

白色风车

预估：简单

这道题很简单，直接统计子串数量即可。

有机鱼

预估：简单

这道题是个简单数学题，求圆的最大内接凸多边形。直接把多边形拆成三角形来算就可以了，很多人错的原因是 π 取了3.1415926535甚至更少的位数，实际上在n较大的情况下，是会产生比较大的浮点误差的，可以直接用 $\arccos(-1)$ 来求 π 。

简单数学题

预估：简单

这道题确实是个简单数学题。

$$\begin{aligned}lcm(i, j) &= \frac{i, j}{gcd(i, j)} \\gcd(i, j)lcm(i, j) &= ij \\ \sum_{i=1}^n \sum_{j=1}^m ij &= \sum_{i=1}^n i \sum_{j=1}^m j \\ &= \sum_{i=1}^n i \frac{(1+m)m}{2} \\ &= \frac{(1+m)m}{2} \sum_{i=1}^n i \\ &= \frac{(1+m)m}{2} \frac{(1+n)n}{2}\end{aligned}$$

结果要取模，所以对2求个逆元就好了。

圣杯地牢

预估：中等偏难

这道题是一个校外同学出的思维题，其实题很好，但是因为数据问题，导致各种做法都可以过去。

正解是这样的：

令 $c=b-a$ ，先排除a的影响，问题转变为，能否对 $[0,0,0,...,0]$ 进行操作，变成序列c。

紧接着对c做差分得到d，一次操作令c上的数字循环依次多加1，例如从1开始进行操作，会令序列从 $0,0,0,0,0$ 变为 $1,2,3,4,5$ ，那么在差分空间的效果就是从 $0,0,0,0,0$ 变成了 $-4,1,1,1,1$ 。

考虑单次操作，c的所有元素和会增加 $\sum_{i=1}^n i = (n+1)n/2$ ，可以先对c求和再除以前面的式子，就可以得到总的操作数cnt。

假设以第i个位置为起点的操作数为 opt_i ，那么该点的差分值会增加 $(1-n) \times opt_i + 1 \times (cnt - opt_i)$ ，接下来我们就可以得到 $n+1$ 个公式： $(1-n) \times opt_i + 1 \times (cnt - opt_i) = d_i$ ，以及 $\sum_{i=1}^n opt_i = cnt$ ，判断公式是否有解即可。

$$\begin{aligned}(1-n)opt_i + cnt - opt_i &= d_i \\ cnt - n \times opt_i &= d_i \\ opt_i &= \frac{cnt - d_i}{n}\end{aligned}$$

Long may the sunshine

预估：中等

这道题一开始数据有一点小锅，放走了一个暴力的代码，之后换数据重测了。这题其实很简单的，要维护公差为2的等差数列，实际上就是维护两个最长连续字段的线段树就可以了。

弗林戒指

预估：中等

没想到这道题过的人这么少，本质就是一个二分+背包的问题。

要找最小时间，那就二分t，去check，每一次check都是一个 $O(100nm)$ 的dp，实际上这里根本不需要枚举所有n，因为每件武器重量最少为1，最多装备100件。dp的时候维护i件武器，j负重的时候的最大攻击力。至于弗林戒指，直接在dp完以后把j<28的再计算一次就可以了。

仿身泪滴

预估：中等

仿身泪滴这题是个概率dp，每次向后移动，如果遇到非0的值，可以转移，如果遇到0就有两种情况：模仿一个已经被杀的人；模仿一个没有被杀的人并且连带他一起死。

$$\begin{aligned}\text{令 } dp[i] \text{ 为前 } i \text{ 个位置必然被杀的人的期望, } dp[0] &= 0 \\ \text{对于 } a[i] \neq 0 \quad dp[i] &= dp[i-1] \\ \text{对于 } a[i] == 0 \quad dp[i] &= dp[i-1] + 1 + \frac{i-1-dp[i-1]}{i-1}\end{aligned}$$

365里路

预估：中等

这道题相当于建一个有向图，把每一个点拆成d个点，也就是每一座城市的每一天都认为是一个结点。对于每一条边，都拆分成d条边，分别从第t天的城市u连向第t%d+1天的城市v。然后对这张图跑一个tarjan缩点，变成一个dag，再dag上dfs跑一个结果出来就可以了。

缩点的时候要特殊处理，把每个连通分量中的所有城市可以取到的最大权值求和（同一个分量里可能有同一个城市在不同天数里的多个结点）。dfs的过程中不会遇到两个连通分量里有同一个城市的情况，如果第 t 天从一个城市 u 走到同一个城市的另一天，那么必然可以走这个环若干次回到一开始第 t 天的城市 u ，所以他们必然在一个连通分量中，不需要去重。

反方向的钟

预估：中等

这道题的做法其实很简单，首先可以考虑这个情况：

3 2 1,要把1放到3的位置，直接交换1、3和交换1、2再交换1、3付出的代价是一样的，但是后者会令序列变成1 3 2，如果要让他升序就需要再交换2、3。也就是说后者产生的代价是3，但是前者是2。后者这个额外的代价其实是来源于中间的2，它本来就在自己的目标位置，但是你强行把它交换到旁边的位置，所以它必须再造成一次消耗返回自己的目标位置。

对于给定的序列 a ，可以先对其进行排序，排序第关键字是 a_i 从小到大，第二关键字是初始编号从小到大。

这样我们就能知道给定的序列 a 中每一个元素的目标位置。我们依次找到最右边的、目标位置仍然在自己右边的一个元素，假设这个元素的编号为 i ，那么必然存在一个元素，其编号 $>i$ ，且他的目标位置 $\leq i$ 。找到第一个这样的元素，并交换之前找到的两个元素，这样的一次交换必然没有产生额外代价，因为大家都是朝着自己的目标位置进发的。重复如上操作就可以得到最优解了。

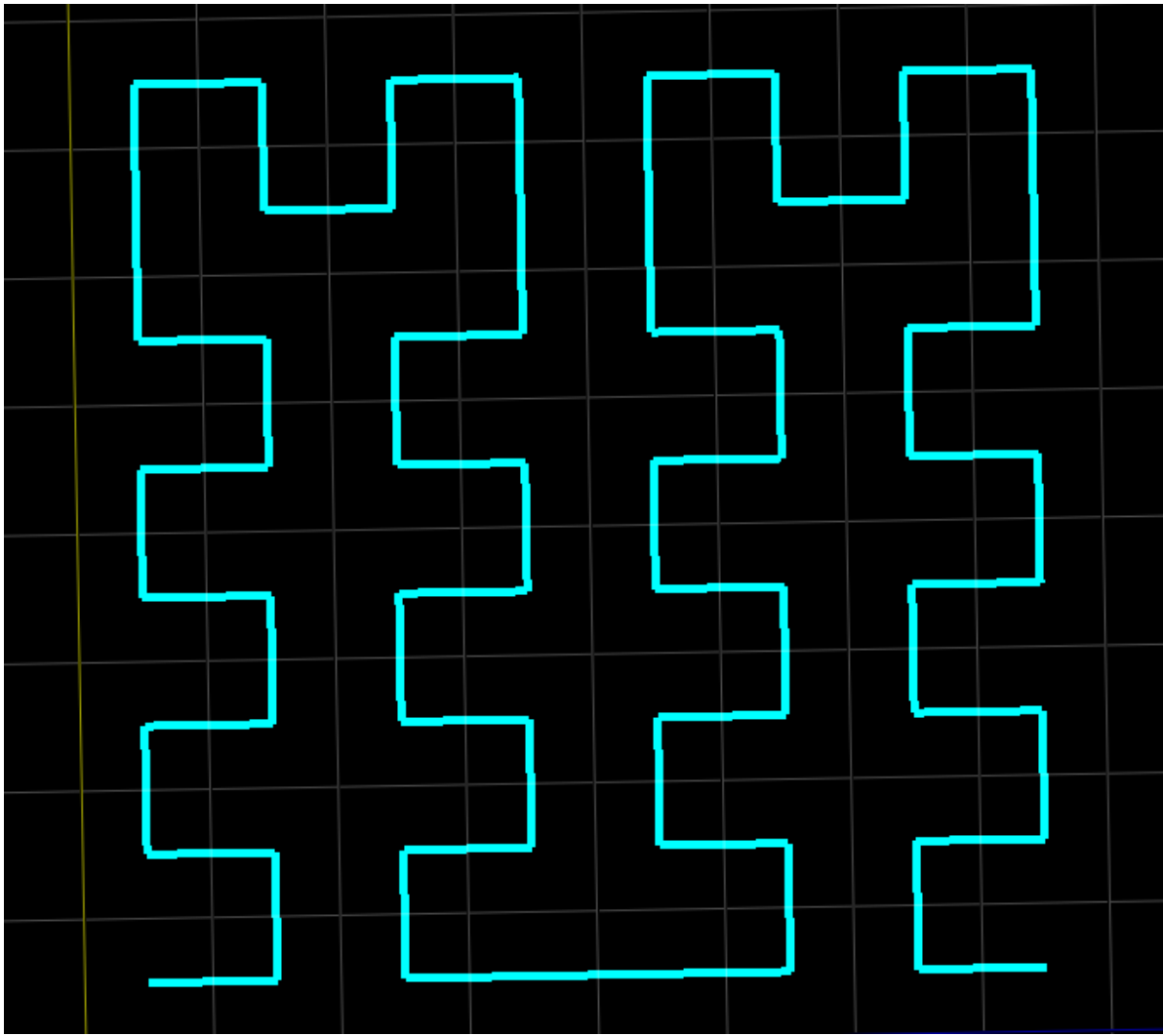
泥头车&泥头车Plus

预估：困难

这两题本质都是构造一条折线，折线的起点和终点给定，你要令折线上的转弯尽可能多，同时要保证折线不超过限定范围且折线不重叠。不同的是前者在二维平面而后者在三位平面。

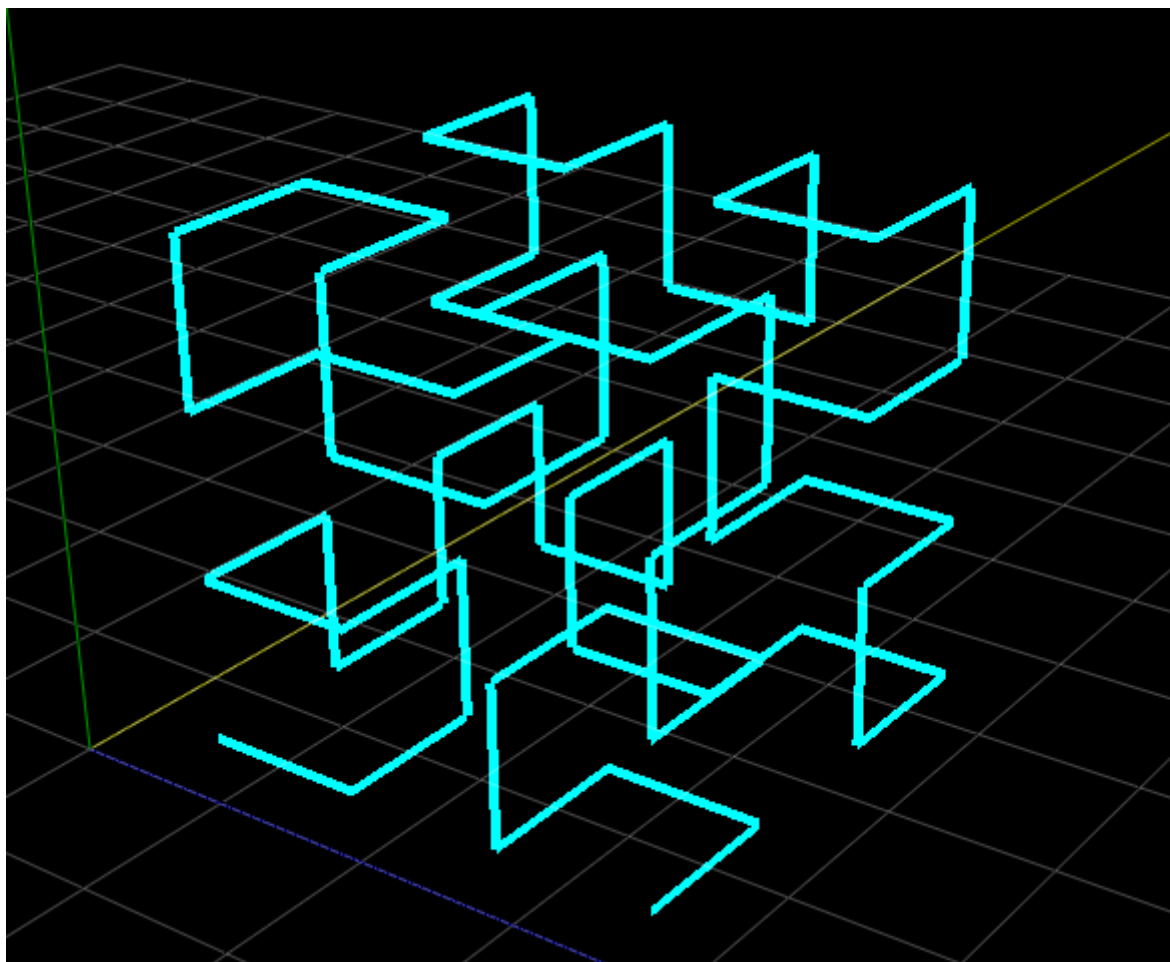
前者的通解如：

RULURULURULUURDRURDDLDRDLDRDLDRRRULURULURULUURDRURDDLDRDLDRDLDR



后者的通解如：

RUBDLUAUBUARBDARURDBULDBRUBDLUALBDLUADRDLDLDBURDARUBDRUADAULDAURD



后者其实不是唯一解，我这个解法是直接分治加暴力跑出来的，每次把空间分成八块，然后dfs跑一个联通顺序出来，这个联通顺序要保证你的第一步操作不能和外面一层的上一步操作相同，最后一步操作不能和外面一层的下一步操作相同，同层之间的7次操作各不相同，就能实现最优解。

星星坠落

预估：困难

这道题可以分成两块：树同构和dfs序建主席树

第一块就是判断树同构，我的std里是双哈希判断的，找到同构树以后再对每一个结点寻找满足条件的运输通道。

紧接着题目转变为，给定树上的若干点对，求每个点对被多少点对的简单路径完全包含。由于第一块的特性保证了每个点对的 u, v 都互不为祖先，所以直接查询一边在 u 的子树一边在 v 的子树中的所有点对即可。这个查询可以用主席树轻松实现，因为dfs序中，每一棵子树中的结点编号都是连续的。