

### A. 雷神之路

题目大意：一条一维路径，每次可以走一步，两步或三步，但不能走到地雷上。

想法：一开始直接用记忆化搜索，但 $1e18$ 的数据量明显TLE，所以考虑采用矩阵快速幂，但到地雷处必须停止结算，否则往下更新错误。

坑点：1.  $1e18$ 数据量，必须仔细读题。2. 地雷处结算必须细致，注意特殊情况的特判。

### B. Snowdrop修长廊

题目大意：用多个区间去覆盖所有点，每个区间的代价通过一个二次方式计算出来。

想法：如果到一个点时要用前面所有点去更新它，则时间复杂度为 $O(n^2)$ ，明显TLE。考虑进行剪枝，注意到，如果某一区间长度 $>w/2$ ，那么，将它归入任意区间性价比降低（即总代价上升），所以，大于 $w/2$ 的区间一定不再更新。

坑点：其实这道题目没什么坑点，但剪枝方法想出来具有一定灵活性，值得注意。另外如果此题另想办法的话似乎并不合理。

### C. Taosama与煎饼

题目大意：按照特定规则在一组数据中取数，求取到的最大总和。

想法：乍一看好想要对所有道具进行全排列，但注意到道具大小最多为4，并且道具使用顺序不限，所以直接考虑用简单搜索实现。

坑点：dp比较繁琐，注意初始化，中间过程要细致，不能犯弱智错误。

### D. 任务

题目大意：一堆任务，先来后到，可以在两台机器上运行，但时间不相同，求最短时间。

想法：一直TLE，但其实这就是积木模型，不过更新上略微繁琐。使用两个堆的差值进行更新，最大差值为通过贪心得到的两堆中的较大高度，不过 $O(n \cdot \max h)$ 的时间复杂度显然卡了大数据。

坑点：dp一个一个更新，非常繁琐，维数也较多；时间复杂度不容易降低。

### E. Goozy的积木

题目大意：一堆积木，搭成两个塔，求能搭出高度相同塔的最高高度。

想法：一开始懵了很长时间，最后发现这是套路题。 $dp[i][j]$ ， $j$ 为两塔高度差的绝对值， $dp$ 为总高度，0-1背包走一遍就结束了。

坑点： $i$ 若保持原状就超出空间了，注意到 $i$ 只能从 $i-1$ 得来，对 $i$ 做 $\%2$ 操作即可。

### F. 先锋看烟花

题目大意：一列烟花，在不同时间放出，每个点看某个烟花都有一个满意度，越近满意度越高，但人有固定移速 $d$ 。

想法： $dp[n]$ ，用所有可以到达 $i$ 的点对 $i$ 进行更新，做法非常简单。

坑点：注意到这里有个RMQ问题，如果用朴素算法时间上会TLE，考虑用线段树，每次更新完后重新建树即可。

### H. 又见背包

题目大意：多重背包

想法：多重背包一般做法，但由于数据量较大，采取队列优化。

坑点：无。

### L. 来签个到吧

题目大意：期望，分为两步操作，加球操作与摸球操作，求平均操作次数。

想法：加球操作即求取最大公倍数，若存在0可以特判，摸球操作即 $\sigma(N/i)$ ，两个操作次数相加即可。

坑点：一直莫名wa，后来发现 $N/i$ 改成 $(N*1.0)/(i*1.0)$ 之后精度提高，细节要十分注意。