

IOI 2018 中国国家队选拔赛暨精英赛

CTSC 2018

第一试

时间：2018 年 5 月 7 日 08:30 ~ 13:30

题目名称	假面	暴力写挂	青蕈领主
题目类型	传统型	传统型	传统型
目录	faceless	wronganswer	green
可执行文件名	faceless	wronganswer	green
输入文件名	faceless.in	wronganswer.in	green.in
输出文件名	faceless.out	wronganswer.out	green.out
每个测试点时限	6.0 秒	4.0 秒	3.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	10	20	20
每个测试点分值	10	5	5

提交源程序文件名

对于 C++ 语言	faceless.cpp	wronganswer.cpp	green.cpp
对于 C 语言	faceless.c	wronganswer.c	green.c
对于 Pascal 语言	faceless.pas	wronganswer.pas	green.pas

编译选项

对于 C++ 语言	-O2 -lm	-O2 -lm	-O2 -lm
对于 C 语言	-O2 -lm	-O2 -lm	-O2 -lm
对于 Pascal 语言	-O2	-O2	-O2

假面 (faceless)

【题目背景】

针针是绿绿的好朋友。

【题目描述】

针针喜欢玩一款叫做 DotA (Defense of the Algorithm) 的游戏, 在这个游戏中, 针针会操纵自己的英雄与队友一起对抗另一支队伍。

针针在 DotA 中最喜欢使用的英雄叫做假面 (Faceless), 该英雄有 2 个技能:

- 锁定: 对一名指定的敌方单位使用, 以 p 的概率对该单位造成 1 点伤害 (使其减少 1 点生命值)。

- 结界: 在一片区域施放结界, 让该区域内的所有其他单位无法动弹。

在游戏中, 如果一个单位的生命值降至 0 或 0 以下, 那么该单位就会死亡。

针针操纵假面的水平一般, 因此他决定勤加练习。

现在有 n 个敌方单位 (编号从 1 至 n), 编号为 i 的敌方单位有 h_i 点生命值。

针针已经安排好了练习的计划, 他会按顺序施放 Q 个技能:

- 对于锁定技能: 针针会指定一个敌方单位 id , 并对它施放。由于决定概率系数 p 的因素很多, 因此每次的 p 都不一定相同。

- 特别地, 如果该敌方单位已经死亡, 那么该技能不会造成任何效果。

- 对于结界技能: 针针会希望对 k 个指定的敌方单位施放, 但由于针针并不擅长施放该技能, 因此他只能命中恰好 1 个敌方单位。命中每个存活的敌方单位的概率是相等的 (也就是说已经死亡的敌方单位不会有任何影响)。

- 特别地, 如果这 k 个敌方单位均已死亡, 那么该技能同样不会命中任何敌方单位。

现在, 围观针针进行练习的绿绿想知道:

1. 对于针针施放的每个结界技能, 它命中各敌人的概率分别是多少。
2. 在针针的所有技能施放完毕后, 所有敌方单位剩余生命值的期望分别是多少。

由于绿绿还要围观针针训练, 所以请你帮他解决这两个问题。

为了防止精度误差, 对于所有需要输出的数值, 请输出其在模 998,244,353 意义下的值。

由于结界为假面的终极技能, 因此针针施放该技能的次数不会太多。具体请见【子任务】。

【输入格式】

从文件 *faceless.in* 中读入数据。

- 第 1 行为 1 个正整数 n ，表示敌方单位的数量。
- 第 2 行为 n 个正整数 m_1, \dots, m_n ，依次表示各敌方单位的初始生命值。
- 第 3 行为 1 个非负整数 Q ，表示技能施放的数量。
- 第 4 行至第 $Q+3$ 行，每行描述一个技能，第 $i+3$ 行描述第 i 个技能。
 - 每行的开头为一个整数 op ，表示该技能的种类。
 - 如果 $op = 0$ ，则表示锁定技能。并在此后跟随着 3 个正整数 id, u, v ，表示技能施放的目标为 id ，技能命中的概率为 $p = \frac{u}{v}$ 。（保证 $1 \leq id \leq n$ ， $0 < u \leq v < 998,244,353$ ）
 - 如果 $op = 1$ ，则表示结界技能。并在此后跟随着 1 个正整数 k 表示技能施放的目标数量，随后还有额外的 k 个数 id_1, \dots, id_k 描述技能施放的所有目标。（保证所有 $1 \leq id_i \leq n$ 互不相同）

对于每一行，如果行内包含多个数，则用单个空格将它们隔开。

【输出格式】

输出到文件 `faceless.out` 中。

输出包 $C+1$ 行（其中 C 为结界技能的数量）：

- 前 C 行依次对应每个结界技能，对于每行：
 - 输出 k 个数，第 i 个数表示结界命中敌方单位 id_i 的概率。
- 第 $C+1$ 行输出 n 个数，第 i 个数表示在所有技能施放完毕后，敌方单位 i 剩余生命值的期望值。

对于每一行，如果行内包含多个数，则用单个空格将它们隔开。

对于所有数值，请输出它们对 998,244,353 取模的结果：即设答案化为最简分式后的形式为 $\frac{a}{b}$ ，其中 a 和 b 互质。输出整数 x 使得 $bx \equiv a \pmod{998244353}$ 且 $0 \leq x < 998244353$ 。（可以证明这样的整数 x 是唯一的）

【样例 1 输入】

```

3
1 2 3
6
0 2 1 1
1 1 2
0 2 1 1
0 3 1 1
1 1 2
1 3 1 2 3

```

【样例 1 输出】

```

1
0
499122177 0 499122177
1 0 2

```

【样例 1 解释】

针针按顺序释放如下技能：

1. 对敌方单位 2 释放技能锁定：以 1 的概率对其造成 1 点伤害。
 - 此时 2 号敌方单位必定剩余 1 点生命值。
2. 对敌方单位 2 释放技能结界：（由于 2 号敌方单位尚存活，）必定命中 2 号单位。
3. 对敌方单位 2 释放技能锁定：以 1 的概率对其造成 1 点伤害。
4. 对敌方单位 3 释放技能锁定：以 1 的概率对其造成 1 点伤害。
 - 此时三个敌方单位的生命值一定分别为 1,0,2，敌方单位 2 一定死亡。
5. 对敌方单位 2 释放技能结界：（由于 2 号敌方单位已死亡，）必定不命中任何单位。
6. 对敌方单位 1,2,3 释放技能结界：命中敌方单位 1,3 的概率是相等的，即各位 $\frac{1}{2}$ 。最终，三个敌方单位的剩余生命值一定为 1,0,2。

【样例 2 输入】

```

3
1 1 1
4
0 2 1 2
1 2 1 2
0 3 2 3
1 3 1 2 3

```

【样例 2 输出】

```

249561089 748683265
804141285 887328314 305019108
1 499122177 332748118

```

【样例 2 解释】

对于各结界技能的分析：

1. 第 1 个结界（目标为敌方单位 1, 2）：

- 2 号敌方单位存活的概率为 $\frac{1}{2}$ ，1 号敌方单位必定存活。
- 如果 2 号敌方单位存活，那么结界命中 1, 2 的概率相等，均为 $\frac{1}{2}$ ；如果 2 号敌方单位死亡，那么结界必定命中 1 号敌方单位。
- 因此：命中 1 号敌方单位的概率为 $\frac{1}{2} \times 1 + \frac{1}{2} \times \frac{1}{2} = \frac{3}{4}$ ；命中 2 号敌方单位的概率为 $\frac{1}{2} \times 0 + \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$ 。

2. 第 2 个结界（目标为敌方单位 1, 2, 3）：

- 三个敌方单位存活的概率分别为 $1, \frac{1}{2}, \frac{1}{3}$ 。
- 1, 2, 3 同时存活的概率为 $\frac{1}{6}$ ；只有 1, 2 存活的概率为 $\frac{1}{3}$ ；只有 1, 3 存活的概率为 $\frac{1}{6}$ ；只有 1 存活的概率为 $\frac{1}{3}$ 。
- 因此：命中 1 号敌方单位的概率为 $\frac{1}{6} \times \frac{1}{3} + \left(\frac{1}{3} + \frac{1}{6}\right) \times \frac{1}{2} + \frac{1}{3} \times 1 = \frac{23}{36}$ ；命中 2 号敌方单位的概率为 $\frac{1}{6} \times \frac{1}{3} + \frac{1}{3} \times \frac{1}{2} = \frac{2}{9}$ ；命中 3 号敌方单位的概率为 $\frac{1}{6} \times \frac{1}{3} + \frac{1}{6} \times \frac{1}{2} = \frac{5}{36}$ 。

最终，三个敌方单位的剩余生命值的期望值为 $1, \frac{1}{2}, \frac{1}{3}$ 。

【样例 3】

见选手目录下的 *faceless/faceless3.in* 与 *faceless/faceless3.ans*。

【样例 4】

见选手目录下的 *faceless/faceless4.in* 与 *faceless/faceless4.ans*。

【子任务】

我们记 C 为结界技能的数量。

$n =$	$Q =$	$C =$	测试点编号	u, v	其他限制
5	21	6	1		无
60	199,992	500	2	$u < v$	所有 p 均相等
	23	6	3		所有 $m_i = 1$
	199,994	500	4		
	199,995		5		
	199,996	0	6		
	199,997	500	7	$u = v$	
	200	199,998	1000	8	$u < v$
199,999		9			
200,000		10			

为了优化你的阅读体验，我们把测试点编号放在了表格的中间，请注意这一点。
 对于所有测试点，保证 $n \leq 200$ ， $Q \leq 200,000$ ， $C \leq 1000$ ， $m_i \leq 100$ 。

【提示】

- 第 3 个样例满足测试点 1 的数据规模限制。
- 第 4 个样例满足限制“所有 p 均相等”。
- Q 的个位可以帮助你快速确定测试点的编号。
- 测试点顺序可能与难度无关。

暴力写挂 (wronganswer)

【题目描述】

temporaryDO 是一个很菜的 OIer。在 4 月，他在省队选拔赛的考场上见到了《林克卡特树》一题，其中 $k = 0$ 的部分是求树 T 上的最长链。可怜的 temporaryDO 并不会做这道题，他在考场上抓猫耳挠猫腮都想不出一丝思路。

这时，善良的板板出现在了空中，他的身上发出璀璨却柔和的光芒，荡漾在考场上。“题目并不难。”板板说。那充满磁性的声音，让 temporaryDO 全身充满了力量。他决定：写一个枚举点对求 LCA 算距离的 $k = 0$ 的 $O(n^2 \log n)$ 的部分分程序！于是，temporaryDO 选择以 1 为根，建立了求 LCA 的树链剖分结构，然后写了二重 `for` 循环枚举点对。

然而，菜菜的 temporaryDO 不小心开小了数组，于是数组越界到了一片神秘的内存区域。但恰好的是，那片内存区域存储的区域恰好是另一棵树 T' 。这样一来，程序并没有 RE，但他求 x 和 y 的距离的时候，计算的是

$$\text{depth}(x) + \text{depth}(y) - (\text{depth}(\text{LCA}(x, y)) + \text{depth}'(\text{LCA}'(x, y)))$$

最后程序会输出每一对点对 i, j ($i \leq j$) 的如上定义的“距离”的最大值。

temporaryDO 的程序在评测时光荣地爆零了。但他并不服气，他决定花好几天把自己的程序跑出来。请你根据 T 和 T' 帮帮可怜的 temporaryDO 求出他程序的输出。

【输入格式】

从文件 *wronganswer.in* 中读入数据。

第一行包含一个整数 n ，表示树上的节点个数；

第 2 到第 n 行，每行三个整数 x, y, v ，表示 T 中存在一条从 x 到 y 的边，其长度为 v ；

第 $n + 1$ 到第 $2n - 1$ 行，每行三个整数 x, y, v ，表示 T' 中存在一条从 x 到 y 的边，其长度为 v 。

【输出格式】

输出到文件 *wronganswer.out* 中。

输出一行一个整数，表示 temporaryDO 的程序的输出。

【样例 1 输入】

6

1 2 2

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

【样例 1 输出】

5

【样例 1 解释】

点对 (3,4) 的距离计算为 $3 + 0 - (0 + (-2)) = 5$ 。

【样例 2】

见选手目录下的 *wronganswer/wronganswer2.in* 与 *wronganswer/wronganswer2.ans*。

【子任务】

对于所有数据， $n \leq 366,666$ ， $|v| \leq 2,017,011,328$ 。

详细数据范围见下表，表格中的“无”表示无特殊限制。

测试点编号	$n \leq$	v	T 是一条链	T' 是一条链
1	36	= 1	否	否
2	366			
3	1388	> 0		
4	1999			
5	2666			
6	5666	无		
7	8666			
8	11111			
9	12345			
10	366666	> 0	是	是
11		无		
12		> 0		否
13		无		
14		无	否	是
15		> 0		
16		> 0		
17		无		否
18	否			
19				
20	无			

【提示】

$\text{depth}(p)$ 和 $\text{depth}'(p)$ 分别表示树 T 、 T' 中点 1 到点 p 的距离，这里规定，距离指的是经过的边的边权总和，其中 $\text{depth}(1) = 0$ 。

$\text{LCA}(x, y)$ 和 $\text{LCA}'(x, y)$ 分别表示树 T 、 T' 中点 x 与点 y 的最近公共祖先，即在从 x 到 y 的最短路径上的距离根经过边数最少的点。

青葙领主 (green)

【题目背景】

“也许，我的生命也已经如同风中残烛了吧。”小绿如是说。

【题目描述】

小绿同学因为微积分这门课，对“连续”这一概念产生了浓厚的兴趣。小绿打算把连续的概念放到由整数构成的序列上，他定义一个长度为 m 的整数序列是连续的，当且仅当这个序列中的最大值与最小值的差，不超过 $m - 1$ 。例如 $\{1, 3, 2\}$ 是连续的，而 $\{1, 3\}$ 不是连续的。

某天，小绿的顶头上司板老大，给了小绿 T 个长度为 n 的排列。小绿拿到之后十分欢喜，他求出了每个排列的每个区间是否是他所定义的“连续”的。然而，小绿觉得被别的“连续”区间包含住的“连续”区间不够优秀，于是对于每个排列的所有右端点相同的“连续”区间，他只记录下了长度最长的那个“连续”区间的长度。也就是说，对于板老大给他的每一个排列，他都只记录下了在这个排列中，对于每一个 $1 \leq i \leq n$ ，右端点为 i 的最长“连续”区间的长度 L_i 。显然这个长度最少为 1，因为所有长度为 1 的整数序列都是连续的。

做完这一切后，小绿爬上绿色床，美美地做了一个绿色的梦。

可是第二天醒来之后，小绿惊讶的发现板老大给他的所有排列都不见了，只剩下他记录下来的 T 组信息。小绿知道自己在劫难逃，但是作为一个好奇的青年，他还是想知道：对于每一组信息，有多少个和信息符合的长度为 n 的排列。

由于小绿已经放弃治疗了，你只需要告诉他每一个答案对 998244353 取模的结果。

我们并不保证一定存在至少一个符合信息的排列，因为小绿也是人，他也有可能犯错。

【输入格式】

从文件 *green.in* 中读入数据。

输入的第一行包含两个整数 T, n ，分别表示板老大给小绿的排列个数、以及每个排列的长度。

接下来 T 行，每行描述一组信息，包含 n 个正整数，第 i 组信息的从左往右第 j 个整数 $L_{i,j}$ 表示第 i 个排列中右端点为第 j 个数的最长“连续”区间的长度。

对于每一行，如果行内包含多个数，则用单个空格将它们隔开。

【输出格式】

输出到文件 *green.out* 中。

对于每组信息，输出一行一个整数表示可能的排列个数对 998244353 取模的结果。由于是计算机帮你算，所以我们不给你犯错的机会。

【样例 1 输入】

```
1 3
1 1 3
```

【样例 1 输出】

```
2
```

【样例 2】

见选手目录下的 *green/green2.in* 与 *green/green2.ans*。

【样例 3】

见选手目录下的 *green/green3.in* 与 *green/green3.ans*。

【子任务】

测试点编号	$n \leq$	$T \leq$	特殊性质
1 ~ 2	10	1	无
3 ~ 4	10	100	
5	300	1	$L_{i,j} = j$
6			$L_{i,j} = 1$ 且 $j < n$
7 ~ 8		100	无
9	1000	1	$L_{i,j} = 1$ 且 $j < n$
10 ~ 12		100	无
13 ~ 16	5000		
17 ~ 20	50,000		

对于所有测试数据， $1 \leq T \leq 100$ ， $1 \leq N \leq 50000$ ， $1 \leq L_{i,j} \leq j$ 。

【提示】

本题部分测试点的输入规模较大，请注意读入效率。