A:

求一个边数尽量少的联通子图，并且最省钱，输出消耗。

直接最小生成树。

B：

有一个人在一个图上，要拆一些墙使得他可以到达任意一个区域，问最小花费。

分析：想象一张平面，我们这个平面只能有一个联通块，也就是说，如果墙形成了环，就被分成了两块。所以直接做最大生成树，不在生成树里的边都是要搞掉的。

C：

求一棵生成树，最大权值-最小权值最小。

分析：枚举最小的边重做mst，check min即可

D：

你可以选择买一些套餐，这样套餐里所有的边就是直接建好的，你也可以建一些边，然后搞成联通需要多少钱。边数很多点数不多。

分析：

我们只需要考虑n-1条边和套餐（证明显然）。枚举用了哪些套餐，先建好套餐内的边，然后考虑n-1条边还需要哪些，加上去，不断取最小。

E：

给一堆有向关系，问哪些可以相互形成一个圈圈。

分析：直接Floyd闭包

F：

从a点到b点， 找到一条路径，使得这条路径上的所有噪音中最大的值是所有路径中最小的

分析：最小瓶颈树类似，离线询问，加入一条边就看询问中哪些是又被连上了的，那么对于那个询问的答案就是刚加入的边.

G：

最短路，每条边有a,b属性，从出发开始，打开amin, 关闭bmin

思路：最短路的时候，边是开就过，关的就等开的时候过。

H：

A在点1，B在点2，有1e5个点，他们最多被分成1e6块，块块之间可有重叠点，一个块内的点之间消耗是固定的，问二者相遇的最短时间以及可能相遇在哪些点。

每一块整体当成一个点，然后向块内的点连价值的边，答案/2就是原本答案。

I:

按照dis建图，点也当洞洞跑最短路

J

K：

一个点拆成快慢，快慢分别拆成上下左右，跑最短路