

CSP2023模拟赛

题目名称	简单计数	机房惨案	石头游戏	递增路径
题目类型	传统型	传统型	传统型	传统型
可执行文件名	count	network	stone	increase
输入文件名	count.in	network.in	stone.in	increase.in
输出文件名	count.out	network.out	stone.out	increase.out
每个测试点时限	1.0 秒	2.0 秒	1.0 秒	1.0 秒
内存限	512 MiB	512 MiB	512 MiB	512 MiB
子任务/测试点数目	20	3	10	10
是否等分	是	否	是	是

提交源文件程序名

对于C++语言	count.cpp	network.cpp	stone.cpp	increase.cpp
---------	-----------	-------------	-----------	--------------

编译选项

对于C++语言	-lm -O2 -std=c++17
---------	--------------------

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/C++中函数main()的返回类型必须是int，程序正常结束时返回值必须是0。
- 选手提交的程序代码文件请在**个人目录下以及子文件夹内各放一份**。
- 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于题目的内存限制一直。
- 使用std::deque等STL容器时，请注意其内存空间消耗。
- 评测时采用的机器配置为 AMD Ryzen 7 5800H with Radeon Graphics，内存16GiB。上述时限以此配置为准。
- 评测在Windows 10下进行，使用LemonLine进行评测。

简单计数 (count)

【题目描述】

请计算有多少长为 n 的正整数序列 c 满足以下条件。

- $\forall 1 \leq i \leq n, c_i \in [1, m]$ 。
- 对于数字 x ，不能在 c 中某一段连续出现超过 a_x 次。

两种方案不同，当且仅当某个位置上的值不同。

【输入格式】

第一行两个正整数 n, m 。

第二行 m 个正整数，第 i 个表示 a_i 。

【输出格式】

输出一个非负整数，表示答案对 998244353 取模后的结果。

【输入输出样例1】

count.in	count.out
3 3 1 2 3	21

【数据规模与约定】

- 对于 10% 的数据， $n, m \leq 5$;
- 对于 30% 的数据， $n, m \leq 50$;
- 对于 50% 的数据， $n, m \leq 500$;
- 对于 70% 的数据， $n, m \leq 5000$;

对于所有测试数据，保证 $a_i \leq n \leq 5000, m \leq 10^5$ 。

机房惨案 (network)

【题目描述】

在机房中共有 n 台电脑，编号为 $1, 2, \dots, n$ ，由 $n - 1$ 条网线连成一个连通块。现在 你是一个 JCer，由于你太巨了，机房里所有的电脑你都能进入并访问。

定义 JC 某台电脑的难度为这台电脑的编号。现在你会到机房 q 次。每次到机房时，你都会干下面两件事之一：

1. 把编号为 x 的电脑 JC 掉。
2. 从某一台电脑 x 开始，通过网线形成的路径逐个访问**所有**被你 JC 了的电脑。因为 JC 也是要被发现，所以你在从 x 开始访问某一台被你 JC 了的电脑时，一定会沿着它们在树上的最短路径进行访问。在此过程中，你会记录所有这些路径上的电脑的难度值，无论这些电脑是否被 JC 过。因为你很懒，所以你只想知道这些电脑中**最小**的难度值。

现在，你需要对于每个事件 2，输出最小的难度值。

【输入格式】

第一行两个正整数 n, q 。

接下来 $n - 1$ 行，每行两个正整数 u, v ，代表 (u, v) 之间有一条网线。

接下来 q 行，每行格式形如下面两种之一：

1. JC x ，代表你把编号为 x 的电脑 JC 掉了。
2. Query x ，代表你从编号为 x 的电脑访问**所有**被你 JC 了的电脑。

数据保证第一次为 JC。不保证一台电脑仅被 JC 一次。

【输出格式】

输出 q 行，对于每个 Query，输出一行一个正整数表示答案。

【输入输出样例1】

network.in	network.out
5 9	
1 3	
5 3	
2 4	
4 3	3
JC 4	4
Query 5	2
Query 4	1
JC 2	1
Query 5	1
JC 1	
Query 5	
Query 1	
Query 4	

【数据规模与约定】

本题采用**捆绑测试**。

- Subtask 1 (25 pts): $1 \leq n, q \leq 4000$
- Subtask 2 (15 pts): 保证给出的是一条链
- Subtask 3 (60 pts): $1 \leq n, q \leq 10^6$

对于所有测试数据, 保证 $1 \leq n, q \leq 10^6, 1 \leq u, v, x \leq n, u \neq v$ 。

本题输入输出量较大, 请选手选择更为高效的输入输出方式。

石头游戏 (stone)

【题目描述】

miya 玩起了游戏!

- n 堆石头排成一排, 初始的时候, 第 i 堆有 a_i 个石头。
- miya 每次可以选择第 i 堆石头满足该堆石头**恰好**有 i 个石头, 然后拿走所有的石头, 并且将 $1 \sim i - 1$ 的石头数目都 $+1$ 。
- 最后剩下的石头就是 miya 的得分, miya 希望得分**越小越好**。

给定 n, k , 表示初始所有 $a_i \in [0, k]$, 求出所有情况下 miya 的得分之和 $\bmod 10^9 + 7$ 。

【输入格式】

第一行两个正整数 n, k 。

【输出格式】

一行一个数表示答案。

【输入输出样例1】

stone.in	stone.out
5 1	64

【数据规模与约定】

对于所有测试数据, 保证 $n \leq 100, 1 \leq k \leq n$ 。

测试点编号	$n, k \leq$
1 ~ 3	6
4 ~ 5	20
6 ~ 7	40
8 ~ 10	100

递增路径 (increase)

【题目描述】

有一张 n 个点 m 条边的无向图，每条边有一个 $1 \sim m$ 之间的整数边权，且边权互不相同。Alice 和 Bob 在上面玩一个游戏。

首先规定一个起点 s ，起点上有一个棋子，接下来 Alice 和 Bob 轮流操作（Alice 先手）：

- 若当前棋子在点 u 上，则选择一条与 u 相邻的边 (u, v) ，需要满足如果这不是第一次操作，则该边的边权必须大于上一次操作所选择的边的边权，然后把棋子移动到 v 上。如果无法操作，则游戏结束。

Alice 希望游戏进行的轮数尽量多，而 Bob 希望游戏进行的轮数尽量少。对于每个 $s = 1, 2, \dots, n$ 求出，若双方都采用最优策略，游戏会进行多少轮。

【输入格式】

第一行：两个整数 n, m 。

接下来 m 行：每行三个整数 u, v, w 表示一条连接 u, v 的边，边权为 w 。

【输出格式】

一行 n 个整数，分别表示以每个点作为起点的答案。

【输入输出样例1】

increase.in	increase.out
4 6 3 1 2 2 3 1 4 1 5 1 4 3 2 1 6 1 3 4	3 3 2 2

【输入输出样例2】

increase.in	increase.out
6 12 5 3 2 4 5 4 4 1 7 4 2 8 3 6 3 6 6 10 1 1 6 5 2 11 3 6 9 4 2 12 4 6 5 3 4 1	4 2 3 3 3 3

【数据规模与约定】

- 对于前 20% 的测试数据，满足 $n, m \leq 20$ 。
- 对于前 50% 的测试数据，满足 $n, m \leq 5000$ 。

对于所有测试数据，保证 $1 \leq n \leq 2 \times 10^5, 1 \leq m \leq 5 \times 10^5, 1 \leq u, v \leq n, 1 \leq w \leq m$ 。