

# 冲刺 CCF NOI2025 全国赛模拟题

## 梦熊集训-Day3

时间：2025 年 07 月 06 日 08:00 ~ 13:00

题目名称	迷宫	排列计数	白井黑子
题目类型	传统型	传统型	交互型
目录	maze	permutation	kuroko
可执行文件名	maze	permutation	kuroko
输入文件名	maze.in	permutation.in	N/A
输出文件名	maze.out	permutation.out	N/A
每个测试点时限	3.0 秒	1.0 秒	7.0 秒
内存限制	512 MiB	256 MiB	256 MiB
测试点数目	50	10	25
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	maze.cpp	permutation.cpp	kuroko.cpp
-----------	----------	-----------------	------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

注意事项（请仔细阅读）

1. 选手提交的源程序必须存放在以自己姓名命名的文件夹中，文件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。
14. 评测时采用的机器配置为：Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz，内存 32GB。上述时限以此配置为准。

# 迷宫 (maze)

## 【题目描述】

小 S 没有妹子陪他玩，因此只能在迷宫里玩。

迷宫里一共有  $n$  个房间，其间有  $n - 1$  个走廊。保证所有的房间都连通。容易发现任意两个房间在不走回头路的情况下只有一种方案从一个房间经过一系列房间走到另一个房间。

在每个走廊上都有一个不超过  $k$  的正整数  $w_i$ 。

小 S 决定进行  $q$  次行走，每次行走中他会从房间  $x$  不经过重复房间走到房间  $y$ 。在行走的过程中，他准备了一张纸。经过每条走廊时，他会选择是否将走廊上的数记录到纸的下一个位置。

如果他的纸上的数单调不降，那么小 S 会很开心。

对于每次行走，请求出小 S 在开心的情况下至多记录下多少个数。

## 【输入格式】

从文件 `maze.in` 中读入数据。

第一行一个正整数  $id$ ，表示测试点编号。特别地，对于所有样例，其  $id$  为对应所符合要求的测试点的编号。

第二行三个正整数  $n, q, k$ ，分别表示房间数和小 S 进行的行走数。

接下来  $n - 1$  行，每行三个正整数  $u, v, w$ ，表示  $u$  房间与  $v$  房间之间有一条走廊，上面的正整数为  $w$ 。

接下来一行五个正整数  $x_0, y_0, a, b, c$ 。设第  $i$  次询问的  $x, y$  为  $x_i, y_i$ ，则  $x_i = (x_{i-1} \times a + y_{i-1}^2 \times b + c + i^2) \bmod n + 1$ ， $y_i = (x_{i-1}^2 \times a + y_{i-1} \times b + c^2 + i) \bmod n + 1$ 。

## 【输出格式】

输出到文件 `maze.out` 中。

输出一行，一个整数，设第  $i$  次询问的答案为  $ans_i$ ，则输出  $\bigoplus_{i=1}^q (i \times ans_i)$ ，正解不依赖于特殊的输出方式。

## 【样例 1 输入】

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

6	1 3 1 1 1
---	-----------

**【样例 1 输出】**

1	1
---	---

**【样例 1 解释】**

根据推算可以得到： $\{x_1, y_1\} = \{1, 3\}, \{x_2, y_2\} = \{4, 4\}, \{x_3, y_3\} = \{3, 1\}$ 。

- 对于第一组询问，可以在经过走廊 1, 2 时将数字记录下来，经过走廊 2, 4 时将数字记录下来，纸上的数为  $\{1, 2\}$ ，可以证明最多记录下来 2 个数。
- 对于第二组询问，无法经过走廊，于是最多记录下来 0 个数。
- 对于第三组询问，可以在经过走廊 4, 2 时将数字记录下来，经过走廊 2, 1 时不将数字记录下来，纸上的数为  $\{2\}$ ，可以证明最多记录下来 1 个数。

于是  $ans = \{2, 0, 1\}$ ， $\bigoplus_{i=1}^q (i \times ans_i) = 1$ 。

**【样例 2】**

见选手目录下的 *maze/maze2.in* 与 *maze/maze2.ans*。

**【样例 3】**

见选手目录下的 *maze/maze3.in* 与 *maze/maze3.ans*。

**【样例 4】**

见选手目录下的 *maze/maze4.in* 与 *maze/maze4.ans*。

**【样例 5】**

见选手目录下的 *maze/maze5.in* 与 *maze/maze5.ans*。

**【样例 6】**

见选手目录下的 *maze/maze6.in* 与 *maze/maze6.ans*。

**【样例 7】**

见选手目录下的 *maze/maze7.in* 与 *maze/maze7.ans*。

【样例 8】

见选手目录下的 `maze/maze8.in` 与 `maze/maze8.ans`。

【数据范围】

对于 100% 的数据： $1 \leq n \leq 10^5, 1 \leq q \leq 3 \times 10^6, 1 \leq w \leq k \leq 10, 1 \leq u, v, x_0, y_0 \leq n, 1 \leq a, b \leq 10^5$ 。

测试点编号	$n \leq$	$q \leq$	$k$	特殊性质
1	10	10	$\leq 10$	无
2 ~ 3	$10^5$	$3 \times 10^6$		$= 1$
4 ~ 5			无	
6	400	400	$\leq 10$	B
7 ~ 8				无
9 ~ 11		$3 \times 10^6$		B
12 ~ 15				无
16	3000	3000		B
17 ~ 18				无
19 ~ 21		$3 \times 10^6$		B
22 ~ 25				无
26	$5 \times 10^4$	$5 \times 10^4$	$\leq 5$	C
27 ~ 30				B
31 ~ 33				无
34		$2 \times 10^5$		C
35 ~ 38				B
39 ~ 41				无
42 ~ 43	$10^5$	$3 \times 10^6$	$\leq 10$	C
44 ~ 47				B
48 ~ 50				无

- 特殊性质 A：保证对于第  $i$  条边， $u = 1, v = i + 1$ 。
- 特殊性质 B：保证对于第  $i$  条边， $u = i, v = i + 1$ 。
- 特殊性质 C：保证对于第  $i$  条边， $u = i + 1, v$  在  $[1, i]$  中随机选择。

## 排列计数 (permutation)

### 【题目描述】

对于一个排列  $P$  (下标和值域均为  $1 \cdots n$ )，定义一个下标  $x (2 \leq x \leq n-1)$  是好的是指：

- 对于  $\forall 1 \leq i \leq x-1$  都有  $P_i < P_x$ ，且  $P_x > P_{x+1}$ 。

另外，下标 1 在  $n \geq 2$  且  $P_1 > P_2$  时也是好的，而下标  $n$  无论如何都不会是好的。对排列  $P$  定义函数  $F(P)$  如下：

1. 定义整数  $S$ ，初始值为 0。
2. 将排列  $P$  中好的下标的数量累加到  $S$ 。
3. 对排列  $P$  做一次冒泡排序，即枚举  $i$  从 1 到  $n-1$ ，如果  $P_i > P_{i+1}$ ，交换  $P_i, P_{i+1}$ 。

重复操作 2,3，直到排列  $P$  升序，此时  $S$  的值即为  $F(P)$ 。

给定  $n$ ，有一个长度为  $n$  的排列  $p$ ，其中  $m$  个位置已知，其余位置未知。

对于所有可能的排列  $p$ ，求  $F(p)$  的和，对 998244353 取模。

### 【输入格式】

从文件 `permutation.in` 中读入数据。

第一行两个整数  $n, m$ ，表示排列  $p$  的长度和确定的位置个数。

接下来  $m$  行，每行两个整数  $x, y$ ，表示  $p_x = y$ 。

保证至少有一个排列  $p$  满足这  $m$  个约束。

### 【输出格式】

输出到文件 `permutation.out` 中。

一行一个整数表示答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
1 3
```

**【样例 1 解释】**

$$p = (2, 1, 3), F(p) = 1$$

$$p = (2, 3, 1), F(p) = 2$$

**【样例 2 输入】**

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

**【样例 2 输出】**

```
1 20
```

**【样例 3】**

见选手目录下的 *permutation/permutation3.in* 与 *permutation/permutation3.ans*。

**【样例 4】**

见选手目录下的 *permutation/permutation4.in* 与 *permutation/permutation4.ans*。

**【数据范围】**

对于所有数据，保证  $1 \leq n \leq 10^6, 0 \leq m \leq n$ 。

测试点编号	$n \leq$	特殊性质
1	8	无
2	300	A
3		无
4		
5	2000	B
6		无
7	$10^5$	A
8		无
9	$10^6$	B
10		无

特殊性质 A：满足  $m = n$ ；

特殊性质 B：满足  $m = 0$ ；



## 白井黑子 (kuroko)

### 【题目描述】

此题是黑子测试你是否具有空间系超能力而制作的。

给定一棵以 1 为根有根树，标号 1 到  $n$ ，求这棵树的 DFS 序。

具体地，有如下伪代码：

```

1 int d[n], tot = 0
2 set son[n]
3 void dfs(int u):
4     d[tot] = u
5     tot += 1
6     for v in son[u]: // in any order
7         dfs(v)
8 dfs(1)

```

$d$  数组即为所求。

显然，一棵树可以有多个 DFS 序，你只要求出任意一个即可。

在本题中，你不可以直接定义数组，而是要使用交互库提供的数组来计算答案。

### 【实现细节】

请确保你的程序开头有 `#include "dfs.h"`。

头文件 `dfs.h` 包含了如下内容：

引用了头文件 `bits/stdc++.h`，这一点对本题的空间限制至关重要。

定义了结构体 `Array` 表示长度  $n$ ，值域  $[0, n]$  的数组。该类型有两个成员函数：

```

1 int operator[](int x);

```

该函数表示获取该数组第  $x$  位的值。必须满足  $x \in [0, n]$ 。

```

1 void set(int x, int y);

```

该函数表示将数组第  $x$  位设置为  $y$ 。必须满足  $x \in [0, n]$  以及  $y \in [0, n]$ 。

另外，你还可以调用函数

```

1 int fa(int x);

```

它返回  $f[x]$  的初始值 ( $x$  的父亲)，要求  $x \in [0, n]$ 。如果调用了该函数，仅能获得当前测试点 50% 的分数。

以上所有函数均消耗常数时间。你可以使用下发文件中的样例交互库来进一步测试效率，保证下发文件和实际评测时的交互库中以上函数和结构体的实现完全相同。

你需要编写一个函数

```
1 void dfs(int n, Array &f, Array &a);
```

其中  $f[i]$  的初始值为点  $i$  的父亲，这里规定根的父亲为 0，且  $f[0] = 0$ ； $a[i]$  初始值全部为 0。

该函数会被调用恰好一次。你需要在函数返回时，在  $a$  中写入你求出的 DFS 序。具体地，对于每一个  $i \in [1, n]$ ，最后一次调用 `a.set(i, x)` 的  $x$  需为 DFS 序的第  $x$  项， $a[0]$  的值没有要求。

保证交互库在交互以外的用时不超过 1s。交互函数的用时可使用样例交互库测试。

保证交互库消耗的空间不超过 255.15MiB。

请注意，任何不使用提供的函数操作结构体（如直接修改数据成员）的行为被视为攻击交互库。

### 【程序测试】

下发文件中提供了 `grader.cpp`，可以与你的代码交互并检查输出是否正确。

可使用如下命令得到可执行文件：

```
1 g++ kuroko.cpp grader.cpp -o kuroko -std=c++14 -O2 -lm -static
```

得到可执行文件后，需按如下格式输入：

第一行，一个正整数  $n$ 。

接下来一行  $n - 1$  个整数，表示  $2 \sim n$  号结点的父亲。

与 `grader` 不同的是，样例输出为一组合法的答案。

### 【样例 1 输入】

```
1 6
2 6 5 1 1 5
```

### 【样例 1 输出】

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

### 【样例 2】

见选手目录下的 `kuroko/kuroko2.in` 与 `kuroko/kuroko2.ans`。

### 【样例 3】

见选手目录下的 *kuroko/kuroko3.in* 与 *kuroko/kuroko3.ans*。

**【数据范围】**

对于 100% 的数据： $1 \leq n \leq 10^7$ 。

测试点编号	$n \leq$	特殊性质
1 ~ 3	1000	无
4 ~ 7	$7 \times 10^4$	
8 ~ 11	$10^6$	
12 ~ 15	$5 \times 10^6$	
16 ~ 20	$10^7$	A
21 ~ 25		无

特殊性质 A：对于  $i > 0$ ，有初始  $f[i] < i$ 。