

FOC-PINC 模拟赛

cdqz

10 月 16 日

一、题目概况

题目名称	路哥	密电	战争	送信
子目录名	luge	telegram	warfare	letter
源文件名	luge.cpp	telegram.cpp	warfare.cpp	letter.cpp
可执行文件名	luge	telegram	warfare	letter
输入文件名	luge.in	telegram.in	warfare.in	letter.in
输出文件名	luge.out	telegram.out	warfare.out	letter.out
每个测试点时限	1s	1s	1s	1s
试题总分	100	100	100	100
是否有部分分	否	否	否	否
题目类型	传统	传统	传统	传统
内存限制	256MB	256MB	256MB	256MB

二、注意事项

1. 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。
2. 评测环境为 64 位 Ubuntu 系统，编译选项：-lm -O2 -DONLINE_JUDGE。
3. 程序可使用的栈空间大小与该题内存空间限制一致。
4. 个人目录名请使用英文字母、数字、下划线、短横，不要使用中文字符或其他特殊字符，以免被评测机忽略。
5. 个人目录下不要建立子目录。
6. 题目按照故事顺序排序，不按照难度排序。

路哥

(luge.cpp)

【问题描述】

众所周知，曾经的杨吞天是个霸气威风的大男孩，他心中一直住着一位女生。

9080 年，年迈的杨吞天突然想起了自己当年为了追求 Luge 所作出种种傻事，他从自己的杂物柜中翻出了一张草稿纸，上面密密麻麻的点竟然构成了一棵树，他仔细回想，终于想起了那段往事。

当年，Luge 是一位潇洒的奇女子，她的身上常常有着各式各样的花朵，她为了响应杨吞天的追求，对他承诺：只要他能在小镇中捡到她留下的 k 朵花，她就嫁给杨吞天。杨吞天为了实现自己日思夜想的梦想，特地在草稿纸上画出了小镇的模样以更好地安排自己的路线。他的捡花规则如下：

- 小镇的形态是一棵树，每个结点处都有一些花朵。xx 学校的位置在 1 号点，杨吞天会从这里出发去捡花朵，不断地沿着树边走，直到将他能走到的结点的所有花朵都捡起来；
- 杨吞天为了更大概率地追求 Luge，他在测试的前一天去请教了一位懂得仙术的大神青蛙人，青蛙人说：“爱情的事情要靠缘分，我随机地帮你断掉一些道路，这样可以避免你鬼迷心窍将所有花朵都捡了。”于是青蛙人会随机断掉一些边（当然有可能一条边都不断，每条边被断掉的概率为 0.5 且相互独立）将杨吞天的路线破坏掉。

但是就当他信心满满地准备参加测试的时候，他却去参加 IOI 了！现在他想起了这件事来找到了你，你作为一位大学者，被杨吞天强迫着打破爱情的禁忌告诉他当年成功追求到 Luge 的概率。

【输入格式】

从文件 luge.in 中读入。

第一行两个正整数 n, k ，分别表示小镇的结点数和 Luge 承诺的花朵数。

接下来一行 n 个非负整数，第 i 个数表示第 i 号结点的花朵数。

接下来 $(n - 1)$ 行，每行两个正整数表示一条树边，描述小镇的连通状况。

【输出格式】

向文件 luge.out 中输出。

输出一行一个非负整数表示杨吞天追求到 Luge 的概率，答案对 998244353 取模。

【输入输出样例 1】

luge.in	luge.out
3 1 1 1 1 1 2 2 3	499122177

见选手目录下的 luge/luge1.in 和 luge/luge1.ans。

【输入输出样例 1 说明】

断掉第 1,2 条边或者只断第 1 条边均可恰好获得 1 朵花朵，这两种情况的概率的和为 0.5。

【输入输出样例 2】

luge.in	luge.out
9 4 0 0 1 4 1 2 0 4 7 1 2 2 5 3 1 3 4 2 6 4 8 3 7 8 9	967049217

见选手目录下的 luge/luge2.in 和 luge/luge2.ans。

【输入输出样例 3】

见选手目录下的 luge/luge3.in 和 luge/luge3.ans。

【数据规模与约定】

- 对于 20% 的数据，满足 $n \leq 20$, $k \leq 100$;
- 对于 40% 的数据，满足 $n, k \leq 300$;
- 对于 60% 的数据，满足 $n, k \leq 1000$;
- 另有 10% 的数据，满足树的每个结点最多只有两条树边与之相连。

对于 100% 的数据，满足 $1 \leq n, k \leq 5000$ ，树的每个结点处的花朵个数不超过 5000。

密电

(telegram.cpp)

【问题描述】

为了争夺大陆上的霸权地位，L 国发起的大战席卷了整个大陆。在这场旷日持久的大战中，情报战成为极为重要的一环。作为 L 国情报处的军人，杨吞天的主要任务就是截获敌方的密电然后破解，从而得到情报。

情报的内容包含 n 个无序的正整数，敌方加密的方法是将这 n 个数两两相加从而得到 $\frac{n(n-1)}{2}$ 个正整数。现在杨吞天截获了这个密电得到了这 $\frac{n(n-1)}{2}$ 个正整数，他的任务是破解密电还原这 n 个正整数。

现在战事紧急，战况瞬息万变，杨吞天必须尽快破解密电！注意，对于一个密电，解密后可能会有多个对应的情报，为了不错过所有的情报，杨吞天必须破解出所有可能的情报。

【输入格式】

从文件 telegram.in 中读入。

输入共两行。

第一行包含一个正整数 n ，表示这个情报的长度。

第二行包含 $\frac{n(n-1)}{2}$ 个以空格分隔正整数，表示加密后的密电。

【输出格式】

向文件 telegram.out 中输出。

输出包括若干行。

第一行一个正整数 S ，表示可能的情报个数。注意可能的情报个数可能为 0。

接下来 S 行，每行 n 个以单个空格分隔正整数，表示破解后的情报内容。数字从小到大输出，情报的顺序按字典序从大到小输出。我们称一个情报比另一个情报字典序大，当且仅当存在一个小于等于 n 的正整数 k ，两个情报前 $(k-1)$ 小的数都分别相同且前者第 k 小的数大于后者第 k 小的数。

【输入输出样例 1】

telegram.in	telegram.out
4	1
3 5 4 7 6 5	1 2 3 4

见选手目录下的 telegram/telegram1.in 和 telegram/telegram1.ans。

【输入输出样例 1 说明】

显然 1, 2, 3, 4 是一组可行解:

$$3 = 1 + 2, \quad 5 = 1 + 4, \quad 4 = 1 + 3, \quad 7 = 3 + 4, \quad 6 = 2 + 4, \quad 5 = 2 + 3.$$

可以发现这也是唯一解。

【输入输出样例 2】

telegram.in	telegram.out
4	2
11 17 21 12 20 15	4 7 8 13
	3 8 9 12

见选手目录下的 telegram/telegram2.in 和 telegram/telegram2.ans。

【输入输出样例 3】

见选手目录下的 telegram/telegram3.in 和 telegram/telegram3.ans。

【数据规模与约定】

- 对于占总分 10% 的子任务 1, 满足 $n \leq 10$, 加密后的数的小于等于 20;
- 对于占总分 20% 的子任务 2, 满足 $n \leq 50$, 加密后的数的小于等于 20;
- 对于占总分 20% 的子任务 3, 满足 $n \leq 50$, 加密后的数的小于等于 400;
- 对于占总分 50% 的子任务 4, 无特殊性质。

对于 100% 的数据, 满足 $3 \leq n \leq 500$, 加密后的数的小于等于 2×10^8 。

战争

(warfare.cpp)

【问题描述】

大陆上有 n 个国家，每个国家都由一个或更多的部族组成，并且在同一个部族中的人们只会在同一座国家中生活。

L 国发起的大战包含多场战争，一场战争会涉及恰好两个不同的国家。具体地，我们假设参与一场战争的两个国家 A, B 在战争前分别由 x, y 个部族组成，那么：

- 国家 A 有 p 的概率赢得这场战争，并在战争后收服国家 B 的一个部族成为一个有 $(x + 1)$ 个部族的国家，而国家 B 则会分裂为 $(y - 1)$ 个国家，且每个国家都只有一个部族；
- 国家 B 有 p 的概率赢得这场战争，并在战争后收服国家 A 的一个部族成为一个有 $(y + 1)$ 个部族的国家，而国家 A 则会分裂为 $(x - 1)$ 个国家，且每个国家都只有一个部族；
- 还另外存在 $1 - 2p$ 的概率两败俱伤，国家 A 会分裂为 x 个国家，国家 B 会分裂为 y 个国家，且每个国家都只有一个部族。

其中 $p = s/t$ 是一个可观测的常数，且对于任意两个国家之间的战争，这个值都相等。

为了方便起见，我们假设任意一个时刻都最多只会有一场战争在进行，且战争的发生是完全随机的。即，在一场战争结束后才有可能开始另外一场战争，并且在一个即将发生战争的时刻，所有国家参与这场战争的概率是相等的。

杨吞天知道现在大陆上国家的个数 n 与第 i 个国家的部族数 a_i ，他又依靠高智能数据分析得知了影响战争胜率的因子 p 。他想要估算大战结束的时间，所以他希望能够知道在大陆上只剩一个国家前总共经历的战争的次数的期望。由于杨吞天不喜欢分数与大数，所以他只关心这个期望对 $10^9 + 7$ 取模的结果。

【输入格式】

从文件 warfare.in 中读入。

由于本题的数据范围较大，部分测试点的 a_i 将在程序内生成。

输入的第一行为一个整数 T ，表示输入方式。若 $T = 0$ ，则该测试点的 a_i 直接给出；若 $T = 1$ ，则该测试点的 a_i 将特殊生成。

第二行三个以空格分隔的正整数 n, s, t 。 n 表示现在大陆上国家的个数， $p = s/t$ 为一个影响战争胜率的因子，其具体意义见【问题描述】。

若 $T = 0$ ，则第三行为 n 个以空格分隔的正整数，第 i 个数 a_i 表示现在大陆上第 i 个国家的部族数。

若 $T = 1$ ，则第三行六个以空格分隔的非负整数 m, x, y, z, b_1, b_2 。接下来 m 行中，第 i 行包含三个以空格分隔的正整数 q_i, l_i, r_i 。其意义见下：

若 $n > 2$ ，则 $\forall 3 \leq i \leq n, b_i = (x \times b_{i-1} + y \times b_{i-2} + z) \bmod 2^{64}$ 。

令 $q_0 = 0$ ，对于所有 $1 \leq j \leq m$ ，若下标值 i 满足 $q_{j-1} < i \leq q_j$ ，则有

$$a_i = (b_i \bmod (r_j - l_j + 1)) + l_j.$$

请注意上述数据生成方式仅是为了减少输入量大小，标准算法不依赖于该生成方式。但部分子任务可能会依赖于上述数据生成方式。

【输出格式】

向文件 `warfare.out` 中输出。

输出一行一个非负整数，表示这片大陆上只剩下一个国家前总共经历的战争的次数的期望对 $10^9 + 7$ 取模的值。

【输入输出样例 1】

warfare.in	warfare.out
0	500000005
2 1 3	
1 1	

见选手目录下的 `warfare/warfare1.in` 和 `warfare/warfare1.ans`。

【输入输出样例 1 说明】

每次战争都有 $2/3$ 的概率使仅有的两个部族联合为一个国家，所以期望 $3/2$ 场战争后大陆上将只剩下一个国家。

【输入输出样例 2】

warfare.in	warfare.out
0	1231135
5 6 7 8	
9 10 11 12 13	

见选手目录下的 `warfare/warfare2.in` 和 `warfare/warfare2.ans`。

【输入输出样例 3】

见选手目录下的 `warfare/warfare3.in` 和 `warfare/warfare3.ans`。

【数据规模与约定】

- 对于占总分 20% 的子任务 1, 满足 $T = 0$, $\sum_{i=1}^n a_i \leq 13$;
- 对于占总分 20% 的子任务 2, 满足 $T = 0$, $n \leq 1000$, $2s = t$;
- 对于占总分 20% 的子任务 3, 满足 $T = 0$, $n \leq 3 \times 10^5$;
- 对于占总分 20% 的子任务 4, 满足 $T = 1$, $r_i = 1$;
- 对于占总分 20% 的子任务 5, 满足 $T = 1$ 。

对于 100% 的数据, 满足 $T \in \{0, 1\}$, $2 \leq n \leq 4 \times 10^7$, $0 < 2s \leq t < 10^9 + 7$, $1 \leq a_i \leq 10^{18}$ 。对于 $T = 1$ 的测试点, 满足 $1 \leq m \leq 10^5$, $0 \leq x, y, z, b_1, b_2 < 2^{64}$, $1 \leq q_i \leq n$, $q_m = n$, $q_i < q_{i+1}$, $1 \leq l_i \leq r_i \leq 10^{18}$ 。

送信

(letter.cpp)

【问题描述】

经过长年累月的征战，这场空前绝后的大战以 L 国的战败告终了。

战后的大陆满目疮痍，L 国更是到了崩溃的边缘。这个拥有 n 个城市国家，为了保证国内交通的连通性，仅仅只修建了 $(n - 1)$ 条分别连接两个不同城市的双向道路。

根据战后的国际公约，L 国的军队被解散了。前军人杨吞天随 L 国征战多年后不得不回到了自己的家乡。所幸，杨吞天家乡的一家邮局为杨吞天提供了工作。

因为大战刚刚结束，整个国家动荡不安，许许多多的人身处异乡想要与故乡人保持联络。这就是大概是邮政公司最最繁忙的时期。

杨吞天现在就要完成这个工作，具体来说，邮政局的头子会给他发一张单子，上面密密麻麻地填满了城市的名字。其中两个为一组，代表这两个城市的两个人在互相传信。

为了完成这些任务，邮政局给杨吞天安排了一路线，每次从 x_i 号城市出发到 y_i 号城市。为了尽可能快地完成任务，杨吞天并不想走回头路，于是他只会走唯一的简单路径。在行走的过程中，一旦到达某个城市，杨吞天会检查自己的手上有没有要送到这个城市的信并送出。杨吞天还会检查这个城市有没有可以拿的信，他会拿上这个城市送往其他城市的信然后继续出发。

现在，对于邮政局的安排的每次旅行，杨吞天想知道能送出多少信。注意，对于每一对互相送信的人，杨吞天送走信后这个任务并不会消失，也就是说当杨吞天下一次抵达这些城市时，仍可以送信。还有，杨吞天重新开始旅行的时候，他手上的信件会消失，也就是说，当杨吞天重新开始旅行的时候，他是一封信都没有的。

然而，随着国内局势的稳定，写信的人越来越多，在杨吞天旅行的过程中，邮政局还会发来电报安排一些新的任务。

【输入格式】

从文件 letter.in 中读入。

第一行包括三个以空格分隔正整数 n, m, q ，分别代表城市数、最初安排的任务数和邮政局的命令数。

接下来有 $(n - 1)$ 行，每行两个以空格分隔正整数 u_i, v_i ，表示有一条连接 u_i 号城市和 v_i 号城市的道路。

接下来 m 行，每行两个以空格分隔正整数 a_i, b_i ，表示 a_i 号城市的某个人和 b_i 号城市的某个人想要互相传信。

接下来 q 行，每行三个以空格分隔正整数 t_i, x_i, y_i ：

- 若 $t_i = 1$ ，表示一组询问，代表邮政局安排了一次从 x_i 号城市到 y_i 号城市的旅行。
- 若 $t_i = 2$ ，则表示邮政局又发来了新的送信任务，代表 x_i 号城市的某个人和 y_i 号城市的人互相传信。

【输出格式】

向文件 letter.out 中输出。

输出若干行，对于每次询问输出一行，表示这一次旅行送出的信件数。

【输入输出样例 1】

letter.in	letter.out
7 2 5	1
1 2	0
2 3	2
2 5	3
1 4	
4 6	
6 7	
1 2	
2 6	
1 3 1	
1 1 7	
1 5 7	
2 1 4	
1 5 7	

见选手目录下的 letter/letter1.in 和 letter/letter1.ans。

【输入输出样例 2】

见选手目录下的 letter/letter2.in 和 letter/letter2.ans。

【输入输出样例 3】

见选手目录下的 letter/letter3.in 和 letter/letter3.ans。

【数据规模与约定】

- 对于 10% 的数据，满足 $n, m, q \leq 10$ ；

- 对于 30% 的数据，满足 $n, m, q \leq 5000$;
- 另有 20% 的数据，满足 L 国的每个城市至多只有两条道路相连;
- 另有 20% 的数据，满足 $t_i = 1$ 。

对于 100% 的数据，满足 $1 \leq n, m, q \leq 10^5$ ，保证 L 国内的任意两个城市可以通过现在修建的道路到达，保证 $a_i \neq b_i$ ， $x_i \neq y_i$ 。