

# 全国青少年博特学奥林匹克竞赛模拟赛

## BotOI 2024 模拟赛

时间：2024 年 3 月 12 日 07:20 ~ 11:50

题目名称	机器狗	机器人	机器猫
题目类型	传统型	传统型	传统型
目录	alphadog	infinity	unknown
可执行文件名	alphadog	infinity	unknown
输入文件名	alphadog.in	infinity.in	unknown.in
输出文件名	alphadog.out	infinity.out	unknown.out
每个测试点时限	1.0 秒	1.0 秒	2.0 秒
内存限制	256 MiB	256 MiB	512 MiB
测试点数目	20	20	20
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	alphadog.cpp	infinity.cpp	unknown.cpp
-----------	--------------	--------------	-------------

编译选项

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

### 注意事项（请仔细阅读）

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- 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`。
- 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）。
- 选手提交的程序源文件必须不大于 100KB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 只提供 Linux 格式附加样例文件。
- 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）和其他可能造成不公平的方法。
- 请确认本地编译命令与题面给出的编译命令一致，对于 Linux 选手，可以使用 `alias` 简化命令。

## 机器狗 (alphadog)

### 【题目背景】

机器狗被羊角兽追杀，在机房横冲直撞。

### 【题目描述】

机房可以抽象为一张有  $N$  个点， $M$  条边的连通无向图，这  $M$  条边被机器狗编号为  $1, 2, \dots, M$ 。

机器狗有一个可以规划路线的程序，这个程序如下：

1. 机器狗可以存储集合  $S$ ，变量  $h$ 。初始  $S = \{v\}$ ， $h = 0$ 。
2. 当  $|S| < N$ ，重复执行：
  1. 在仅有一个顶点在集合  $S$  中的边中，找到编号最小的那条。记这条边的编号为  $e$ 。
  2. 将  $e$  不在  $S$  中的那个端点加入集合  $S$ 。
  3. 将  $h$  更新为  $10h + e$ 。
3. 执行结束后，机器狗的屏幕上显示  $h$  对  $10^9 + 7$  取模的值。

羊角兽想要尽早抓到机器狗，为此，她要知道，机器狗从起点  $v = 1, 2, \dots, N$  出发，最终屏幕上显示的值。

### 【输入格式】

从文件 *alphadog.in* 中读入数据。

输入的第一行包含两个整数  $N$  和  $M$ 。

接下来  $M$  行，每行两个整数  $u_i, v_i$ ，描述编号为  $i$  的边。

### 【输出格式】

输出到文件 *alphadog.out* 中。

输出  $N$  行，每行一个整数，第  $i$  行包含机器狗从起点  $i$  出发，最终屏幕上显示的值。

### 【样例 1 输入】

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

**【样例 1 输出】**

```
1 12
2 12
3 21
```

**【样例 2 输入】**

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

**【样例 2 输出】**

```
1 1325
2 1325
3 2315
4 2315
5 5132
```

**【样例 2 解释】**

以机器狗从起点 3 出发为例。

1. 选择 2 号边,  $S = \{3, 4\}$ ,  $h = 2$ 。
2. 选择 3 号边,  $S = \{2, 3, 4\}$ ,  $h = 23$ 。
3. 选择 1 号边,  $S = \{1, 2, 3, 4\}$ ,  $h = 231$ 。
4. 选择 5 号边,  $S = \{1, 2, 3, 4, 5\}$ ,  $h = 2315$ 。

**【样例 3】**

见选手目录下的 *alphadog/alphadog3.in* 与 *alphadog/alphadog3.ans*。

【样例 4】

见选手目录下的 *alphadog/alphadog4.in* 与 *alphadog/alphadog4.ans*。

【样例 5】

见选手目录下的 *alphadog/alphadog5.in* 与 *alphadog/alphadog5.ans*。

【样例 6】

见选手目录下的 *alphadog/alphadog6.in* 与 *alphadog/alphadog6.ans*。

【子任务】

对于所有测试数据保证： $2 \leq N \leq 2 \times 10^5$ ， $N - 1 \leq M \leq 4 \times 10^5$ ，给定的无向图连通。

每个测试点的具体限制见下表：

测试点编号	$N \leq$	$M \leq$	特殊性质
1	$2 \times 10^3$	$2 \times 10^3$	无
2		$4 \times 10^5$	
3			
4	$10^4$		
5			
6			
7			
8	$2 \times 10^5$		$u_i + 1 = v_i$
9			
10			
11			无
12			
13			
14			
15			
16			
17			
18			
19			
20			

# 机器人 (infinity)

## 【题目背景】

机器人要背单词。

## 【题目描述】

机器人的单词书里有很多单词，长度为  $i$  的单词有  $f_i$  个。 $f_i$  的定义式如下：

$$f_i = \begin{cases} a_i & \text{for } 1 \leq i \leq N \\ \sum_{j=1}^n c_j \cdot f_{i-j} & \text{for } i > N \end{cases}$$

机器人充满了正能量，他要为自己制定背单词计划。

具体地，机器人的单词计划可以表示为序列  $\{w_1, w_2, w_3, \dots\}$ ，且满足  $\sum |w| = M$ 。

换言之，相同长度的不同单词和相同单词集合的不同顺序均会被机器人认为是不同的计划。

机器人想知道他有多少制定计划的方案，对 998244353 取模。

## 【输入格式】

从文件 *infinity.in* 中读入数据。

输入的第一行包含两个整数  $N$  和  $M$ 。

接下来一行包含  $N$  个整数，表示  $a_{1\dots N}$ 。

接下来一行包含  $N$  个整数，表示  $c_{1\dots N}$ 。

## 【输出格式】

输出到文件 *infinity.out* 中。

输出一行一个整数，表示方案数对 998244353 取模的值。

## 【样例 1 输入】

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

## 【样例 1 输出】

```
1 10
```

**【样例 2 输入】**

```
1 2 1000000000
2 1 2
3 3 4
```

**【样例 2 输出】**

```
1 168267027
```

**【样例 3】**

见选手目录下的 *infinity/infinity3.in* 与 *infinity/infinity3.ans*。

**【样例 4】**

见选手目录下的 *infinity/infinity4.in* 与 *infinity/infinity4.ans*。

**【子任务】**

对于所有测试数据保证： $1 \leq N \leq 50$ ， $1 \leq M \leq 10^9$ ， $0 \leq a_i, c_i < 998244353$ 。  
每个测试点的具体限制见下表：

测试点编号	$N \leq$	$M \leq$	特殊性质	
1	1	$10^9$	$c_1 = 0$	
2			$a_1 = 1, c_1 = 1$	
3	50	$5 \times 10^3$	无	
4				
5				
6				
7				
8	2	$10^5$	$N = 2, c_1 = c_2 = 0$	
9				
10				
11	50		$10^5$	无
12				
13				
14				
15				
16				
17		$10^9$		
18				
19				
20				

## 机器猫 (unknown)

### 【题目背景】

机器猫爬到了树上。

### 【题目描述】

机器猫所在的树共有  $N$  个结点与  $N-1$  条边，结点从  $1, 2, \dots, N$  编号。

初始时，每个结点上都有一个  $1 \sim N$  的数字，且每个  $1 \sim N$  的数字都只在恰好一个结点上出现。

机器猫想要进行  $N-1$  次删边操作。每次操作，机器猫会选一条未被删去的边，交换这条边所连接的两个结点上的数字，然后这条边将被删去。

在所有边都被删去后，机器猫将数字  $1 \sim N$  所在的结点编号依次排列，得到一个结点编号的排列  $P$ 。

机器猫想要得到字典序最小的排列  $P$ 。

聪明的机器猫可以通过排列反推出操作序列。所以，你只需要告诉机器猫这个排列，而无需输出方案。

### 【输入格式】

从文件 `unknown.in` 中读入数据。

本题输入包含多组测试数据。

第一行一个正整数  $T$ ，表示数据组数。

对于每组测试数据，第一行一个整数  $N$ ，表示树的大小。

接下来一行  $n$  个整数，第  $i$  ( $1 \leq i \leq N$ ) 个整数表示数字  $i$  初始时所在的结点编号。

接下来  $N-1$  行每行两个整数  $u, v$ ，表示一条连接  $u$  号结点与  $v$  号结点的边。

### 【输出格式】

输出到文件 `unknown.out` 中。

对于每组测试数据，输出一行  $N$  个整数，表示最优操作方案下所能得到的字典序最小的排列  $P$ 。

### 【样例 1 输入】

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



```
5 1 4
6 2 4
7 4 5
8 5
9 3 4 2 1 5
10 1 2
11 2 3
12 3 4
13 4 5
14 5
15 1 2 5 3 4
16 1 2
17 1 3
18 1 4
19 1 5
20 10
21 1 2 3 4 5 7 8 9 10 6
22 1 2
23 1 3
24 1 4
25 1 5
26 5 6
27 6 7
28 7 8
29 8 9
30 9 10
```

**【样例 1 输出】**

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

**【样例 2】**

见选手目录下的 *unknown/unknown2.in* 与 *unknown/unknown2.ans*。

### 【样例 3】

见选手目录下的 *unknown/unknown3.in* 与 *unknown/unknown3.ans*。

### 【样例 4】

见选手目录下的 *unknown/unknown4.in* 与 *unknown/unknown4.ans*。

### 【子任务】

对于所有测试数据保证： $1 \leq T \leq 10$ ， $1 \leq N \leq 2 \times 10^3$ ，给出的是一棵树。  
每个测试点的具体限制见下表：

测试点编号	$N \leq$	特殊性质
1	10	无
2		
3	160	树的形态是链
4		
5	$2 \times 10^3$	
6		
7		
8	160	树的形态是菊花
9		
10	$2 \times 10^3$	
11		
12		
13	160	无
14		
15		
16		
17	$2 \times 10^3$	
18		
19		
20		