《某科学的超电磁炮》系列
solution

Mys_C_K

- 超炮T完结啦！
- 啥时候能出超炮F呢？
- 学园都市七名level5剩下5名表示很不满：同是level5为啥我们就不能出某科学系列？（那当然是因为你们人气低啊（笑））
- 于是就把他们出了出来（上条当麻&蓝悦花&麦野沈利：？？？）

sogitta

- 本题内核：教人打表/找规律/写更优的暴力
- 实际上大概能猜到答案不会很大，又显然有一个O(ans!)或者O(2^ans)级别的做法。
- 那就写写看，根据实现细节的处理，能跑到60~100的样子。不过实际上实现优秀可以跑到120，只不过会超1s就是了。
- 于是打个表就能有60了。

- 如果你没啥梦想写了个O(ans!)就走了的话我估计以你能跑的数据范围很难看出什么规律。

- 不过这种看上去就没啥思路的数学问题可以考虑花几分钟找找规律。
- 不过这需要你打一个尽量快的暴力。
- 本题状压一下大约能跑到80，观察一下就能观察到如下规律。

$\boxed{2|n}$

- 当 $k(k+1)<=n<k(k+2)$，答案是 $2k-1$
- 当 $k(k+2)<=n<(k+1)(k+2)$，答案是 $2k$。

k

1111222…33…

- 证明？非常复杂而且实际上和OI没有什么关系。
- 就不讲了，过程大概就是讨论+构造。

- 类似于NOIP2017D1T1《小凯的疑惑》和NOIP2018D2T2《（忘了叫什么名字了）》，都是属于虽然出题人的意图不是让你找规律但是大部分人其实并不会std，但由于问题特殊性可以找规律。
- 不过类似后者，场上写一个一般的暴力是看不出所有规律的。
- 所以需要权衡一下到底要不要写一个不能多得分的暴力找规律。

touma

- 本题内核：卡常。

$\sum i$

$1^2+\dots+n^2=\frac{1}{3}n^3$

$\frac{1}{2}n^2$

$i=1\dots n$

$j=i\dots n$

(x,y)

$(0,0)$

$g_1(i,x,y)$

$i\le t-1\ (2k-2)$

$f_1(i,(x,y))$：恰好i步第一次到x,y的方案数

$(0,0)$

$g_2(i,x,y)$：从(x,y)出发

$f(i,x,y)$

- 显然考虑一个点在多少路径被经过了至少k次。

- 那么对每个点就处理，恰好走x步第一次到它的方案数、从这个点出发恰好走x步回到自己、从这个点出发恰好走x步第k次回到自己。然后计算即可。

- 这看起来复杂度爆炸，但实际上想一下就会发现常数小的可怜，比如k很大的时候每一部分路径长度不会很长，处理dp数组的时候，算走x步的情况只要考虑到那个点距离是x的点即可，不用考虑全屏。

$O(n)$ $i_1 + i_2 \le n$ $f_1(i_1, \ldots)$ $f_4(i_2, \ldots)$

$O(n^4)$ $t - i_1 - i_2$

② f_3

- 再比如一个点距离原点远的时候用于绕圈就不会很多。

$D = \Sigma 00$

- 反正加上一些常数优化就能让代码常数显著变小就能过了。

① $a = (a+b) \% P$ $\times$

$O(n^5)$

$a += b - (a > b ? a - P : 0)$

$a += (a-b)$, $if(a<0)\ a += p.$

$3n^5 \approx$ 也不是很大 $2e7$

$P \sim 1e9$

$ans = (ans + (ll) a_i b_i) \% P$

shokuhou

- 本题内核：这是一个正常的题

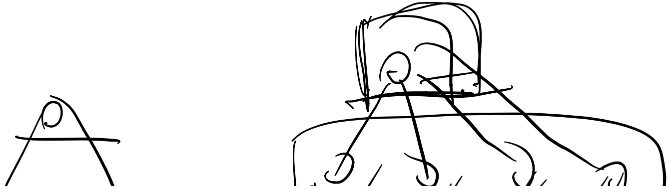
- 本次比赛唯一正常题目。

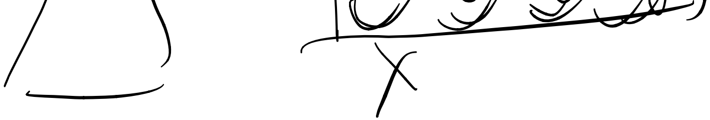
$f(x,i)$

- k<=3：可以树dp，难写程度指数级增长。
- t=1：几乎是个星星。爆搜这8个点怎么选即可。
- t=2，k=1：如果是星星（1的度数是1）就做完了。
- 否则肯定选1最优，然后选和1相连的一些点把剩下的覆盖掉即可。

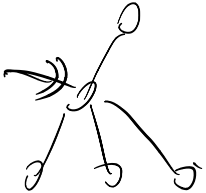
- 100%：仍可以树dp。
- 我有一个绝妙的做法，但是编辑器的空间太小了，我写不下。
- 这不写死人我把御坂御坂的名字前两个字和后两个字对调着念。
- （御坂御坂：？？？

127, 2





- 考虑贪心。
- 显然看起来选深度小的点更优，但不总成立。



- 反过来考虑，从深度大的点开始，遇到没有被覆盖的点，就找他的k级祖先选了，可以发现这样是正确的。

- 考虑怎么维护，我们处理每个点还能延伸多少。
- 每次这个发生变化就暴力更新，显然每个点这个变化不超过k次，所以复杂度O(nk)。

O(nk).

kakine

input output.

- 又名《消失的题面》《怠惰的出题人》《不见标程》《我们仍未知道那天的输入》《不是样例的样例》《我的压轴题不可能这么毒瘤》《请问您今天要来点数据吗》
- 一般出这种题的出题人，我们称之为**出题人或者sb***
- 大家玩得愉快么（逃

SDOI 2019

希望 hope [illegible]

- 口胡。

历年NOIP(CSP?)选讲/策略分析

CSP2019 Day1

通常，人们习惯将所有 n 位二进制串按照字典序排列，例如所有 2 位二进制串按字典序从小到大排列为：00，01，10，11。

格雷码（Gray Code）是一种特殊的 n 位二进制串排列法，它要求相邻的两个二进制串间**恰好**有一位**不同**，特别地，第一个串与最后一个串也算作相邻。

所有 2 位二进制串按格雷码排列的一个例子为：00，01，11，10。

n 位格雷码不止一种，下面给出其中一种格雷码的生成算法：

1. 1 位格雷码由两个 1 位二进制串组成，顺序为：0，1。
2. $n+1$ 位格雷码的前 2^n 个二进制串，可以由依此算法生成的 n 位格雷码（总共 2^n 个 n 位二进制串）按**顺序**排列，再在每个串前加一个前缀 0 构成。
3. $n+1$ 位格雷码的后 2^n 个二进制串，可以由依此算法生成的 n 位格雷码（总共 2^n 个 n 位二进制串）按**逆序**排列，再在每个串前加一个前缀 1 构成。

综上，$n+1$ 位格雷码，由 n 位格雷码的 2^n 个二进制串按顺序排列再加前缀 0，和按逆序排列再加前缀 1 构成，共 2^{n+1} 个二进制串。另外，对于 n 位格雷码中的 2^n 个 二进制串，我们按上述算法得到的排列顺序将它们从 $0 \sim 2^n - 1$ 编号。

现在给出 n，k，请你求出按上述算法生成的 n 位格雷码中的 k 号二进制串。

对于 50% 的数据：$n \le 10$

对于 80% 的数据：$k \le 5 \times 10^6$

对于 95% 的数据：$k \le 2^{63} - 1$

对于 100% 的数据：$1 \le n \le 64, 0 \le k < 2^n$

100+50+10

一个大小为 n 的树包含 n 个结点和 $n-1$ 条边，每条边连接两个结点，且任意两个结点间**有且仅有**一条简单路径互相可达。

小 Q 是一个充满好奇心的小朋友，有一天他在上学的路上碰见了一个大小为 n 的树，树上结点从 $1 \sim n$ 编号，1 号结点为树的根。除 1 号结点外，每个结点有一个父亲结点，u（$2 \le u \le n$）号结点的父亲为 f_u（$1 \le f_u < u$）号结点。

小 Q 发现这个树的每个结点上**恰有**一个括号，可能是 `(` 或 `)`。小 Q 定义 s_i 为：将根结点到 i 号结点的简单路径上的括号，按结点经过顺序依次排列组成的字符串。

显然 s_i 是个括号串，但不一定是合法括号串，因此现在小 Q 想对所有的 i（$1 \le i \le n$）求出，s_i 中有多少个**互不相同的子串**是**合法括号串**。

这个问题难倒了小 Q，他只好向你求助。设 s_i 共有 k_i 个不同子串是合法括号串， 你只需要告诉小 Q 所有 $i \times k_i$ 的异或和，即：

$$(1 \times k_1) \text{ xor } (2 \times k_2) \text{ xor } (3 \times k_3) \text{ xor } \cdots \text{ xor } (n \times k_n)$$

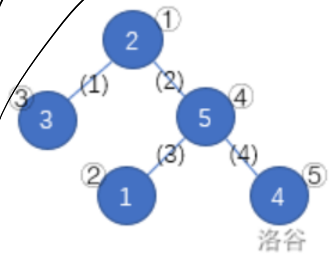
其中 xor 是位异或运算。

测试点编号	$n \le$	特殊性质
$1 \sim 2$	8	$f_i = i - 1$
$3 \sim 4$	200	
$5 \sim 7$	2000	
$8 \sim 10$		无
$11 \sim 14$	10^5	$f_i = i - 1$
$15 \sim 16$		无
$17 \sim 20$	5×10^5	

给定一个大小为 n 的树，它共有 n 个结点与 $n-1$ 条边，结点从 $1 \sim n$ 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。

接下来你需要进行**恰好** $n-1$ 次删边操作，每次操作你需要选一条**未被删去**的边，此时这条边所连接的两个结点上的数字将会**交换**，然后这条边将被删去。

$n-1$ 次操作过后，所有的边都将被删去。此时，按数字从小到大的顺序，将数字 $1 \sim n$ 所在的结点编号依次排列，就得到一个结点编号的排列 P_i。现在请你求出，在最优操作方案下能得到的**字典序最小**的 P_i。



如上图，蓝圈中的数字 $1 \sim 5$ 一开始分别在结点②、①、③、⑤、④。按照 (1)(4)(3)(2) 的顺序删去所有边，树变为下图。按数字顺序得到的结点编号排列为①③④②⑤，该排列是所有可能的结果中字典序最小的。

测试点编号	$n \le$	特殊性质
$1 \sim 2$	10	无
$3 \sim 4$	160	树的形态是一条链
$5 \sim 7$	2000	同上
$8 \sim 9$	160	存在度数为 $n-1$ 的结点
$10 \sim 12$	2000	同上
$13 \sim 16$	160	无
$17 \sim 20$	2000	无

对于所有测试点：$1 \le T \le 10$，保证给出的是一个树。

80

day2

160 + ~~80~~

120
40

330

320

测试点编号	$n=$	$m=$	$a_{i,j}<$	测试点编号	$n=$	$m=$	$a_{i,j}<$
1	2	2	2	7	10	2	10^3
2	2	3	2	8	10	3	10^3
3	5	2	2	$9\sim12$	40	2	10^3
4	5	3	2	$13\sim16$	40	3	10^3
5	10	2	2	$17\sim21$	40	500	10^3
6	10	3	2	$22\sim25$	100	2×10^3	998244353

对于所有测试点，保证 $1\le n\le100$，$1\le m\le2000$，$0\le a_{i,j}<998,244,353$。

测试点编号	$n \leq$	$a_i \leq$	$type =$
$1 \sim 3$	10	10	0
$4 \sim 6$	50	10^3	0
$7 \sim 9$	400	10^4	0
$10 \sim 16$	5000	10^5	0
$17 \sim 22$	5×10^5	10^6	0
$23 \sim 25$	4×10^7	10^9	1

80

84

120pts

对于$type = 0$的所有测试点,保证最后输出的答案$\leq 4 \times 10^{18}$

所有测试点满足：$type \in \{0,1\}$，$2 \leq n \leq 4 \times 10^7$，$1 \leq a_i \leq 10^9$，$1 \leq m \leq 10^5$，$1 \leq l_i \leq r_i \leq 10^9$，$0 \leq x,y,z,b_1,b_2 < 2^{30}$。

小简单正在学习离散数学，今天的内容是图论基础，在课上他做了如下两条笔记：

1. 一个大小为 n 的树由 n 个结点与 $n-1$ 条无向边构成，且满足任意两个结点间**有且仅有**一条简单路径。在树中删去一个结点及与它关联的边，树将分裂为若干个子树；而在树中删去一条边（保留关联结点，下同），树将分裂为**恰好**两个子树。
2. 对于一个大小为 n 的树与任意一个树中结点 c，称 c 是该树的**重心**当且仅当在树中删去 c 及与它关联的边后，分裂出的所有子树的大小均**不超过** $\lfloor \frac{n}{2} \rfloor$（其中 $\lfloor x \rfloor$ 是下取整函数）。对于包含至少一个结点的树，它的重心只可能有 1 或 2 个。

课后老师给出了一个大小为 n 的树 S，树中结点从 $1 \sim n$ 编号。小简单的课后作业是求出 S 单独删去每条边后，分裂出的两个子树的重心编号和之和。即：

$$\sum_{(u,v)\in E}\left(\sum_{\substack{1\le x\le n \\ \text{且 } x \text{ 号点是 } S'_u \text{ 的重心}}} x + \sum_{\substack{1\le y\le n \\ \text{且 } y \text{ 号点是 } S'_v \text{ 的重心}}} y\right)$$

上式中，E 表示树 S 的边集，(u,v) 表示一条连接 u 号点和 v 号点的边。S'_u 与 S'_v 分别表示树 S 删去边 (u,v) 后，u 号点与 v 号点所在的被分裂出的子树。

小简单觉得作业并不简单，只好向你求助，请你教教他。

测试点编号	$n =$	特殊性质
$1 \sim 2$	7	无
$3 \sim 5$	199	无
$6 \sim 8$	1999	无
$9 \sim 11$	49991	A
$12 \sim 15$	262143	B
16	99995	无
$17 \sim 18$	199995	无
$19 \sim 20$	299995	无

40

60pts

55 ~ 75

分析

表中特殊性质一栏，两个变量的含义为存在一个 $1 \sim n$ 的排列 $p_i(1 \leq i \leq n)$，使得：

- A：树的形态是一条链。即 $\forall 1 \leq i < n$，存在一条边 $(p_i, p_i + 1)$。
- B：树的形态是一个完美二叉树。即 $\forall 1 \leq i \leq \frac{n-1}{2}$，存在两条边 (p_i, p_{2i}) 与 (p_i, p_{2i+1})。

对于所有测试点：$1 \leq T \leq 5, 1 \leq u_i, v_i \leq n$。保证给出的图是一个树。

30min 读题

NOIP2018day1

一眼就能
得100分

春春是一名道路工程师，负责铺设一条长度为 n 的道路。

铺设道路的主要工作是填平下陷的地表。整段道路可以看作是 n 块首尾相连的区域，一开始，第 i 块区域下陷的深度为 d_i 。

春春每天可以选择一段连续区间 $[L, R]$ ，填充这段区间中的每块区域，让其下陷深度减少 1。在选择区间时，需要保证，区间内的每块区域在填充前下陷深度均不为 0 。

春春希望你能帮他设计一种方案，可以在最短的时间内将整段道路的下陷深度都变为 0 。

对于 30% 的数据，$1 \le n \le 10$;
对于 70% 的数据，$1 \le n \le 1000$;
对于 100% 的数据，$1 \le n \le 100000, 0 \le d_i \le 10000$ 。

2

在网友的国度中共有 n 种不同面额的货币，第 i 种货币的面额为 $a[i]$，你可以假设每一种货币都有无穷多张。为了方便，我们把货币种数为 n、面额数组为 $a[1..n]$ 的货币系统记作 (n, a)。

在一个完善的货币系统中，每一个非负整数的金额 x 都应该可以被表示出，即对每一个非负整数 x，都存在 n 个非负整数 $t[i]$ 满足 $a[i] \times t[i]$ 的和为 x。然而， 在网友的国度中，货币系统可能是不完善的，即可能存在金额 x 不能被该货币系统表示出。例如在货币系统 $n = 3, a = [2, 5, 9]$ 中，金额 $1, 3$ 就无法被表示出来。

a 小到大

两个货币系统 (n, a) 和 (m, b) 是等价的，当且仅当对于任意非负整数 x，它要么均可以被两个货币系统表出，要么不能被其中任何一个表出。

(n, a) $a_1 \rightarrow (n', a')$

现在网友们打算简化一下货币系统。他们希望找到一个货币系统 (m, b)，满足 (m, b) 与原来的货币系统 (n, a) 等价，且 m 尽可能的小。他们希望你来协助完成这个艰巨的任务：找到最小的 m。

$b_1 = a_1$

$b_2 = a'_1$

$dp(i, j)$

a_1 a

a_1 a_n

$dp[i+1, j+a_{i+1}]$

$dp[k, a_i] = 1$ x

=?

测试点	n	a_i	测试点	n	a_i
1			11		
2	$=2$		12	≤ 13	≤ 16
3			13		
4			14		
5	$=3$		15	≤ 25	≤ 40
6		≤ 1000	16		
7	$=4$		17		
8			18	≤ 100	≤ 25000
9	$=5$		19		
10			20		

对于 100% 的数据，满足 $1 \le T \le 20, n, a[i] \ge 1$。

3

300 pts

C 城将要举办一系列的赛车比赛。在比赛前，需要在城内修建 m 条赛道。

C 城一共有 n 个路口，这些路口编号为 $1, 2, \ldots, n$，有 $n-1$ 条适合于修建赛道的双向通行的道路，每条道路连接着两个路口。其中，第 i 条道路连接的两个路口编号为 a_i 和 b_i，该道路的长度为 l_i。借助这 $n-1$ 条道路，从任何一个路口出发都能到达其他所有的路口。

一条赛道是一组互不相同的道路 $e_1, e_2, \ldots, e_k$，满足可以从某个路口出发，依次经过 道路 $e_1, e_2, \ldots, e_k$（每条道路经过一次，不允许调头）到达另一个路口。一条赛道的长度等于经过的各道路的长度之和。为保证安全，要求每条道路至多被一条赛道经过。

目前赛道修建的方案尚未确定。你的任务是设计一种赛道修建的方案，使得修建的 m 条赛道中长度最小的赛道长度最大（即 m 条赛道中最短赛道的长度尽可能大）

二分ans

m=1：直径

dp

测试点编号	n	m	$a_i = 1$	$b_i = a_i + 1$	分支不超过 3
1	≤ 5	$= 1$	否	否	是
2	≤ 10	$\le n-1$	否	是	是
3	≤ 15	$\le n-1$	是	否	否
4	$\le 10^3$	$= 1$	否	否	是
5	$\le 3 \times 10^4$	$= 1$	是	否	否
6	$\le 3 \times 10^4$	$= 1$	否	否	否
7	$\le 3 \times 10^4$	$\le n-1$	是	否	否
8	$\le 5 \times 10^4$	$\le n-1$	是	否	否
9	$\le 10^3$	$\le n-1$	否	是	是
10	$\le 3 \times 10^4$	$\le n-1$	否	是	是
11	$\le 5 \times 10^4$	$\le n-1$	否	是	是
12	≤ 50	$\le n-1$	否	否	是
13	≤ 50	$\le n-1$	否	否	是
14	≤ 200	$\le n-1$	否	否	是
15	≤ 200	$\le n-1$	否	否	是
16	$\le 10^3$	$\le n-1$	否	否	是
17	$\le 10^3$	$\le n-1$	否	否	否
18	$\le 3 \times 10^4$	$\le n-1$	否	否	否
19	$\le 3 \times 10^4$	$\le n-1$	否	否	否

5pts

n-1

二分 ans

其中，「分支不超过 3」的含义为：每个路口至多有 3 条道路与其相连。

对于所有的数据，$2 \le n \le 5 \times 10^4$，$1 \le m \le n-1$，$1 \le a_i, b_i \le n$，$1 \le l_i \le 10^4$。

≥B ans

$\underline{O(n \lg^2 n)}$

$\cancel{n \lg^{1.5} n}$

ans(x)
ans' ≠ ans(x)

255 ~ ~~300~~

day2

NOIP 2018 Day 2

旅行

填数 game

保卫王国

15 min

15:52

小 Y 是一个爱好旅行的 OIer。她来到 X 国，打算将各个城市都玩一遍。

小Y了解到， X国的 n 个城市之间有 m 条双向道路。每条双向道路连接两个城市。 不存在两条连接同一对城市的道路，也不存在一条连接一个城市和它本身的道路。并且， 从任意一个城市出发，通过这些道路都可以到达任意一个其他城市。小 Y 只能通过这些 道路从一个城市前往另一个城市。

小 Y 的旅行方案是这样的：任意选定一个城市作为起点，然后从起点开始，每次可 以选择一条与当前城市相连的道路，走向一个没有去过的城市，或者沿着第一次访问该 城市时经过的道路后退到上一个城市。当小 Y 回到起点时，她可以选择结束这次旅行或 继续旅行。需要注意的是，小 Y 要求在旅行方案中，每个城市都被访问到。

为了让自己的旅行更有意义，小 Y 决定在每到达一个新的城市（包括起点）时，将 它的编号记录下来。她知道这样会形成一个长度为 n 的序列。她希望这个序列的字典序 最小，你能帮帮她吗？ 对于两个长度均为 n 的序列 A 和 B，当且仅当存在一个正整数 x，满足以下条件时， 我们说序列 A 的字典序小于 B。

- 对于任意正整数 $1 \leq i < x$，序列 A 的第 i 个元素 A_i 和序列 B 的第 i 个元素 B_i 相同。
- 序列 A 的第 x 个元素的值小于序列 B 的第 x 个元素的值。

对于 100% 的数据和所有样例， $1 \le n \le 5000$ 且 $m = n-1$ 或 $m = n$ 。

对于不同的测试点， 我们约定数据的规模如下：

测试点编号	$n =$	$m =$	特殊性质
1, 2, 3	10	$n-1$	无
4, 5	100	$n-1$	无
6, 7, 8	1000	$n-1$	每个城市最多与两个城市相连
9, 10	1000	$n-1$	无
11, 12, 13	5000	$n-1$	每个城市最多与三个城市相连
14, 15	5000	$n-1$	无
16, 17	10	n	无
18, 19	100	n	无
20, 21, 22	1000	n	每个城市最多与两个城市相连
23, 24, 25	5000	n	无

30min ~ 45min

删条边

题目描述

小 D 特别喜欢玩游戏。这一天，他在玩一款填数游戏。

这个填数游戏的棋盘是一个 $n \times m$ 的矩形表格。玩家需要在表格的每个格子中填入 一个数字（数字 0 或者数字 1），填数时需要满足一些限制。

下面我们来具体描述这些限制。

为了方便描述，我们先给出一些定义：

- 我们用每个格子的行列坐标来表示一个格子，即（行坐标，列坐标）。（注意： 行列坐标均从 0 开始编号）
- 合法路径 P：一条路径是合法的当且仅当：
  1. 这条路径从矩形表格的左上角的格子$(0,0)$出发，到矩形的右下角格子 $(n-1,m-1)$结束；
  2. 在这条路径中，每次只能从当前的格子移动到右边与它相邻的格子，或者 从当前格子移动到下面与它相邻的格子。
     例如：在下面这个矩形中，只有两条路径是合法的，它们分别是P_1： $(0,0) \to (0,1) \to (1,1)$和 P_2： $(0,0) \to (1,0) \to (1,1)$。

对于一条合法的路径 P，我们可以用一个字符串$w(P)$来表示，该字符串的长度为$n+m-2$，其中只包含字符“R”或者字符“D”，第 i 个字符记录了路径P 中第 i 步的移动 方法，“ R”表示移动到当前格子右边与它相邻的格子，“ D”表示移动到当前格子下面 与它相邻的格子。例如，上图中对于路径 P_1，有 $w(P_1) = "RD"$；而对于另一条路径 P_2，有$w(P_2) = "DR"$。

同时，将每条合法路径 P 经过的每个格子上填入的数字依次连接后，会得到一个长 度为$n+m-1$的 01 字符串，记为 $s(P)$。例如，如果我们在格子$(0,0)$和$(1,0)$上填入数字 0，在格子$(0,1)$和$(1,1)$上填入数字 1（见上图红色数字）。那么对于路径P_1，我们可以得 到$s(P_1) = "011"$,对于路径P_2，有 $s(P_2) = "001"$。

游戏要求小 D 找到一种填数字0、1 的方法，使得对于两条路径P_1， P_2，如果$w(P_1) > w(P_2)$，那么必须$s(P_1) \le s(P_2)$。我们说字符串 a 比字符串 b 小，当且仅当字符串 a 的字典序小于字符串 b 的字典序，字典序的定义详见第一题。但是仅仅是找一种方法无法满 足小 D 的好奇心，小 D 更想知道这个游戏有多少种玩法，也就是说，有多少种填数字 的方法满足游戏的要求?

小 D 能力有限，希望你帮助他解决这个问题，即有多少种填 0、1 的方法能满足题目要求。由于答案可能很大，你需要输出答案对 10^9+7 取模的结果。

$\frac{WC}{2yb}$

测试点编号	n ≤	m ≤
1~4	3	3
5~10	2	1000000
11~13	3	1000000
14~16	8	8
17~20	8	1000000

65pts

Z 国有 n 座城市，$(n-1)$ 条双向道路，每条双向道路连接两座城市，且任意两座城市都能通过若干条道路相互到达。

Z 国的国防部长小 Z 要在城市中驻扎军队。驻扎军队需要满足如下几个条件：

- 一座城市可以驻扎一支军队，也可以不驻扎军队。
- 由道路直接连接的两座城市中至少要有一座城市驻扎军队。
- 在城市里驻扎军队会产生花费，在编号为 i 的城市中驻扎军队的花费是 p_i。

小 Z 很快就规划出了一种驻扎军队的方案，使总花费最小。但是国王又给小 Z 提出了 m 个要求，每个要求规定了其中两座城市是否驻扎军队。小 Z 需要针对每个要求逐一给出回答。具体而言，如果国王提出的第 j 个要求能够满足上述驻扎条件（不需要考虑第 j 个要求之外的其它要求），则需要给出在此要求前提下驻扎军队的最小开销。如果国王提出的第 j 个要求无法满足，则需要输出 -1。现在请你来帮助小 Z。

测试点编号	type	$n = m =$
$1, 2$	A3	10
$3, 4$	C3	10
$5, 6$	A3	100
7	C3	100
$8, 9$	A3	2×10^3
$10, 11$	C3	2×10^3
$12, 13$	A1	10^5
$14, 15, 16$	A2	10^5
17	A3	10^5
$18, 19$	B1	10^5
$20, 21$	C1	10^5
22	C2	10^5
$23, 24, 25$	C3	10^5

- A：城市i与城市$i+1$直接相连。
- B：任意城市与城市 1 的距离不超过 100（距离定义为最短路径上边的数量），即如果这棵树以 1 号城市为根，深度不超过 100。
- C：在树的形态上无特殊约束。
- 1：询问时保证$a = 1, x = 1$，即要求在城市 1 驻军。对b,y没有限制。
- 2：询问时保证a, b是相邻的（由一条道路直接连通）
- 3：在询问上无特殊约束。

对于 100%的数据，保证 $1 \le n, m \le 10^5$，$1 \le p_i \le 10^5$，$1 \le u, v, a, b \le n$，$a \ne b$，$x, y \in \{0, 1\}$。

dp

B. $n, m \le 2000$

44

AB. 1.

565

552

24

162

64

24pt

68pts

NOIP2017 Day 1 T1 小凯的疑惑

小凯手中有两种面值的金币，两种面值均为正整数且彼此互素。每种金币小凯都有 无数个。在不找零的情况下，仅凭这两种金币，有些物品他是无法准确支付的。现在小 凯想知道在无法准确支付的物品中，最贵的价值是多少金币？注意：输入数据保证存在 小凯无法准确支付的商品。

$ab-a-b$

对于 30%的数据：$1 \le a, b \le 50$。

对于 60%的数据：$1 \le a, b \le 10^4$。

对于100%的数据：$1 \le a, b \le 10^9$。

10min

16:12

小明正在学习一种新的编程语言 A++，刚学会循环语句的他激动地写了好多程序并 给出了他自己算出的时间复杂度，可他的编程老师实在不想一个一个检查小明的程序， 于是你的机会来啦！下面请你编写程序来判断小明对他的每个程序给出的时间复杂度是否正确。

A++语言的循环结构如下：

时间复杂

```
F i x y
    循环体
E
```

其中 `F i x y` 表示新建变量 i（变量 i 不可与未被销毁的变量重名）并初始化为 x， 然后判断 i 和 y 的大小关系，若 i 小于等于 y 则进入循环，否则不进入。每次循环结束后 i 都会被修改成 $i+1$，一旦 i 大于 y 终止循环。

x 和 y 可以是正整数（x 和 y 的大小关系不定）或变量 n。n 是一个表示数据规模的变量，在时间复杂度计算中需保留该变量而不能将其视为常数，该数远大于 100。

“E”表示循环体结束。循环体结束时，这个循环体新建的变量也被销毁。

注：本题中为了书写方便，在描述复杂度时，使用大写英文字母“O”表示通常意义下“Θ”的概念。

输入格式

输入文件第一行一个正整数 t，表示有 t（$t \le 10$）个程序需要计算时间复杂度。 每个程序我们只需抽取其中 `F i x y` 和 `E` 即可计算时间复杂度。注意：循环结构 允许嵌套。

接下来每个程序的第一行包含一个正整数 L 和一个字符串，L 代表程序行数，字符 串表示这个程序的复杂度， `O(1)` 表示常数复杂度， `O(n^w)` 表示复杂度为n^w，其 中w是一个小于100的正整数（输入中不包含引号），输入保证复杂度只有 `O(1)` 和 `O(n^w)` 两种类型。

接下来 L 行代表程序中循环结构中的 `F i x y` 或者 `E`。 程序行若以 `F` 开头，表示进入一个循环，之后有空格分离的三个字符（串） `i x y`， 其中 i 是一个小写字母（保证不为n），表示新建的变量名，x 和 y 可能是正整数或 n ，已知若为正整数则一定小于 100。

程序行若以 `E` 开头，则表示循环体结束。

输出格式

输出文件共 t 行，对应输入的 t 个程序，每行输出 `Yes` 或 `No` 或者 `ERR` （输出中不包含引号），若程序实际复杂度与输入给出的复杂度一致则输出 `Yes` ，不一致则输出 `No` ，若程序有语法错误（其中语法错误只有: ① F 和 E 不匹配 ②新建的变量与已经存在但未被销毁的变量重复两种情况），则输出 `ERR` 。

注意：即使在程序不会执行的循环体中出现了语法错误也会编译错误，要输出 `ERR` 。

对于 30%的数据：不存在语法错误，数据保证小明给出的每个程序的前 $L/2$ 行一定为以 `F` 开头的语句，第 $L/2+1$ 行至第 L 行一定为以 E 开头的语句，$L \le 10$，若 x、y 均为整数，x 一定小于 y，且只有 y 有可能为 n。

对于 50%的数据：不存在语法错误，$L \le 100$，且若 x、y 均为整数，x 一定小于 y，且只有 y 有可能为 n。

对于 70%的数据：不存在语法错误，$L \le 100$。

对于 100%的数据：$L \le 100$。

20

50

200

策策同学特别喜欢逛公园。公园可以看成一张N个点M条边构成的有向图，且没有 自环和重边。其中1号点是公园的入口，N号点是公园的出口，每条边有一个非负权值， 代表策策经过这条边所要花的时间。

策策每天都会去逛公园，他总是从1号点进去，从N号点出来。

策策喜欢新鲜的事物，它不希望有两天逛公园的路线完全一样，同时策策还是一个 特别热爱学习的好孩子，它不希望每天在逛公园这件事上花费太多的时间。如果1号点 到N号点的最短路长为d，那么策策只会喜欢长度不超过$d + K$的路线。

策策同学想知道总共有多少条满足条件的路线，你能帮帮它吗?

为避免输出过大，答案对P取模。

如果有无穷多条合法的路线，请输出-1。

测试点编号	T	N	M	K	是否有0边
1	5	5	10	0	否
2	5	1000	2000	0	否
3	5	1000	2000	50	否
4	5	1000	2000	50	否
5	5	1000	2000	50	否
6	5	1000	2000	50	是
7	5	100000	200000	0	否
8	3	100000	200000	50	否
9	3	100000	200000	50	是
10	3	100000	200000	50	是

对于 100%的数据, $1 \le P \le 10^9, 1 \le a_i, b_i \le N, 0 \le c_i \le 1000$。

数据保证：至少存在一条合法的路线。

$dp[x, t]$

$t \le k$

$O(mk)$

70 pts

day2

现有一块大奶酪，它的高度为 h，它的长度和宽度我们可以认为是无限大的，奶酪 中间有许多 半径相同的球形空洞。我们可以在这块奶酪中建立空间坐标系，在坐标系中， 奶酪的下表面为$z = 0$，奶酪的上表面为$z = h$。

现在，奶酪的下表面有一只小老鼠 Jerry，它知道奶酪中所有空洞的球心所在的坐 标。如果两个空洞相切或是相交，则 Jerry 可以从其中一个空洞跑到另一个空洞，特别 地，如果一个空洞与下表面相切或是相交，Jerry 则可以从奶酪下表面跑进空洞；如果 一个空洞与上表面相切或是相交，Jerry 则可以从空洞跑到奶酪上表面。

位于奶酪下表面的 Jerry 想知道，在 不破坏奶酪 的情况下，能否利用已有的空洞跑 到奶酪的上表面去?

空间内两点$P_1(x_1, y_1, z_1)$、$P2(x_2, y_2, z_2)$的距离公式如下：

$$\mathrm{dist}(P_1, P_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

95/95

12 min

4:30

对于 20%的数据,$n = 1$,$1 \le h$, $r \le 10,000$,坐标的绝对值不超过 $10,000$。

对于 40%的数据,$1 \le n \le 8$, $1 \le h$, $r \le 10,000$,坐标的绝对值不超过 $10,000$。

对于80%的数据, $1 \le n \le 1,000$, $1 \le h, r \le 10,000$,坐标的绝对值不超过$10,000$。

对于 100%的数据,$1 \le n \le 1,000$,$1 \le h, r \le 1,000,000,000$,$T \le 20$,坐标的 绝对值不超过 $1,000,000,000$。

n^2

参与考古挖掘的小明得到了一份藏宝图，藏宝图上标出了 n 个深埋在地下的宝藏屋， 也给出了这 n 个宝藏屋之间可供开发的m 条道路和它们的长度。

小明决心亲自前往挖掘所有宝藏屋中的宝藏。但是，每个宝藏屋距离地面都很远， 也就是说，从地面打通一条到某个宝藏屋的道路是很困难的，而开发宝藏屋之间的道路 则相对容易很多。

小明的决心感动了考古挖掘的赞助商，赞助商决定免费赞助他打通一条从地面到某 个宝藏屋的通道，通往哪个宝藏屋则由小明来决定。

在此基础上，小明还需要考虑如何开凿宝藏屋之间的道路。已经开凿出的道路可以 任意通行不消耗代价。每开凿出一条新道路，小明就会与考古队一起挖掘出由该条道路 所能到达的宝藏屋的宝藏。另外，小明不想开发无用道路，即两个已经被挖掘过的宝藏 屋之间的道路无需再开发。

新开发一条道路的代价是：

随机

$$\mathrm{L} \times \mathrm{K}$$

L代表这条道路的长度，K代表从赞助商帮你打通的宝藏屋到这条道路起点的宝藏屋所经过的 宝藏屋的数量（包括赞助商帮你打通的宝藏屋和这条道路起点的宝藏屋） 。

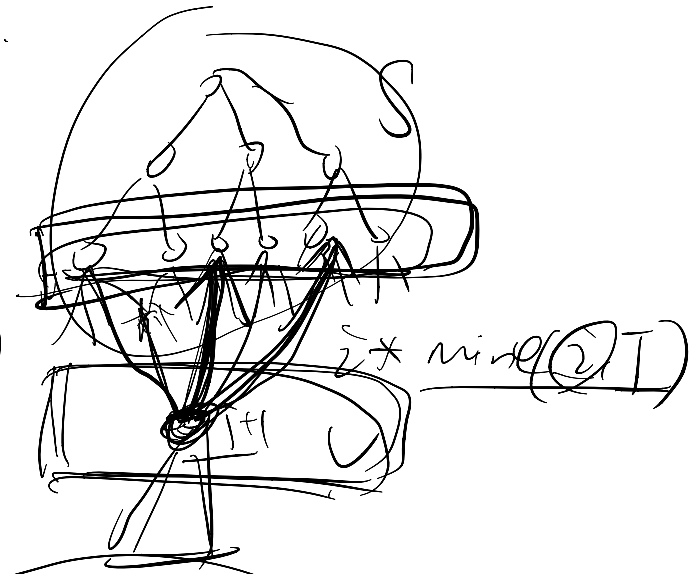
请你编写程序为小明选定由赞助商打通的宝藏屋和之后开凿的道路，使得工程总代 价最小，并输出这个最小值。

TCS

$dp(i, S, T)$

前i, 已选

枚举 $V \subset U/S$

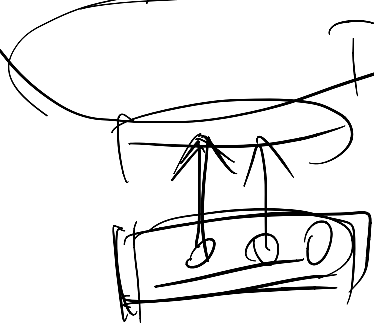


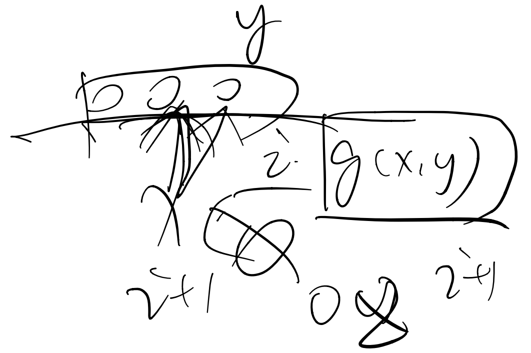
$O\left(n \cdot \sum_{S} 2^{n}\right) = O(n \cdot 4^{n})$

2^n

$T \subset S$

$dp(i, S, T) + i * mine(V, T) \xrightarrow{V \subset (U-S)} \min \to dp(i, S \cup V, V)$





$mine(V, T) = \sum_{x \in V} mine(x, T)$

$= \min_{y \in T} g(x, y)$

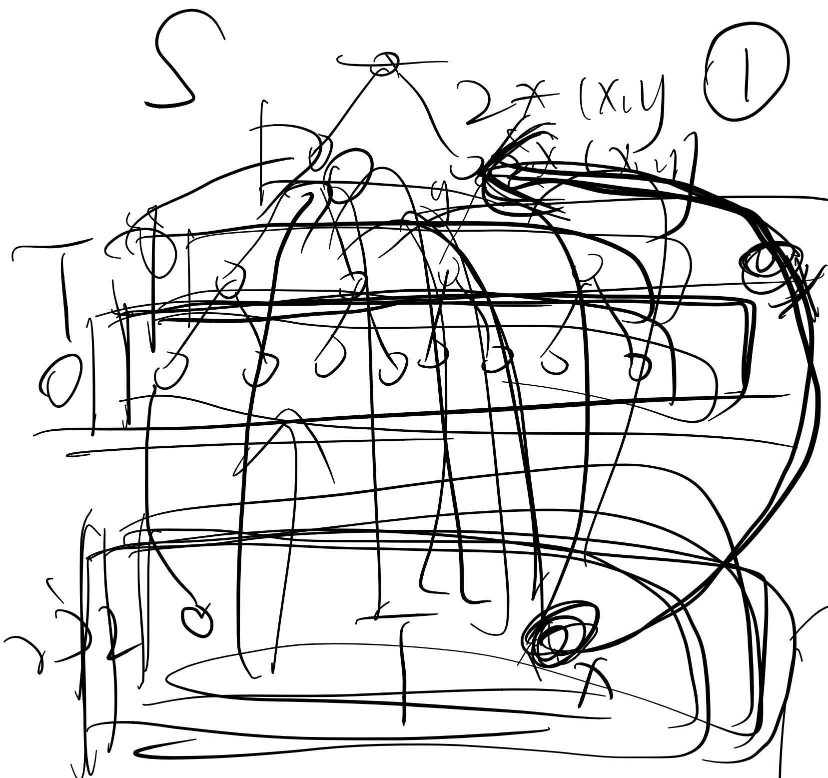
$\boxed{\text{不论}T}$

$T \subseteq (U-S)$

$dp(i, S) + i \times \boxed{mine(T, S)} \to dp(i+1, S \cup T)$

$n \cdot \sum_{S} 2^{|S|} = 3^n \cdot n$

$\boxed{dp(n, U)} \leq$ 最优解



① 最优解. 全算到

$dp(n, U) \geq$ 最优

$dp(n, U) = ans$

$O(n 3^n)$

`Sylvia` 是一个热爱学习的女孩子。

前段时间，`Sylvia` 参加了学校的军训。众所周知，军训的时候需要站方阵。

Sylvia 所在的方阵中有$n \times m$名学生，方阵的行数为 n，列数为 m。

为了便于管理，教官在训练开始时，按照从前到后，从左到右的顺序给方阵中 的学生从 1 到 $n \times m$ 编上了号码（参见后面的样例）。即：初始时，第 i 行第 j 列 的学生的编号是$(i-1) \times m + j$。

然而在练习方阵的时候，经常会有学生因为各种各样的事情需要离队。在一天 中，一共发生了 q件这样的离队事件。每一次离队事件可以用数对$(x,y)(1 \le x \le n, 1 \le y \le m)$描述，表示第 x 行第 y 列的学生离队。

在有学生离队后，队伍中出现了一个空位。为了队伍的整齐，教官会依次下达 这样的两条指令：

1. 向左看齐。这时第一列保持不动，所有学生向左填补空缺。不难发现在这条 指令之后，空位在第 x 行第 m 列。
2. 向前看齐。这时第一行保持不动，所有学生向前填补空缺。不难发现在这条 指令之后，空位在第 n 行第 m 列。

教官规定不能有两个或更多学生同时离队。即在前一个离队的学生归队之后， 下一个学生才能离队。因此在每一个离队的学生要归队时，队伍中有且仅有第 n 行 第 m 列一个空位，这时这个学生会自然地填补到这个位置。

因为站方阵真的很无聊，所以 `Sylvia` 想要计算每一次离队事件中，离队的同学 的编号是多少。

k

[n+m]

测试点编号	n	m	q	其他约定
$1 \sim 6$	$\le 10^3$	$\le 10^3$	≤ 500	无
$7 \sim 10$	$\le 5 \times 10^4$	$\le 5 \times 10^4$	≤ 500	无
$11 \sim 12$	$= 1$	$\le 10^5$	$\le 10^5$	所有事件 $x = 1$
$13 \sim 14$	$= 1$	$\le 3 \times 10^5$	$\le 3 \times 10^5$	所有事件 $x = 1$
$15 \sim 16$	$\le 3 \times 10^5$	$\le 3 \times 10^5$	$\le 3 \times 10^5$	所有事件 $x = 1$
$17 \sim 18$	$\le 10^5$	$\le 10^5$	$\le 10^5$	无
$19 \sim 20$	$\le 3 \times 10^5$	$\le 3 \times 10^5$	$\le 3 \times 10^5$	无

30pts

30pts

$O(q\lg n)$

数据保证每一个事件满足 $1 \le x \le n, 1 \le y \le m$

80pts → 5pts

BIT

noip2016day1

1e5

这时$singer$告诉小南一个谜题:“眼镜藏在我左数第3个玩具小人的右数第1个玩具小人的左数第2个玩具小人那里。”

小南发现,这个谜题中玩具小人的朝向非常关键,因为朝内和朝外的玩具小人的左右方向是相反的:面朝圈内的玩具小人,它的左边是顺时针方向,右边是逆时针方向;而面向圈外的玩具小人,它的左边是逆时针方向,右边是顺时针方向。

小南一边艰难地辨认着玩具小人,一边数着:

$singer$朝内,左数第3个是$archer$。

$archer$朝外,右数第1个是$thinker$。

$thinker$朝外,左数第2个是$writer$。

所以眼镜藏在$writer$这里!

虽然成功找回了眼镜,但小南并没有放心。如果下次有更多的玩具小人藏他的眼镜,或是谜题的长度更长,他可能就无法找到眼镜了。所以小南希望你写程序帮他解决类似的谜题。这样的谜题具体可以描述为:

有 n个玩具小人围成一圈,已知它们的职业和朝向。现在第1个玩具小人告诉小南一个包含m条指令的谜题,其中第 z条指令形如“左数/右数第s,个玩具小人”。你需要输出依次数完这些指令后,到达的玩具小人的职业。

`小c` 同学认为跑步非常有趣，于是决定制作一款叫做《天天爱跑步》的游戏。《天天爱跑步》是一个养成类游戏，需要玩家每天按时上线,完成打卡任务。

这个游戏的地图可以看作一一棵包含 n 个结点和 $n-1$ 条边的树，每条边连接两个结点,且任意两个结点存在一条路径互相可达。树上结点编号为从 1 到 n 的连续正整数。

现在有 m 个玩家，第 i 个玩家的起点为 s_i，终点为 t_i。每天打卡任务开始时，所有玩家在第 0 秒同时从自己的起点出发，以每秒跑一条边的速度，不间断地沿着最短路径向着自己的终点跑去，跑到终点后该玩家就算完成了打卡任务。(由于地图是一棵树，所以每个人的路径是唯一的)

`小c` 想知道游戏的活跃度，所以在每个结点上都放置了一个观察员。在结点 j 的观察员会选择在第 w_j 秒观察玩家，一个玩家能被这个观察员观察到当且仅当该玩家在第 w_j 秒也正好到达了结点 j 。`小c` 想知道每个观察员会观察到多少人?

注意：我们认为一个玩家到达自己的终点后该玩家就会结束游戏，他不能等待一 段时间后再被观察员观察到。 即对于把结点 j 作为终点的玩家：若他在第 w_j 秒前到达终点,则在结点 j 的观察员不能观察到该玩家；若他正好在第 w_j 秒到达终点,则在结点 j 的观察员可以观察到这个玩家。

天天爱跑步

<table>
<tr><th>测试点编号</th><th>n</th><th>m</th><th>约定</th></tr>
<tr><td>1</td><td rowspan="2">= 991</td><td rowspan="2">= 991</td><td rowspan="2">所有人的起点等于自己的终点，即Si = Ti</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td><td rowspan="2">= 992</td><td rowspan="2">= 992</td><td rowspan="2">Wj = 0</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td><td>= 993</td><td>= 993</td><td>无</td></tr>
<tr><td>6</td><td rowspan="3">= 99994</td><td rowspan="3">= 99994</td><td rowspan="3">树退化成一条链，其中1与2有边，2与3有边，…，n-1与n有边</td></tr>
<tr><td>7</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td rowspan="4">= 99995</td><td rowspan="4">= 99995</td><td rowspan="4">所有的si = 1</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td></tr>
<tr><td>12</td></tr>
<tr><td>13</td><td rowspan="4">99996</td><td rowspan="4">= 99996</td><td rowspan="4">所有的Ti = 1</td></tr>
<tr><td>14</td></tr>
<tr><td>15</td></tr>
<tr><td>16</td></tr>
<tr><td>17</td><td rowspan="3">= 99997</td><td rowspan="3">= 99997</td><td rowspan="4">无</td></tr>
<tr><td>18</td></tr>
<tr><td>19</td></tr>
<tr><td>20</td><td>= 299998</td><td>= 299998</td></tr>
</table>

对于刚上大学的牛牛来说，他面临的第一个问题是如何根据实际情况申请合适的课程。

在可以选择的课程中，有 $2n$ 节课程安排在 n 个时间段上。在第 i（$1 \le i \le n$）个时间段上，两节内容相同的课程同时在不同的地点进行，其中，牛牛预先被安排在教室 c_i 上课，而另一节课程在教室 d_i 进行。

在不提交任何申请的情况下，学生们需要按时间段的顺序依次完成所有的 n 节安排好的课程。如果学生想更换第 i 节课程的教室，则需要提出申请。若申请通过，学生就可以在第 i 个时间段去教室 d_i 上课，否则仍然在教室 c_i 上课。

由于更换教室的需求太多，申请不一定能获得通过。通过计算，牛牛发现申请更换第 i 节课程的教室时，申请被通过的概率是一个已知的实数 k_i，并且对于不同课程的申请，被通过的概率是互相独立的。

学校规定，所有的申请只能在学期开始前一次性提交，并且每个人只能选择至多 m 节课程进行申请。这意味着牛牛必须一次性决定是否申请更换每节课的教室，而不能根据某些课程的申请结果来决定其他课程是否申请；牛牛可以申请自己最希望更换教室的 m 门课程，也可以不用完这 m 个申请的机会，甚至可以一门课程都不申请。

因为不同的课程可能会被安排在不同的教室进行，所以牛牛需要利用课间时间从一间教室赶到另一间教室。

牛牛所在的大学有 v 个教室，有 e 条道路。每条道路连接两间教室，并且是可以双向通行的。由于道路的长度和拥堵程度不同，通过不同的道路耗费的体力可能会有所不同。 当第 i（$1 \le i \le n-1$）节课结束后，牛牛就会从这节课的教室出发，选择一条耗费体力最少的路径前往下一节课的教室。

现在牛牛想知道，申请哪几门课程可以使他因在教室间移动耗费的体力值的总和的期望值最小，请你帮他求出这个最小值。

测试点	n	m	v	特殊性质1	特殊性质2
1	≤1	≤1	≤300	×	×
2	≤2	≤0	≤20		
3		≤1	≤100		
4		≤2	≤300		
5	≤3	≤0	≤20	✓	✓
6		≤1	≤100		×
7		≤2	≤300	×	
8	≤10	≤0		✓	✓
9		≤1	≤20		×
10		≤2	≤100	×	
11		≤10	≤300		✓
12	≤20	≤0	≤20	✓	×
13		≤1	≤100	×	
14		≤2	≤300	✓	
15		≤20		×	✓
16	≤300	≤0	≤20		×
17		≤1	≤100		
18		≤2	≤300	✓	✓
19		≤300		×	
20	≤2000	≤0	≤20		×
21		≤1			
22		≤2	≤100		
23		≤2000			
24			≤300		
25					

特殊性质1:图上任意两点 $a_i, b_i, a_i \neq b_i$间,存在一条耗费体力最少的路径只包含一条道路。

特殊性质2:对于所有的 $1 \le i \le n$, $k_i = 1$。

day2

组合数 $\binom{n}{m}$ 表示的是从 n 个物品中选出 m 个物品的方案数。举个例子，从 $(1, 2, 3)$ 三个物品中选择两个物品可以有 $(1, 2), (1, 3), (2, 3)$ 这三种选择方法。根据组合数的定义，我们可以给出计算组合数 $\binom{n}{m}$ 的一般公式:

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

其中 $n! = 1 \times 2 \times \cdots \times n$；特别地，定义 $0! = 1$。

小葱想知道如果给定 n, m 和 k，对于所有的 $0 \leq i \leq n, 0 \leq j \leq \min(i, m)$ 有多少对 (i, j) 满足 $k | \binom{i}{j}$。

测试点	n	m	k	t
1	≤ 3	≤ 3	$= 2$	$= 1$
2			$= 3$	$\le 10^4$
3	≤ 7	≤ 7	$= 4$	$= 1$
4			$= 5$	$\le 10^4$
5	≤ 10	≤ 10	$= 6$	$= 1$
6			$= 7$	$\le 10^4$
7	≤ 20	≤ 100	$= 8$	$= 1$
8			$= 9$	$\le 10^4$
9	≤ 25	≤ 2000	$= 10$	$= 1$
10			$= 11$	$\le 10^4$
11	≤ 60	≤ 20	$= 12$	$= 1$
12			$= 13$	$\le 10^4$
13	≤ 100	≤ 25	$= 14$	$= 1$
14			$= 15$	$\le 10^4$
15		≤ 60	$= 16$	$= 1$
16			$= 17$	$\le 10^4$
17	≤ 2000	≤ 100	$= 18$	$= 1$
18			$= 19$	$\le 10^4$
19		≤ 2000	$= 20$	$= 1$
20			$= 21$	$\le 10^4$

- 对于全部的测试点，保证 $0 \le n, m \le 2 \times 10^3$，$1 \le t \le 10^4$。

本题中，我们将用符号 $\lfloor c \rfloor$ 表示对 c 向下取整，例如：$\lfloor 3.0 \rfloor = \lfloor 3.1 \rfloor = \lfloor 3.9 \rfloor = 3$。

蛐蛐国最近蚯蚓成灾了！隔壁跳蚤国的跳蚤也拿蚯蚓们没办法，蛐蛐国王只好去请神刀手来帮他们消灭蚯蚓。

蛐蛐国里现在共有 n 只蚯蚓（n 为正整数）。每只蚯蚓拥有长度，我们设第 i 只蚯蚓的长度为 a_i $(i = 1, 2, \dots, n)$，并保证所有的长度都是非负整数（即：可能存在长度为 0 的蚯蚓）。

每一秒，神刀手会在所有的蚯蚓中，准确地找到最长的那一只（如有多个则任选一个）将其切成两半。神刀手切开蚯蚓的位置由常数 p（是满足 $0 < p < 1$ 的有理数）决定，设这只蚯蚓长度为 x，神刀手会将其切成两只长度分别为 $\lfloor px \rfloor$ 和 $x - \lfloor px \rfloor$ 的蚯蚓。特殊地，如果这两个数的其中一个等于 0，则这个长度为 0 的蚯蚓也会被保留。此外，除了刚刚产生的两只新蚯蚓，其余蚯蚓的长度都会增加 q（是一个非负整常数）。

蛐蛐国王知道这样不是长久之计，因为蚯蚓不仅会越来越多，还会越来越长。蛐蛐国王决定求助于一位有着洪荒之力的神秘人物，但是救兵还需要 m 秒才能到来……（m 为非负整数）

蛐蛐国王希望知道这 m 秒内的战况。具体来说，他希望知道：

- m 秒内，每一秒被切断的蚯蚓被切断前的长度（有 m 个数）；
- m 秒后，所有蚯蚓的长度（有 $n + m$ 个数）。

蛐蛐国王当然知道怎么做啦！但是他想考考你……

<table>
<tr><th>测试点</th><th>n</th><th>m</th><th>t</th><th>a_i</th><th>v</th><th>q</th></tr>
<tr><td>1</td><td>$=1$</td><td rowspan="3">$=0$</td><td rowspan="8">$=1$</td><td rowspan="11">$\le 10^6$</td><td rowspan="12">≤ 2</td><td rowspan="5">$=0$</td></tr>
<tr><td>2</td><td>$=10^3$</td></tr>
<tr><td>3</td><td>$=10^5$</td></tr>
<tr><td>4</td><td>$=1$</td><td rowspan="4">$=10^3$</td></tr>
<tr><td>5</td><td>$=10^3$</td></tr>
<tr><td>6</td><td>$=1$</td><td rowspan="2">≤ 200</td></tr>
<tr><td>7</td><td>$=10^3$</td></tr>
<tr><td>8</td><td>$=5\times10^4$</td><td>$=5\times10^4$</td><td rowspan="7">$=0$</td></tr>
<tr><td>9</td><td rowspan="6">$=10^5$</td><td>$=10^5$</td><td>$=2$</td></tr>
<tr><td>10</td><td>$=2\times10^6$</td><td>$=21$</td></tr>
<tr><td>11</td><td>$=2.5\times10^6$</td><td>$=26$</td></tr>
<tr><td>12</td><td>$=3.5\times10^6$</td><td>$=36$</td><td rowspan="2">$\le 10^7$</td></tr>
<tr><td>13</td><td>$=5\times10^6$</td><td>$=51$</td><td rowspan="2">$\le 10^9$</td></tr>
<tr><td>14</td><td>$=7\times10^6$</td><td>$=71$</td><td rowspan="7">$\le 10^8$</td></tr>
<tr><td>15</td><td rowspan="2">$=5\times10^4$</td><td>$=5\times10^4$</td><td>$=1$</td><td rowspan="2">≤ 2</td><td rowspan="6">≤ 200</td></tr>
<tr><td>16</td><td>$=1.5\times10^5$</td><td>$=2$</td></tr>
<tr><td>17</td><td rowspan="4">$=10^5$</td><td>$=10^5$</td><td>$=3$</td><td rowspan="4">$\le 10^9$</td></tr>
<tr><td>18</td><td>$=3\times10^5$</td><td>$=4$</td></tr>
<tr><td>19</td><td>$=3.5\times10^6$</td><td>$=36$</td></tr>
<tr><td>20</td><td>$=7\times10^6$</td><td>$=71$</td></tr>
</table>

`Kiana` 最近沉迷于一款神奇的游戏无法自拔。

简单来说，这款游戏是在一个平面上进行的。

有一架弹弓位于 $(0,0)$ 处，每次 `Kiana` 可以用它向第一象限发射一只红色的小鸟，小鸟们的飞行轨迹均为形如 $y=ax^2+bx$ 的曲线，其中 a,b 是 `Kiana` 指定的参数，且必须满足 $a<0$，a,b 都是实数。

当小鸟落回地面（即 x 轴）时，它就会瞬间消失。

在游戏的某个关卡里，平面的第一象限中有 n 只绿色的小猪，其中第 i 只小猪所在的坐标为 (x_i,y_i)。

如果某只小鸟的飞行轨迹经过了 (x_i,y_i)，那么第 i 只小猪就会被消灭掉，同时小鸟将会沿着原先的轨迹继续飞行；

如果一只小鸟的飞行轨迹没有经过 (x_i,y_i)，那么这只小鸟飞行的全过程就不会对第 i 只小猪产生任何影响。

例如，若两只小猪分别位于 $(1,3)$ 和 $(3,3)$，`Kiana` 可以选择发射一只飞行轨迹为 $y=-x^2+4x$ 的小鸟，这样两只小猪就会被这只小鸟一起消灭。

而这个游戏的目的，就是通过发射小鸟消灭所有的小猪。

这款神奇游戏的每个关卡对 `Kiana` 来说都很难，所以 `Kiana` 还输入了一些神秘的指令，使得自己能更轻松地完成这个游戏。这些指令将在【输入格式】中详述。

假设这款游戏一共有 T 个关卡，现在 `Kiana` 想知道，对于每一个关卡，至少需要发射多少只小鸟才能消灭所有的小猪。由于她不会算，所以希望由你告诉她。

第一行包含一个正整数 T，表示游戏的关卡总数。

下面依次输入这 T 个关卡的信息。每个关卡第一行包含两个非负整数 n, m，分别表示该关卡中的小猪数量和 Kiana 输入的神秘指令类型。接下来的 n 行中，第 i 行包含两个正实数 x_i, y_i，表示第 i 只小猪坐标为 (x_i, y_i)。数据保证同一个关卡中不存在两只坐标完全相同的小猪。

如果 $m = 0$，表示 `Kiana` 输入了一个没有任何作用的指令。

如果 $m = 1$，则这个关卡将会满足：至多用 $\lceil n/3 + 1 \rceil$ 只小鸟即可消灭所有小猪。

如果 $m = 2$，则这个关卡将会满足：一定存在一种最优解，其中有一只小鸟消灭了至少 $\lfloor n/3 \rfloor$ 只小猪。

保证 $1 \leq n \leq 18$，$0 \leq m \leq 2$，$0 < x_i, y_i < 10$，输入中的实数均保留到小数点后两位。

上文中，符号 $\lceil c \rceil$ 和 $\lfloor c \rfloor$ 分别表示对 c 向上取整和向下取整，例如：$\lceil 2.1 \rceil = \lceil 2.9 \rceil = \lceil 3.0 \rceil = \lfloor 3.0 \rfloor = \lfloor 3.1 \rfloor = \lfloor 3.9 \rfloor = 3$。

【数据范围】

<table>
<tr><th>测试点编号</th><th>n</th><th>m</th><th>T</th></tr>
<tr><td>1</td><td rowspan="2">≤ 2</td><td rowspan="12">$= 0$</td><td>≤ 10</td></tr>
<tr><td>2</td><td>≤ 30</td></tr>
<tr><td>3</td><td rowspan="2">≤ 3</td><td>≤ 10</td></tr>
<tr><td>4</td><td>≤ 30</td></tr>
<tr><td>5</td><td rowspan="2">≤ 4</td><td>≤ 10</td></tr>
<tr><td>6</td><td>≤ 30</td></tr>
<tr><td>7</td><td>≤ 5</td><td rowspan="4">≤ 10</td></tr>
<tr><td>8</td><td>≤ 6</td></tr>
<tr><td>9</td><td>≤ 7</td></tr>
<tr><td>10</td><td>≤ 8</td></tr>
<tr><td>11</td><td>≤ 9</td><td rowspan="4">≤ 30</td></tr>
<tr><td>12</td><td>≤ 10</td></tr>
<tr><td>13</td><td rowspan="2">≤ 12</td><td>$= 1$</td></tr>
<tr><td>14</td><td>$= 2$</td></tr>
<tr><td>15</td><td rowspan="3">≤ 15</td><td>$= 0$</td><td rowspan="3">≤ 15</td></tr>
<tr><td>16</td><td>$= 1$</td></tr>
<tr><td>17</td><td>$= 2$</td></tr>
<tr><td>18</td><td rowspan="3">≤ 18</td><td>$= 0$</td><td rowspan="3">≤ 5</td></tr>
<tr><td>19</td><td>$= 1$</td></tr>
<tr><td>20</td><td>$= 2$</td></tr>
</table>