $T$  行每行一个 1 到  $m$  之间的整数  $x$  表示删掉输入中的第  $x$  条边。你需要保证  $T$  个整数互不相同

## 【评分标准】

对于每个测试点，设有评分四个参数  $s_1, s_2, s_3, s_4$ 。假设你的方案的最短路为  $ans$ 。

如果你没有输出，或者输出不合法，或者最短路不存在，得 0 分。

如果最短路存在，得 1 分。

如果  $ans \leq s_1$ ，得 3 分。

如果  $ans \leq s_2$ ，得 5 分。

如果  $ans \leq s_3$ ，得 8 分。

如果  $ans = s_4$ ，得 10 分。

如果  $ans > s_4$ ，得 12 分。

取满足条件的分数中的最高得分为该测试点你的得分。

## 【如何测试你的输出】

首先编译 `checker.cpp`

在你的目录下有一个名为 `checker` 的程序可以用来检查你的输出，你可以在终端中使用以下命令来检查你的输出：

`./checker N`

（不同环境下自行更改命令）其中 N 为测试点的编号，例如，要测试第 3 个测试点可以使用

`./checker 3`

该程序会检测你的输出方案是否合法。如果方案合法，程序还会给出该方案的最短路的长度值。