

# NOIP模拟赛

时间：2023 年 10 月 ?? 日 ??:?? ~ ??:??

题目名称	道路	烟花	彩排	树上的数
题目类型	传统型	传统型	传统型	传统型
输入文件名	road.in	firework.in	yacolorful.in	tree.in
输出文件名	road.out	firework.out	yacolorful.out	tree.out
每个测试点时限	2.0 秒	2.0 秒	2.0 秒	3.0 秒
内存限制	512 MB	512 MB	512 MB	512 MB
测试点数目	10	10	10	10
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	road.cpp	firework.cpp	yacolorful.cpp	tree.cpp
-----------	----------	--------------	----------------	----------

编译选项

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

## 注意事项

1. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 `0`。
3. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格分隔。
4. 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。
5. 程序可使用的栈空间内存限制与题目的内存限制一致。
6. 题目不一定按照难度顺序排序，请注意掌握时间。

## 道路 (road)

### 【题目描述】

小 T 在城市里随机游走。

城市的交通网络由  $n$  条横向道路与  $m$  条纵向道路构成，每条道路都是足够长的，所以它们形成了  $n \times m$  个十字路口。

按从上到下的顺序将  $n$  条横向道路标号为  $1 \sim n$ ，按从左到右的顺序将  $m$  条纵向道路标号为  $1 \sim m$ 。由横向道路  $i$  和纵向道路  $j$  交叉形成的十字路口记作  $(i, j)$ 。

小 T 通过导航发现，第  $i$  条横向道路的拥挤度为  $a_i$ ，第  $j$  条纵向道路的拥挤度为  $b_j$ 。

有  $q$  次询问，每次询问给定  $s_x, s_y, t_x, t_y$ 。表示假如小 T 从十字路口  $(s_x, s_y)$  出发，经过若干条道路到达十字路口  $(t_x, t_y)$ ，那么他经过的道路中，拥挤度的最大值最小是多少。保证  $(s_x, s_y)$  与  $(t_x, t_y)$  不重合。

### 【输入格式】

从文件 `road.in` 中读入数据。

第一行三个整数  $n, m, q$ ，分别表示横向道路数，纵向道路数和询问次数。

第二行  $n$  个整数  $a_i$ ，表示每条横向道路的拥挤度。

第三行  $m$  个整数  $b_i$ ，表示每条纵向道路的拥挤度。

接下来  $q$  行，每行四个整数  $s_x, s_y, t_x, t_y$ ，表示这次询问的参数。

### 【输出格式】

输出到文件 `road.out` 中。

共  $q$  行，每行一个整数  $ans$ ，表示答案。

### 【样例 1 输入】

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

### 【样例 1 输出】

```
3
3
```

5  
2  
5

### 【样例 1 解释】

对于第 1 次询问，从十字路口 (1,1) 到十字路口 (2,2) 的最优策略是，先经过横向道路 1 到达十字路口 (1,4)，再经过纵向道路 4 到达十字路口 (2,4)，最后经过横向道路 2 到达十字路口 (2,2)。所以答案为  $\max\{a_1, b_4, a_2\} = 3$ 。

### 【样例 2】

见选手目录下的 *road/road2.in* 与 *road/road2.ans*。

### 【样例 3】

见选手目录下的 *road/road3.in* 与 *road/road3.ans*。

### 【样例 4】

见选手目录下的 *road/road4.in* 与 *road/road4.ans*。

### 【数据范围与提示】

对于 20% 的数据， $n, m, q \leq 300$ 。

对于 40% 的数据， $n, m, q \leq 5000$ 。

对于另外 30% 的数据， $b_i = 1$ 。

对于 100% 的数据， $2 \leq n, m, q \leq 2 \times 10^5$ ， $1 \leq a_i, b_i \leq 10^9$ ， $1 \leq s_x, t_x \leq n$ ， $1 \leq s_y, t_y \leq m$ ，保证  $(s_x, s_y)$  与  $(t_x, t_y)$  不重合。

## 烟花 (firework)

### 【题目描述】

小 T 有一棵  $n$  个点的树  $T$ 。

定义一个点的度数为有多少与它相连的边。

称一棵有根树是好的，当且仅当根的度数  $\leq m$ ，且除了根以外，每个点的度数  $\leq 2$ 。其中  $m$  是一个给定的参数。

对于树  $T$  的每一条边  $(u, v)$ ，小 T 想知道，如果将  $u, v$  这两个点缩成一个新点  $w$ ，那么在缩点之后的新树  $T'$  中可以找到多少包含  $w$  的连通子图，使得在以  $w$  为根时，这个连通子图是好的。你只需要求出答案对 998244353 取模的结果。注意，询问之间独立。

更精确地说， $T'$  的点集由  $T$  的点集删掉点  $u, v$  再加入新点  $w$  构成。 $T$  的边集删除边  $(u, v)$  之后，再将边集里所有的点  $u, v$  均替换为点  $w$ ，这构成了  $T'$  的边集。

### 【输入格式】

从文件 **firework.in** 中读入数据。第一行两个整数  $n, m$ ，表示树  $T$  的大小与给定参数。

接下来  $n - 1$  行，第  $i$  行两个整数  $u_i, v_i$ ，表示一条树边  $(u_i, v_i)$ ，保证这些边构成了一棵树。

### 【输出格式】

输出到文件 **firework.out** 中。

输出  $n - 1$  行，第  $i$  行一个整数  $ans_i$ ，表示边  $(u_i, v_i)$  的答案对 998244353 的结果。

### 【样例 1 输入】

```
7 3
1 2
1 3
2 4
2 5
3 6
1 7
```

### 【样例 1 输出】

```
22
16
10
10
6
12
```

**【样例 1 解释】**

例如，将边  $(1, 2)$  缩成一个新点 8 的时候，点集  $\{8, 3, 7, 6, 5\}$  所在的连通子图在以 8 为根时是好的，点集  $\{8\}$  所在的连通子图在以 8 为根是也是好的。但是，点集  $\{8, 3, 7, 4, 5\}$  不是好的，因为此时点 8 的度数为 4。

**【样例 2】**

见选手目录下的 *firework/firework2.in* 与 *firework/firework2.ans*。

**【样例 3】**

见选手目录下的 *firework/firework3.in* 与 *firework/firework3.ans*。

**【数据范围与提示】**

对于 30% 的数据， $n \leq 5000$ 。

对于另外 10% 的数据，保证树是一条链，但不保证点的标号有性质。

对于另外 20% 的数据， $u_i = i + 1$ ， $v_i$  在  $[1, i]$  之间独立均匀随机生成。

对于另外 10% 的数据， $m = 2$ 。

对于 100% 的数据， $2 \leq n \leq 3 \times 10^5$ ， $2 \leq m \leq 50$ ， $1 \leq u_i, v_i \leq n$ ，保证这些边构成了一棵树。

## 彩排 (yacolorful)

### 【题目描述】

小 T 有一个  $1 \sim n$  的排列  $a$ 。保证排列  $a$  均匀随机生成。

称一个长为  $L$  的序列  $b$  是好的，当且仅当对所有  $1 \leq i < j \leq L$  且  $j - i \leq n - 2$ ，都满足  $b_i \neq b_j$ 。

请找出一个长为  $L$  的好的序列  $b$ ，满足  $n \leq L \leq 5.5 \times 10^5$ ， $1 \leq b_i \leq n$ ，且对所有  $1 \leq j \leq n$ ，有  $b_j = j$  且  $b_{L-n+j} = a_j$ 。

你只需要输出任意一组合法方案。可以证明，一定至少存在一组合法方案。

### 【输入格式】

从文件 `yacolorful.in` 中读入数据。

第一行一个整数  $n$ ，表示序列长度。

第二行  $n$  个整数  $a_i$ ，表示序列  $a$ 。

### 【输出格式】

输出到文件 `yacolorful.out` 中。

第一行一个整数  $L$ ，表示序列  $b$  的长度，你需要保证  $n \leq L \leq 5.5 \times 10^5$ 。

第二行  $n$  个整数  $b_i$ ，你需要保证  $1 \leq b_i \leq n$ ，且序列  $b$  满足题目要求。

### 【样例 1 输入】

```
4
2 4 1 3
```

### 【样例 1 输出】

```
14
1 2 3 4 2 1 4 2 3 1 2 4 1 3
```

### 【样例 2 输入】

```
3
3 2 1
```

### 【样例 2 输出】

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

**【样例 3 输入】**

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

**【样例 3 输出】**

```
26
1 2 3 4 5 6 1 2 3 4 6 1 5 2 3 4 6 1 5 2 4 3 1 6 2 5
```

**【数据范围与提示】**

对于 10% 的数据,  $n \leq 8$ 。

对于 20% 的数据,  $n \leq 10$ 。

对于 30% 的数据,  $n \leq 20$ 。

对于 40% 的数据,  $n \leq 100$ 。

对于 50% 的数据,  $n \leq 500$ 。

对于另外 20% 的数据, 保证恰好有两个  $k$  满足  $a_k \neq k$ 。

对于 100% 的数据,  $3 \leq n \leq 1000$ ,  $1 \leq a_i \leq n$ , 保证  $a_i$  是一个  $1 \sim n$  的排列。保证排列  $a$  均匀随机生成。

## 树上的数 (tree)

### 【题目描述】

对于一棵树，记这棵树的点数为  $p$ ，小 T 认为只有树太单调了，所以小 T 决定在树上的第  $i$  条边上都放上一个数  $a_i$ ，使得  $a_1 = 1$  且对于每个  $a_i > 1$  的边  $i$ ，它都一定和一条边  $j$  相邻，且满足  $a_j < a_i$ 。两条边相邻当且仅当它们有公共点。最后，这些  $a_i$  构成了一个  $1 \sim p-1$  的排列。

放好数之后，小 T 会计算树上每个点相连的边上  $a_i$  的极差，再对这些极差求和，称为这棵树的权值。

这棵树的美丽度被定义为所有满足条件的放数方案中，这棵树权值的最小值。

小 T 有一棵  $n$  个点的树  $T$ ，但小 T 对它不是很满意，所以小 T 决定修剪一下它的枝条。

具体来说，小 T 想知道对于每个  $2 \leq i \leq n-1$ ，将树  $T$  的第  $i$  条边删除后，树  $T$  的第 1 条边所在的连通块的美丽度是多少。每次询问独立，也就是说，每次询问的删除不会影响到之后的询问。

注意：在计算该连通块的美丽度时，每条边的次序是按照原来树  $T$  中的次序重新标号得来的。例如，若连通块中有树  $T$  的第 1,3,4,6 条边，则连通块的第 1,2,3,4 条边分别为树  $T$  的第 1,3,4,6 条边。

### 【输入格式】

从文件 `tree.in` 中读入数据。

第一行一个整数  $n$ ，表示树  $T$  的大小。

接下来  $n-1$  行，第  $i$  行两个整数  $u_i, v_i$ ，表示第  $i$  条边连接了点  $u_i$  和点  $v_i$ 。保证输入的边构成一棵树。

### 【输出格式】

输出到文件 `tree.out` 中。

共  $n-2$  行，第  $i$  行一个整数，表示仅删除第  $i+1$  条边时，第 1 条边所在的连通块的美丽度。

### 【样例 1 输入】

```
7
3 4
1 4
2 4
2 6
7 5
5 3
```

### 【样例 1 输出】

```
6
```



4  
6  
5  
3

### 【样例 1 解释】

当删除第 4 条边的时候, 一种最优的  $a$  为  $[1, 4, 5, 3, 2]$ , 此时点  $1, 2, 3, 4, 5, 7$  上的极差分别为  $0, 0, 1, 4, 1, 0$ , 所以答案为 6。

### 【样例 2】

见选手目录下的 *tree/tree2.in* 与 *tree/tree2.ans*。

### 【样例 3】

见选手目录下的 *tree/tree3.in* 与 *tree/tree3.ans*。

### 【样例 4】

见选手目录下的 *tree/tree4.in* 与 *tree/tree4.ans*。

### 【数据范围与提示】

对于 10% 的数据,  $n \leq 10$ 。

对于另外 30% 的数据,  $n \leq 4000$ 。

对于另外 10% 的数据, 保证树是一条链, 但不保证边的次序和点的标号有性质。

对于另外 20% 的数据,  $u_i = i + 1$ ,  $v_i$  在  $[1, i]$  之间独立均匀随机生成。

对于 100% 的数据,  $3 \leq n \leq 5 \times 10^5$ ,  $1 \leq u_i, v_i \leq n$ , 保证输入的边构成一棵树。