

# NOIP 2024 模拟赛

## NOIP 2024 Simulation

starAndHonor

2024 年 8 月 20 日

题目名称	江南游	狐狸与葫芦	五彩路	未来轨迹的公共部分
目录	journey	fox	road	LCS
可执行文件名	journey	fox	road	LCS
输入文件名	journey.in	fox.in	road.in	LCS.in
输出文件名	journey.out	fox.out	road.out	LCS.out
每个测试点时限	2.0 秒	1.0 秒	4.0 秒	4.0 秒
内存限制	1024 MB	512 MB	1024 MB	2048 MB

提交源程序文件名

C++	journey.cpp	fox.cpp	road.cpp	LCS.cpp
-----	-------------	---------	----------	---------

编译选项

C++	-lm -O2	-lm -O2	-lm -O2	-lm -O2
-----	---------	---------	---------	---------

### 【注意事项（请仔细阅读）】

1. 选手提交的源程序请直接放在个人目录下，无需建立子文件夹；
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. 若无特殊说明，每道题的代码大小限制为 100KB。

9. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等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 题面中的跨页样例，数据将带有页眉页脚，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。

# 江南游 (journey)

【题目描述】

苏瑾又见到她了，在梦里。

桃花树下，他们相约一起去江南。江南水乡有  $n$  个城镇，编号为 1 到  $n$ ，由  $m$  条双向水道相连。他们的旅程将从  $S$  镇出发，到  $T$  镇结束。

她说想和他一起在路上看风景，于是他们决定恰好  $k$  条水道。

她说好事成双，于是他们决定经过偶数次  $X$  镇或不经过  $X$  镇。

她说想他了，于是他们紧紧相拥。

梦醒了，苏瑾想知道他们有多少种方案游江南。

【输入格式】

从文件 *journey.in* 中读入数据。

第一行包含 6 个正整数  $n, m, k, S, T, X$ 。

第二行到  $m+1$  行是水道信息，第  $i$  行包含 2 个正整数  $a_i, b_i$ ，代表城镇  $a_i$  和  $b_i$  之间存在一条双向水道。

【输出格式】

输出到文件 *journey.out* 中。

输出一个正整数，表示方案数对 998244353 取模后的结果。

【样例 1 输入】

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

【样例 1 输出】

```
4
```

【样例 1 解释】

$(1,2,1,2,3), (1,2,3,2,3), (1,4,1,4,3), (1,4,3,4,3)$  满足题意,  $(1,2,3,4,3)$  和  $(1,4,1,2,3)$  中 2 出现奇数次, 因此不满足条件。

【样例 2 输入】

```
6 5 10 1 2 3
2 3
2 4
4 6
3 6
1 5
```

【样例 2 输出】

```
0
```

【样例 2 解释】

所有城镇不一定是连通的。

【样例 3 输入】

```
10 15 20 4 4 6
2 6
2 7
5 7
4 5
2 4
3 7
1 7
1 4
2 9
5 10
1 3
7 8
```

7 9  
1 6  
1 2

【样例 3 输出】

952504739

【样例 3 解释】

答案要求对 998244353 取模。

【测试点约束】

对于所有测试点： $2 \leq N \leq 2000$ ， $1 \leq M \leq 2000$ ， $1 \leq k \leq 2000$ ， $1 \leq S, T, X \leq N$ ， $X \neq S$ ， $X \neq T$ ， $1 \leq a_i < b_i \leq N$ ，对于任意  $i \neq j$  都有  $(a_i, b_i) \neq (a_j, b_j)$   
每个测试点的具体限制见下表：

子任务	特殊限制
1	$N \leq 10, M \leq 20$
2	$M = 1$
3	存在一个城镇连接的水道数为 $M - 1$ ，且其余城镇之间没有水道相连
4	$M = N - 1$ ，且没有水道构成环
5	没有特殊限制

# 狐狸与葫芦 (fox)

### 【题目描述】

苏瑾见过的第一只“元妖”，是一只喝醉了酒钻进葫芦的一只小狐狸。他求了师傅半天，师傅终于同意让他把小狐狸留在身边。

这个世界上存在  $n$  只妖怪，编号 1 到  $n$ ，编号为  $i$  的妖怪有两种属性，记作二元组： $(a_i, b_i)$ 。

苏瑾的师傅有一个大葫芦，可以捉妖。葫芦内部的空间可以看做一个栈，刚开始是空的。如果一只属性为  $(a_i, b_i)$  的新妖怪要进入葫芦，要先不断炼化葫芦顶部的妖直至葫芦空了或葫芦顶部的妖的属性  $(a_j, b_j)$  满足  $a_i \neq a_j$  且  $b_i < b_j$ ，然后再将其装入葫芦的顶部中。

如果一只妖进入葫芦后，葫芦内只有这一只妖，则称该妖是“元妖”。“元妖”是这个世界上最适合作为宠物的妖怪啦！

师傅有  $q$  个询问  $[l_i, r_i]$ ，表示若将编号在  $[l_i, r_i]$  中的妖按编号从小到大依次吸入葫芦，会有多少只妖是“元妖”。

询问之间相互独立。

### 【输入格式】

从文件 *fox.in* 中读入数据。

第一行两个正整数  $n, q$ 。

第二行  $n$  个正整数表示  $a_i$ 。

第三行  $n$  个正整数表示  $b_i$ 。

接下来  $q$  行，每行两个正整数  $l_i, r_i$ ，表示一组询问。

### 【输出格式】

输出到文件 *fox.out* 中。

$q$  行，每行一个自然数表示一组询问的答案。

### 【样例 1 输入】

```
10 4
3 1 3 1 2 3 3 2 1 1
10 10 2 9 7 5 4 7 6 1
1 4
```

```
7 8
7 10
1 8
```

【样例 1 输出】

```
3
2
2
3
```

【样例 1 解释】

以第一次询问  $[1, 4]$  为例。  
一开始葫芦为  $\{\}$ 。  
加入 1 号妖后葫芦为  $\{(3, 10)\}$ ，葫芦中只有一只妖，该妖是“元妖”。  
加入 2 号妖  $(1, 10)$  时，葫芦顶的  $(3, 10)$  的  $b$  值不大于 2 号妖的  $b$  值，因此炼化葫芦顶的妖。此时葫芦空，2 号妖入葫芦，葫芦为  $\{(1, 10)\}$ ，该妖是“元妖”。  
加入 3 号妖  $(3, 2)$ ，此时葫芦顶元素与之  $a$  值不同， $b$  值比它更大，因而不需要炼化葫芦顶的妖，直接将 3 号妖入葫芦，葫芦为  $\{(1, 10), (3, 2)\}$ ，葫芦中有多个元素，该妖不是“元妖”。  
加入 4 号妖  $(1, 9)$ ，此时葫芦顶的妖  $(3, 2)$  的  $b$  值比它小，炼化葫芦顶的妖。炼化葫芦顶的妖后葫芦顶元素  $(1, 10)$  与  $(1, 9)$  的  $a$  值相同，继续炼化葫芦顶的妖。此时葫芦空，4 号妖入葫芦，葫芦为  $\{(1, 9)\}$ ，该妖是“元妖”。共有 3 只妖是“元妖”，因而答案为 3。

【样例 2】

见选手目录下 *fox/fox2.in* 与 *fox/fox2.ans*

【样例 3】

见选手目录下 *fox/fox3.in* 与 *fox/fox3.ans*

【样例 4】

见选手目录下 *fox/fox4.in* 与 *fox/fox4.ans*

【测试点约束】

对于所有测试点： $1 \leq n, q \leq 5 \times 10^5$ ， $1 \leq a_i, b_i \leq n$ ， $1 \leq l_i \leq r_i \leq n$ 。

每个测试点的具体限制见下表：

测试点编号	特殊限制
1 ~ 3	$n, q \leq 1000$
4 ~ 6	$n \leq 5000$
7 ~ 10	$n, q \leq 10^5$
11 ~ 12	$b_i = n - i + 1$
13 ~ 15	$a_i = i$
16 ~ 20	无



# 五彩路 (road)

## 【题目描述】

小狐狸化形了，苏瑾给她起了个名字叫做“狐雪”。

在苏瑾修炼之余，总不忘携狐雪共游这方天地。他们漫步于桃花纷飞的春日小径，穿梭于夏日蝉鸣的林间小道，共赏秋叶染金的绚烂，同迎冬雪皑皑的静美。一人一妖，如影随形，成为了世间最和谐的风景。

他们游到一处秘境，有一个  $N$  个景点，编号为  $1 \sim N$ ， $N - 1$  条路，联通且没有环，每条路有土壤颜色和长度。

苏瑾需要回答狐雪  $Q$  次询问，每次询问给出  $x_i, y_i, u_i, v_i$ ，您需要求出假定所有土壤颜色为  $x_i$  的道路长度全部变成  $y_i$  后， $u_i$  和  $v_i$  之间的距离。询问之间互相独立。

## 【输入格式】

从文件 `road.in` 中读入数据。

第一行两个正整数  $N, Q$  接下来  $N - 1$  行，每行四个正整数  $a_i, b_i, c_i, d_i$ ，表示  $a_i$  景点和  $b_i$  景点之间的道路颜色为  $c_i$ ，长度为  $d_i$  接下来  $Q$  行，每行三个正整数  $x_i, y_i, u_i, v_i$ ，询问假定所有土壤颜色为  $x_i$  的道路长度全部变成  $y_i$  后， $u_i$  和  $v_i$  之间的距离

## 【输出格式】

输出到文件 `road.out` 中。

一共  $Q$  行，每行一个整数表示每一次询问的结果

## 【样例 1 输入】

```
5 3
1 2 1 10
1 3 2 20
2 4 4 30
5 2 1 40
1 100 1 4
1 100 1 5
3 1000 3 4
```

【样例 1 输出】

130  
200  
60

【测试点约束】

对于所有测试点：满足  $2 \leq N \leq 10^5, 1 \leq Q \leq 10^5, 1 \leq a_i, b_i \leq N, 1 \leq c_i \leq N-1, 1 \leq d_i \leq 10^4, 1 \leq x_j \leq N-1, 1 \leq y_j \leq 10^4, 1 \leq u_j < v_j \leq N$   
每个测试点的具体限制见下表：

测试点编号	特殊限制
1 ~ 2	$N \leq 5, Q \leq 5$
3 ~ 12	无特殊限制

## 未来轨迹的公共部分 (lcs)

### 【题目描述】

人与妖的爱情，天地不容，狐雪不得不离开了。师傅算到苏瑾，狐雪，一个不知名的仙人的未来轨迹可以表示为三个从 1 到  $n$  的整数的排列，但是不可知。

然而苏瑾向命运神求到公共未来三元组  $(a,b,c)$ ，可以让苏瑾选取三个从 1 到  $n$  的整数的排列作为三者的未来轨迹。但是选取的三个未来轨迹  $p,q,r$ ，必须满足满足命运神的条件  $F$ 。

具体地，条件  $F$  为：

定义  $LCS(x,y)$  为序列  $x,y$  的最长公共子序列的长度。 $p,q,r$  满足

- $LCS(p,q)=a$
- $LCS(p,r)=b$
- $LCS(q,r)=c$

请你帮助苏瑾确定是否存在三个未来轨迹  $p,q,r$ ，满足条件  $F$ 。如果这样的排列存在，找出这样排列的三元组，并且告诉苏瑾。

### 【输入格式】

从文件 **LCS.in** 中读入数据。

第一行包含一个整数  $t$ ，表示测试点个数。每个测试点描述如下。

每组测试点只有一行，包含五个整数  $n,a,b,c,output$ 。

如果  $output=0$ ，只需要确定这样的排列是否存在。如果  $output=1$ ，如果这样的排列存在，你还要输出这样排列的三元组。

### 【输出格式】

输出到文件 **LCS.out** 中。

对于每个测试点，如果这样的排列  $p,q,r$  存在，输出 YES，否则输出 NO。如果  $output=1$ ，并且这样的排列存在，再输出三行：

第一行输出  $n$  个整数  $p_1,p_2,\dots,p_n$ ，表示排列  $p$ 。

第二行输出  $n$  个整数  $q_1,q_2,\dots,q_n$ ，表示排列  $q$ 。

第三行输出  $n$  个整数  $r_1,r_2,\dots,r_n$ ，表示排列  $r$ 。

如果有多个这样的三元组，输出任意一个即可。

对于每个字母，你可以输出任何大小写情况。（例如，YES，Yes，yes，yEs 都会被判定为正向答案。）

【样例 1 输入】

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

【样例 1 输出】

```
YES
1
1
1
NO
YES
1 3 5 2 6 4
3 1 5 2 4 6
1 3 5 2 4 6
NO
YES
NO
YES
NO
```

【样例 1 解释】

对于第一组测试点， $LCS((1),(1))$  是 1。

第二组测试点中，可以发现没有这样的排列存在。

第三组测试点中，其中一个例子是  $p=(1,3,5,2,6,4),q=(3,1,5,2,4,6),r=(1,3,5,2,4,6)$ 。容易发现：

$LCS(p,q)=4$ （一个最长公共子序列是  $(1,5,2,6)$ ）

LCS(p,r)=5 (一个最长公共子序列是 (1,3,5,2,4))  
LCS(q,r)=5 (一个最长公共子序列是 (3,5,2,4,6))  
第四组测试点中，可以发现没有这样的排列存在。

【测试点约束】

对于所有测试点： $1 \leq t \leq 10^5, 1 \leq a \leq b \leq c \leq n \leq 2 \cdot 10^5, 0 \leq output \leq 1$ , 保证一组测试数据中所有测试点的  $n$  的总和不超过  $2 \cdot 10^5$ 。每个子任务的具体限制见下表：

测试点编号	特殊限制
1	$a = b = 1, c = n, output = 1$
2	$n \leq 6, output = 1$
3	$c = n, output = 1$
4	$a = 1, output = 1$
5	$output = 0$
6	$output = 1$