A 钢筋切割

我们可以把题目简化为向背包中放入物体的问题，长钢筋即为背包，短钢筋即为物品。
搜索出全部的方案再找当然是行不通的,因为这是一高纬度的背包问题。我们可以先二分枚举答案然后判断可行性.这样就会快很多了。但是即使如此,搜索的速度依然会比较慢。我们可以采取以下优化。

1.搜索顺序: 先放大的背包;

2.去重: 把背包进行编号,物品也进行编号,相同的物品(体积相同)有不同的编号,保证编号大的物品所在的背包编号一定大于或等于编号小的物品所在的背包编号.

3.可行性剪枝: 如果不能放任何物品的背包的剩余容量之和大于应剩余的容量则视为不可行方案.这样说大家可能不理解,我举个例子.

假如要放 M 个物品,现在已经在前 i 个背包中放了 j 个物品,并且这 i 个背包已经容不下剩下 M-j 个物品中的任何一个,那么我们知道这 i 个背包中还是有剩余容量的,但这些剩余容量基本是废了的.在最好情况下,是放完了 M 个物品没有出现任何废容量.我们可以理解一下,背包容量之和减去要放 M 个物品的体积代表的是放这 M 个物品最多能允许多少"废容量"的出现.当"废容量"大于这个极限即视为不可行方案。

B 交通改造

最小生成树问题。将路口视为点，道路视为边建图，裸的最小生成树问题。

C 丢手绢

由题目可知最多经过 T=N/gcd(M,N)次操作，每个人都会回到原来位置，据此可以将 10^k 次操作简化为 10^6 以内次操作，直接循环模拟求出答案。

D 敌情侦察

若已知 A 的第一格士兵部署情况，便可依次求出 2、3、……、N 格士兵部署情况，显然遍历所有的情况会超时，根据分析可知第一格士兵的兵力部署存在上界与下界，通过二分法求出上下界，即可得到可行的部署方法数。

E QQ 宠物

我们可以发现整张图把所有的边连起来是一个基环树，简单来说就是一棵树上加了一条边，使整棵树里有一个简单环，一般来说，遇到基环树要么缩环为点要么破环为链，
这道题我们在环上任选一条边，去掉这条边这就变成了一颗树，分别从这条边的两个端点做树 dp，最后取 max（不选左端点时的最大值，不选右端点时的最大值）
至于如何树 dp，我们给每个点设计两个状态，一个是选该点时的最大值 f[i]，一个是不选该点时的最大值 g[i]，所以 f[i]等于它所有子节点 g[i]的和加上它本身的权值，g[i]等于它所有子节点 max(f[i],g[i])的和，最后取去掉那条边的左端点和右端点的 g[i]的最大值就可以了。

F 分割草坪

首先可以用区间 dp 的方式，对于 dp[i][j]的求解，我们只要以 i，j 为底边，枚举顶点 k（i<k<j），划分出三角形（i，j，k），然后我们就将要求的值分为 dp[i][k]+dp[k][j]+三角形（i，j，k）的花费，找到花费最小的值就可以了。
之后可以发现，所有的三角形都有一个顶点为 1 的方案显然是最优的，推导一下公式就可以得出答案

G 数学大师

做这道题要对组合数的求法有深刻的理解。
首先这道题的核心是组合数的线性求法，通过组合数的阶乘式子 C(n,m)=n!/(m!*(n-m)!)我们可以很容易求出组合数的线性求法，下面给出四个式子
C(n,m)=C(n,m-1)*(n-m+1)/m;
C(n,m)=C(n,m+1)*(m+1)/(n-m);
C(n,m)=C(n-1,m)*n/(n-m);
C(n,m)=C(n+1,m)*(n-m+1)/(n+1);
组合数还有一个最常用的递推方式那就是 C(n,m)=C(n-1,m-1)+C(n-1,m)
所以我们推整个方阵的值可以通过每一行加和得出，
并且　第 n 行的值　等于　第 n-1 行的值*2+C(n-1,m-1)-C(n-1,m)
所以我们先推出第一排的和
然后维护一下每一排的和，这一排第一位的权值，还有最后一位的权值，这道题就可以通过在 O(x+y)的时间复杂度之内推出来
这道题还有一些其他的细节，比如 xy 为负数等等，稍稍判断一下就好，除法要转换成逆元，用费马小定理就可以解决。

H 小明喝奶茶

将奶茶店分别按照左端点以及右端点从小到大排序，遍历 1 到 n，用一颗权值线段树维护当前可用的方案数并每次查询前 k 小方案之和。

I 光棱塔

根据方差公式有$v*m^2=m\times\sum_{i=1}^{m}(x_i-x)^2=m\times(\sum_{i=1}^{m}{x_i}^2-\frac{n^2}{m})$，所以只需维护$\sum_{i=1}^{m}{x_i}^2$的最小值，重新定义 dp[i][j]为前 i 个节点安放 j 个光棱塔得到的相邻塔间距离平方的和的最小值

最后答案就是 dp[n][m]*m-n^2，状态转移方程为 dp[i][j]=min(dp[k][j−1]+(s[i]−s[k])^2)（s[i]表示第 i 个节点与前一个节点间的距离），复杂度为 O(n^3)。考虑状态转移方程的实际含义不难发现随着 k 值增大，dp[i][j]值会先减少再增加，可以通过二分法找到最低点，或是采用斜率优化得到最小值。

J AC 自动机

简单的二分答案并验证

K 方程求解

因为是一元一次方程，所以最终一定可以化成 kx+b=0 的形式，整理可得 x=-b/k，为了方便，将等号右边的多项式直接移到等式左边，即系数均乘 -1，可以用一个变量标记一下。

L 矩阵生成

首先开一组循环输入。然后将第一行中间的数赋值为 1，定义一个 x,y 表示当前位置，把 x 赋值为 1，y 赋值为(n+1)/2.
其次在开一组循环，根据 x,y 的值（即 K-1 的坐标)情况判断怎么填写数字，在填写数字完毕后将 x,y 的值赋为当前坐标。
最后当循环数到达 n*n 后结束并输出