判斜率(x/gcd, y/gcd)直接丢map里unique

无方案和答案 % MOD 为 0 是有区别的

打标记使用时间戳

a = 10 \* a + 1 可以用矩乘加速

数组要开1e5 [+ 5]!

pow(a, b)会调用c++自带函数

强联通、双联通要考虑一个孤立点

MOD的时候：(a - b + MOD) % MOD (a + b \* c % MOD) % MOD

stack里有时存的边，这种时候大小不要开错了

选择性段错误: 没return 没赋初值

凸包排序后数组顺序会改变，不可以在这之后求重心

位运算优先级小于 ==

(int)x != round(x)

hash字符串：t = t \* 27(!) + s[i] - 'A' + 1

有些dfs里用到的数组开全局会跪

n = 1e4 时明摆着要 n^2 bitset压位

fact[0] = 1

在打表找规律之前要先自己试几组数据，确保暴力程序的正确性

到最后阶段如果还卡算法应该去冲一冲暴力。

没有测一些极小的数据，为0或1的情况没有考虑，有时候需要特判。

i-- -> i++

1ll << n

判线段相交时考虑线段重叠的情况

个性坑点

cjy:

vector<int> v; for(int i = 0; i <(没有=) v.size(); ++i)

Hash map<unsigned long long, int> hash时乘的常数，以及idx()返回值均需ULL

double 不要开成 int

long long 读入别忘开 lld%

读题还是要有重点的去读

求最短路上的边的集合，要用dist1[u]+dist2[v]+len(u,v)==dis(S,T)。dist1[v]+dist2[v]==dis(S,T)只是说明点v在最短路上。

改动代码之后要检查对原来对的输出结果有没有影响，不能只关注改动的结果变化。

树链剖分搜出DFS序要先访问size最大的儿子，来保证一条重链在DFS序中为一段连续的区间。

行列n/m写错（经常出现），可以自己测一些行列差别较大的数据。这可能也会出现RE的情况。

分解质因数，注意n＝1的情况，质因数个数为零。

位运算<<，范围超过int需要用到long long的时候，要写1LL左移。

对题目中的一些数据进行了重新标号(如离散化、排序、dfs序、拓扑序）之后，使用的时候要注意是原标号还是新的标号，主要区分是用到标号的数组还是数组下标。

想到了正解高斯消元，因为看到精度要到1e-8，感觉精度会有问题而没有进行尝试。

用实数进行高斯消元，找系数非零的方程，直接找系数绝对值最大的，可以不用到eps。

网络流的时候要注意不要漏算连向源和汇的边数。

多组数据时中途－1不要return 0.

xxxxxyt:

1、审题方面：

（1）对题目中的重点应采取恰当的勾画，需要重点勾画出的内容有：明确提出要求的句子、关键词（distinct, succesive, directed etc）、数据范围、特殊的要求或条件、有疑问的地方

（2）尽量不要按照自己的思维模式对不清楚的题意进行猜测，而且也不要过于相信生活经验（因为题目的模型往往与现实又很大差别），即便有这样的猜测也应当明确标注出并告诉队长

（3）不能为了节省一点点时间而跳过某些自以为无聊的句子（条件也可能出现在背景描述中）不读

（4）在听完队友讲述的题意后也应该读一遍input/output确认格式

2、算法方面：

（1）有时会将具体问题过分抽象化，反而导致忽略了最直观的模拟算法

（2）想到一个算法时没有完全check清它的正确性就告诉队长，导致有时队长没有看出错的话就会浪费大量机时，所以想到算法后不应该先急于表达，而且check的时候要带入题目中的所有关键点以查看有没有考虑漏的情况

（3）有时会出错的反例导致思维偏离正确方向，出反例的时候应该更加严谨

（4）敢冲敢过

（5）有时会在仔细思考如何递推之前盲目地打表找规律，浪费了大量时间

3、实现方面：

（1）准备的时候需要考虑这些事：需要用到几个函数以及这些函数应该怎么写、需要用到哪些变量以及与其相关的初值问题清零问题、边界情况和特殊情况，但没有必要将它们都写在纸上

（2）有时候会犯一些粗心的错误，如：忘记删掉调试语句（交之前一定要浏览一次整个代码及跑一遍样例check）、数组大小算错

（3）代码常数经常会很大（暂时还不知道怎么改善）

（4）对待多组测试数据时要有效地进行预处理与反复利用记忆化搜索的结果

做法向

博弈题做法：1.由最终态BFS（类似构了一颗树）2.打表找sg函数规律

没辙时想dp和网络流

启发式合并 nlgn

n / 1 + n / 2 + ... = nlgn