

NOIP2023 模拟赛 R1

— By YeahPotato

2023.9

题目名称	原子	旅行计划	禁止套娃	简单题
题目类型	传统型	传统型	传统型	传统型
源程序文件名	atom.cpp	plan.cpp	nest.cpp	easy.cpp
输入文件名	atom.in	plan.in	nest.in	easy.in
输出文件名	atom.out	plan.out	nest.out	easy.out
每个测试点时限	2s	4s	2s	3s
内存限制	512MB	512MB	512MB	1GB
测试点数目	20	10	10	10
测试点是否等分	是	是	是	是
是否有 spj	是	否	否	否

开 O2、无限栈、C++ 11 以上。

请使用 lemon 测试，否则 spj 无法正常工作。

原子

• 题目背景

- 根据玻尔理论，电子只能在特定的轨道上运动，因此，原子的能量也只能取一系列特定的值，这些量子化的能量值叫做能级，能量最低的状态叫做基态，其他的状态叫做激发态。

小 Z 最近受到了高中物理的毒打，他知道，处于第 n 能级（基态为第一能级）的原子向下跃迁时理论上能辐射出 $\binom{n}{2}$ 种不同能量的光子。而同时身为 Oler 的他在某一天突发奇想：假设任意两个能级的能量差都不同，那么最少需要多少个原子向下跃迁，就能辐射出所有 $\binom{n}{2}$ 种光子呢？

小 Z 通过打表发现他的假设是错误的，因为考虑到氢原子的能级公式 $E_n = E_1/n^2$ ，而：

$$\frac{1}{5^2} - \frac{1}{6^2} = \frac{1}{9^2} - \frac{1}{90^2}$$

但是这在理论上不失为一个有趣的问题。他将问题发给了 YeahPotato 和你，你需要在 YeahPotato 口胡完之前写出代码。

• 题目描述

给定 n 。你需要准备尽量少的原子，它们初始都处于第 n 能级。对于一个第 i 能级的原子，你可以控制它向第 $[1, i) \cap \mathbb{Z}$ 中的某个能级跃迁。原子可以跃迁任意次，且最终不一定要到达基态。

要求每种形如“从第 i 能级向第 j 能级跃迁” ($1 \leq j < i \leq n$) 的情况都至少出现一次。

• 输入格式

一行一个正整数 n 。

• 输出格式

第一行输出一个整数 m ，表示最少需要几个原子。

接下来 m 行每行表示一个原子的操控过程，第一个整数 c 表示当前原子的跃迁次数，接下来 c 个正整数 $d_1 \dots d_c$ 依次表示每次跃迁降低的能级数，你需要保证 $\sum d_i < n$ 。例如， $n = 8, c = 3, d = [2, 3, 1]$ ，那么这个原子的能级变化为 $8 \rightarrow 6 \rightarrow 3 \rightarrow 2$ 。

你需要保证 $\sum c \leq 6 \times 10^6$ 。可以证明存在一组方案满足该条件且使用最少的原子。

• 样例输入 1

```
3
```

• 样例输出 1

```
2
2 1 1
1 2
```

• 样例输入 2

```
4
```

• 样例输出 2

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

• 评分方式 & 数据范围

在一个测试点中，如果你正确输出了最少的原子数，可以得到该测试点 20% 的分数。如果你正确输出了最少的原子数和任意一个满足所有条件的方案，可以得到该测试点 100% 的分数。否则无法得分。

注意，即使你只希望获得 20% 的分数，也需要输出 m 行合法的方案，也就是说，当且仅当你输出的方案不能覆盖所有 $\binom{n}{2}$ 种情况，但满足其他所有格式条件，你才能获得 20% 的分数。一个推荐的方式是输出 m 行 0。

测试点编号	$n \leq$
1 ~ 4	8
5 ~ 8	25
9 ~ 14	100
15 ~ 20	2000

对于 100% 的数据， $2 \leq n \leq 2000$ 。

• 提示

所有测试点的输入 n 互不相同。

本题输出量较大，请使用较快的输出方式。

旅行计划

• 题目描述

YeahPotato 到自然公园去旅行！

自然公园的景点可以抽象成 n 个节点，景点之间由固定的班车连接，班车可以双向地运送旅客，也就是说班车可以抽象成边。一共有 $n - 1$ 对景点之间有班车，任意两个景点之间都可以互相到达。不过这 $n - 1$ 对景点中有一部分对的两点间可能有 > 1 部班车（且它们的实地路线除端点外略有不同），也就是说，自然公园可以抽象成一棵有重边的树。

YeahPotato 考虑坐直升机先到达一个景点（起点），然后走过一些边到达另一个景点（终点，可以与起点重合）后再坐直升机离开。本着“风景在路上”的理念，YeahPotato 希望看到尽量多的风景，也就是走尽量多的边。他不希望重复经过一条边（同一条边的两个方向只能走一个），不过可以重复经过点。

给定自然公园的地图，求 YeahPotato 最多能经过几条边。

• 输入格式

本题有多组数据。第一行一个整数 T 。

每组数据第一行一个整数 n 。

接下来 $n - 1$ 行，第 $i + 1$ 行三个整数 u_i, v_i, w_i ，分别表示一对有边的景点以及它们之间班车（重边）数。

保证输入构成一棵树。

• 输出格式

输出 T 行，每行一个整数依次表示每组输入的答案。

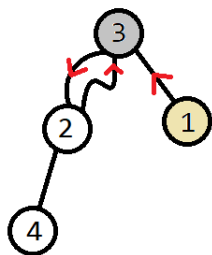
• 样例输入 1

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

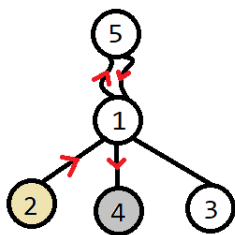
• 样例输出 1

```
3
4
10
```

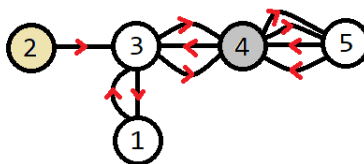
• 样例解释 1



1→3→2→3



2→1→5→1→4



2→3→4→5→4→3→1→3→4→5→4

• 样例输入 2

见下发文件夹中的 plan/plan2.in。

• 样例输出 2

见下发文件夹中的 plan/plan2.ans。

• 样例输入 3

见下发文件夹中的 plan/plan3.in。

• 样例输出 3

见下发文件夹中的 plan/plan3.ans。

• 样例输入 4

见下发文件夹中的 plan/plan4.in。

• 样例输出 4

见下发文件夹中的 plan/plan4.ans。

• 数据范围

记 $\sum w$ 表示该测试点的单组数据内 w 之和, $\sum n$ 表示该测试点内所有 n 之和。

测试点 1 满足 $n, \sum w \leq 7, T \leq 100$ 。

测试点 2 满足 $n, \sum w \leq 18, T \leq 100$ 。

测试点 3, 4, 5 满足 $u_i = i, v_i = i + 1$ 即原树为一条链。

测试点 3, 7, 8 满足 $n \leq 3000, \sum n \leq 15000$ 。

测试点 4, 7, 9 满足 $w_i \geq 2$ 。

测试点 6 满足 $n \leq 300, \sum n \leq 1500$ 。

测试点 9 满足 $n \leq 10^5, \sum n \leq 3 \times 10^5$ 。

测试点 10 无额外限制。

对于 100% 的数据, $1 \leq T \leq 10^4, 1 \leq n \leq 10^6, 1 \leq \sum n \leq 3 \times 10^6, 1 \leq w_i \leq 10^9$ 。

- 提示

本题输入量较大, 请使用较快的输入方式。

禁止套娃

• 题目背景

YeahPotato 喜欢子序列和套娃。

• 题目描述

定义从序列 a 到序列集合 S 的映射 $f(a) = S$ 为, a 的所有子序列组成的不可重集合, 即相同的子序列在其中只考虑一次。空序列是任何序列的子序列。

给定 n 以及序列 $a_1 \dots a_n$, 求:

$$\sum_{b \in f(a)} |f(b)| \bmod (10^9 + 7)$$

• 输入格式

第一行一个整数 n 。

第二行 n 个整数 a_1, \dots, a_n 。

• 输出格式

一行一个整数, 表示答案。

• 样例输入 1

```
3
1 2 1
```

• 样例输出 1

```
23
```

• 样例解释 1

$f([1, 2, 1]) = \{\[], [1], [2], [1, 1], [1, 2], [2, 1], [1, 2, 1]\}$ 。

这些子序列的本质不同子序列数依次为 1, 2, 2, 3, 4, 4, 7, 和为 23。

• 样例输入 2

```
10
1 1 4 5 1 4 1 1 4 5
```

- 样例输出 2

13566

- 样例输入 3

见下发文件夹中的 nest/nest3.in。

- 样例输出 3

见下发文件夹中的 nest/nest3.ans。

- 数据范围

对于前 10% 的数据, $n \leq 10$ 。

对于前 30% 的数据, $n \leq 20$ 。

对于前 60% 的数据, $n \leq 400$ 。

对于另 20% 的数据, $a_i \leq 2$ 。

对于 100% 的数据, $1 \leq a_i \leq n \leq 5000$ 。

简单题

• 题目描述

对于 ≥ 2 的整数 m 和正整数 a , 定义 $f_m(a)$ 为 $[0, m)$ 内的整数 x 的个数, 满足 $x^a \equiv 1 \pmod{m}$, 且对于任何一个整数 $a' \in [1, a)$ 都有 $x^{a'} \not\equiv 1 \pmod{m}$ 。也就是说, $f_m(a)$ 等于模 m 意义下阶为 a 的剩余类个数。

YeahPotato 现在给你了若干个 m , 他希望你对于每个 m , 回答最大的 $f_m(a)$, 即 $\max_{a \in \mathbb{N}_+} \{f_m(a)\}$ 。特殊地, 当 $m = 1$ 时, 你只需要回答 1 即可。

YeahPotato 将会用如下方式给出这些 m : 初始时 $m = 1$, 每次他会将 m 乘上或者除以一个质数 (他保证 m 时刻为正整数), 你则要在每次修改后回答对应的最大值 $\text{mod } 998244353$ 。

• 输入格式

第一行一个正整数 q , 表示修改次数。

接下来 q 行每行先是一个符号 $+$ 或 $-$, 然后一个正整数 p , 保证 p 为质数。符号与数之间不带空格, 也就是可以直接当作有符号整数读入。若符号为 $+$ 则表示将 m 乘上 p , 否则表示将 m 除以 p 。

• 输出格式

输出 q 行, 依次表示每次修改后的答案。

• 样例输入 1

```
4
+2
+3
+2
-3
```

• 样例输出 1

```
1
1
3
1
```

• 样例解释 1

以第三次修改后的 $m = 12$ 为例。注意有些数的幂次永远不可能等于 1, 我们就将它忽略。

使得 $1^a \equiv 1 \pmod{12}$ 的最小正整数 a 为 1。

使得 $5^a \equiv 1 \pmod{12}$ 的最小正整数 a 为 2。

使得 $7^a \equiv 1 \pmod{12}$ 的最小正整数 a 为 2。

使得 $11^a \equiv 1 \pmod{12}$ 的最小正整数 a 为 2。

因此 $f_{12}(1) = 1, f_{12}(2) = 3$, 其余的 $f_{12}(a)$ 值均为 0。

- 样例输入 2

```
9
+5
+7
+3
+3
+2
-5
+7
-3
+23333
```

- 样例输出 2

```
2
8
16
64
64
24
144
36
528768
```

- 样例输入 3

见下发文件夹中的 easy/easy3.in。

- 样例输出 3

见下发文件夹中的 easy/easy3.ans。

- 样例输入 4

见下发文件夹中的 easy/easy4.in。

- 样例输出 4

见下发文件夹中的 easy/easy4.ans。

• 数据范围

测试点编号	$q \leq$	$m \leq$	特殊性质
1	3000	300	
2	50	30000	
3	5×10^5	30000	
4	5×10^5	10^6	
5	2×10^5	10^9	
6	5×10^5		AB
7	300		A
8	3000		A
9	5×10^5		A
10	5×10^5		

上表中 $m \leq M$ 表示任何时刻 m 都 $\leq M$ ，留空表示无限制。

特殊性质 A：输入只有正数。

特殊性质 B：输入除了的第一行外，其余行均相同。

对于 100% 的数据， $1 \leq q \leq 5 \times 10^5, 2 \leq p \leq 10^7$ ，保证 p 为质数。