A 每种座位都建两棵线段树，维护所有桌子，一棵表示向左走，一棵表示向右走；即一共 2\*m 棵线段树。

然后每个人向线段树对应节点连边；线段树叶子就表示座位；同一个桌子的座位之间连边即可。

向左走的线段树，从自己到左孩子连的边，费用需要加上 “跨过右孩子” 的代价。即到达向左走的线段树某节点，默认当前在该区间的右端点。所以每个人向区间连的 log 条边要注意一下初始费用。

需要多路增广费用流才能过。多路增广，就是 spfa 一次之后，根据 dis[ cr ] 和 dis[ v ] 的关系，像 dinic 一样走。 dinic 的优化都可以加（似乎一定要加当前弧优化？），注意要像 dfs 一样打 vis 标记，因为 dis 上可能有 0 环。

B 把所有人看做二分图中顶点Xi，所有工作看做二分图中顶点Yi，建立附加源S汇T。

1、从S向每个Xi连一条容量为1，费用为0的有向边。

2、从每个Yi向T连一条容量为1，费用为0的有向边。

3、从每个Xi向每个Yj连接一条容量为无穷大，费用为Cij的有向边。

求最小费用最大流，最小费用流值就是最少运费，求最大费用最大流，最大费用流值就是最多运费。

C 首先求出所有仓库存货量平均值，设第i个仓库的盈余量为A[i]，A[i] = 第i个仓库原有存货量 - 平均存货量。建立二分图，把每个仓库抽象为两个节点Xi和Yi。增设附加源S汇T。

1、如果A[i]>0，从S向Xi连一条容量为A[i]，费用为0的有向边。

2、如果A[i]<0，从Yi向T连一条容量为-A[i]，费用为0的有向边。

3、每个Xi向两个相邻顶点j，从Xi到Xj连接一条容量为无穷大，费用为1的有向边，从Xi到Yj连接一条容量为无穷大，费用为1的有向边。

求最小费用最大流，最小费用流值就是最少搬运量。

D

方法1

按左端点排序所有区间，把每个区间拆分看做两个顶点<i.a><i.b>，建立附加源S汇T，以及附加顶点S'。

1、连接S到S'一条容量为K，费用为0的有向边。

2、从S'到每个<i.a>连接一条容量为1，费用为0的有向边。

3、从每个<i.b>到T连接一条容量为1，费用为0的有向边。

4、从每个顶点<i.a>到<i.b>连接一条容量为1，费用为区间长度的有向边。

5、对于每个区间i，与它右边的不相交的所有区间j各连一条容量为1，费用为0的有向边。

求最大费用最大流，最大费用流值就是最长k可重区间集的长度。

方法2

离散化所有区间的端点，把每个端点看做一个顶点，建立附加源S汇T。

1、从S到顶点1（最左边顶点）连接一条容量为K，费用为0的有向边。

2、从顶点2N（最右边顶点）到T连接一条容量为K，费用为0的有向边。

3、从顶点i到顶点i+1(i+1<=2N)，连接一条容量为无穷大，费用为0的有向边。

4、对于每个区间[a,b]，从a对应的顶点i到b对应的顶点j连接一条容量为1，费用为区间长度的有向边。

求最大费用最大流，最大费用流值就是最长k可重区间集的长度。

E 首先判断从地球到月球是否存在一条路线，如果不存在那么无解，否则把每个太空站按照每天拆分成d个点，<i,d>表示第i个站第d天。建立附加源S汇T，顺序枚举答案Day。

1、对于第Day天，从S到<0,Day>连接一条容量为无穷大的有向边。

2、从<-1,Day>到T连接一条容量为无穷大的有向边。

3、对于第i个太空船，设第Day-1天在a处，第Day天在b处，从<a,Day-1>到<b,Day>连接一条容量为该太空船容量的有向边。

4、对于第i个太空站，从<i,Day-1>到<i,Day>连接一条容量为无穷大的有向边。

5、求当前网络最大流，如果最大流量大于等于地球上人数K，停止枚举，当前Day值就是答案。

F 如果ai和aj之间相差一个质数，那么它们一定是互为倍数关系且质因子分解后指数和只相差1

容易发现，相互匹配的两个数的指数和一定是一奇一偶  
所以这是一个二分图匹配问题

用最大费用最大流实现匹配即可

但由于费用和非负，我们在最大流统计流量时，如果当前答案加上本次新增费用小于0，就刚好取到不小于0的部分

由于先取大的一定比取小的好，所以这样的贪心策略正确

G 首先把棋盘黑白染色，使相邻格子颜色不同。把所有可用的黑色格子看做二分图X集合中顶点，可用的白色格子看做Y集合顶点。建立附加源S汇T，从S向X集合中每个顶点连接一条容量为1的有向边，从Y集合中每个顶点向T连接一条容量为1的有向边。从每个可用的黑色格子向骑士一步能攻击到的可用的白色格子连接一条容量为无穷大的有向边。求出网络最大流，要求的结果就是可用格子的数量减去最大流量。

H 大体跟D一致

区别：

1.边权要改，很明显啊，公式都给出来了。

2.两条线段能否同时选的判定规则要改，加多一条（当i和j同时垂直x坐标且x坐标相等时，不能相连）。

3.注意精度，开long long

I 首先将图黑白染色。

对于某个点，若它一定在最大匹配上，那么Bob必胜。因为Bob可以一直沿着匹配边都，Alice只能走非匹配边。到最后一定是Alice不能移动。

否则Alice必胜。这个我不会证，但是又举不出反例来qwq。手玩了几个数据发现Alice总会有一种方法走某个非匹配边干掉Bob。

那么如何找不一定在最大匹配上的点呢？首先求出一个最大匹配，结论是从所有不在最大匹配上的点开始dfs，通过交叉边(目标点的匹配边)走到点都是不一定在最大匹配上的点。(总有一种方案使这个点成为最大匹配)

然后直接匈牙利就行了

J 就是按照题目说的先跑一遍最短路，之后建出一张最短路DAG在上面拆点最大流就没了