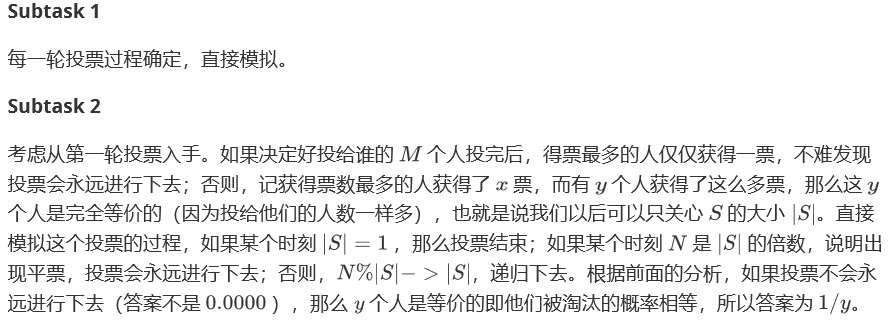
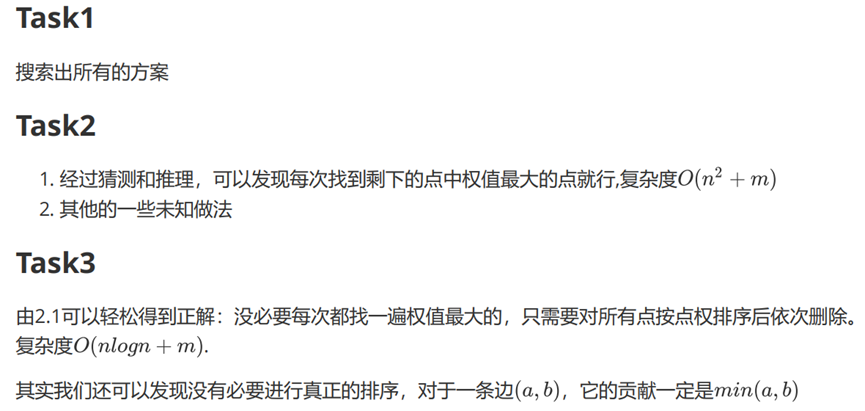
Solution 18

1. **投票**



**2、删点游戏**



**3、遨游**

对于题目描述中既有“最大”又有“最小”的题目，很多都是用二分，这题也不例外。

首先，省级路费是很容易算且是独立的，所以预处理出每条道路经过了省级优惠后是多少路费。

接着，这题要用到二分嵌套。第一层是二分L（l,r,mid），第二层是二分R（l1,r1,mid1），中间要做一次搜索。把所有在[L..R]区间内的边取出，做一遍搜索，如果能到达目的地则记录mid1，并将r1改小，否则将l1改大。第一重二分中如果能取到mid1则记录下mid与mid1，并将l1改大，否则将r1改小。

注意：如果想拿部分分，可使用二重循环或循环套二分。

**4、轨道**

n个正整数的乘积除以k的商不仅是正整数，还与k互质<-别粗心。

让你求这n个数的方案数。

20%的数据

这部分的数据非常水（送分），直接暴力递归每个数选什么即可。

时间复杂度（mn）

40%的数据

事实上，只要动动脑筋就知道可以用DP来做。

我们设f[i][j]表示i个数的乘积为j的方案数。

但这样做显然空间又大时间又大，所以我们可以换个状态。

我们考虑到乘积的关键部分只和k的约数有关。

所以我们设f[i][j]为前i个数，乘积与k的最大公约数为k的第j个约数的方案数（且乘积除以公约数与k互质）<-不标准的最大公约数。

思考：为什么与k的最大公约数一定是k的约数呢？

紧接着我们可以想到动态转移方程。

f[i][j]=sum(f[i-1][k]\*......)(a[j] mod a[k] = 0)

其中a数组为k的约数（以后也是如此）

其中的省略号到底是什么呢？

其实也就是f[1][a[j]/a[k]是k的第几个约数]！

这个DP的原理其实也就是把每个数的“贡献”形象化。

思考：为什么a[j]/a[k]一定是k的约数呢？

问题就变成了（我们称它为问题x）：在1~m内有多少个数与k的最大公约数为k的第i个约数（且其除以公约数与k互质）。

我们可以直接枚举1~m，再枚举k的约数即可。

时间复杂度：

处理问题x：(m\*k的约数)

处理DP：(n\*k的约数\*k的约数)

思考：问题x的原理是什么呢？

70%的数据

这时，我们的瓶颈在于如何处理问题x。

然后，容斥原理闪亮登场！

我们知道，



再想想，1~m内找有多少个与y互质，互质就是只要能整除以任何一个y的质因数就不行。那不就是容斥原理吗？

思考：公式是什么呢？

gcd'(1~m,k)=a[i]的个数

这个不标准最大公约数公式我们可以化简一下。

gcd'(1~m/a[i],k)=1的个数。

这不就跟我刚才说的一模一样吗？

我们可以用递归来实现容斥原理公式。

时间复杂度（问题x）()

思考：为什么最大公约数公式可以化简成这样子？

100%的数据

这时又出现了一个新的障碍，DP超时了！

我们可以发现，实现主要浪费在“k的约数个数的平方”上面。

所以我们可以试着简化一下。

我们重新看一下转移方程......

我们发现，既然a[j] mod a[k]=0，不就有种找约数的感觉吗？！

然后我们立刻想到找约数的套路——平方根！

然后，题目就解出来了！

时间复杂度（DP）(n\*<=20000的数)