Wannafly 挑战赛 7 题解

A. codeJan 与恐怖分子

可以根据 codeJan 的位置，将方格矩阵分成四个子矩阵分别考虑。对于每个子矩阵如果存在边长为 0 的话，就不用考虑。如果存在边长小于 K 那一定不能完成任务，否则依次按照行列炸毁。因为要炸毁所有区域且允许重复炸一个区域，所以对于 a ∗ b 的子矩阵行上需要$\lceil\frac{a}{K}\rceil$排炸弹，列上需要$\lceil\frac{b}{K}\rceil$排炸弹。注意结果会爆 int。

时间复杂度：O(1)

B. codeJan 与旅行

可以想象的是如果 m 足够大，codeJan 最后肯定会选择在相邻的两个城市来回走。所以可以枚举两个相邻的城市
(实际上应该是距离最小的两个城市)。并且直接”奔向”这两个城市的应该是最吼的！但是还要考虑，可能先往后退到 一个城市，再”奔向”枚举的城市。
举个例子就明白了：n = 3,m = 10, p = 2，三个城市的位置是 1 10 14。
那么应该先退回到 1，然后再在 10 和 14 之间来回走。
时间复杂度：O(n)

C. 小 Q 与氪金游戏

如果 x>=y，直接枚举氪金单数进行计算，否则不同的抽卡次数会对应不同的氪金单数，可以直接枚举抽卡次数进行计算, 只需要枚举足够多项即可达到要求的精度, 标程枚举了 10^6 项。

D. codeJan 与青蛙

先把所有的青蛙按照位置从小到大排序。下面用“代价”代称听到的 WA 次数。dp[i][j] 表示用 j 个黑洞收集前 i 个位 置青蛙的最小代价。枚举第 j 个黑洞的放置位置 k，可以得到状态转移方程：

$$\mathrm{dp[i][j]} = \min(\mathrm{dp[k-1][j-1]} + \mathrm{cost[k][i]})$$

其中 cost[k][i] 表示第 k 到 i 位置的青蛙全部被放在 k 位置上的黑洞收集的代价。 如果我们定义:

$$\mathrm{sum[p]} = \sum_{i=1}^{p} b[i], f\mathrm{sum[p]} = \sum_{i=1}^{p} a[i] * b[i]$$

那么可得：

$$\mathrm{cost[k][i]} = \mathrm{fsum[i]} - \mathrm{fsum[i]} - (\mathrm{sum[i]} - \mathrm{sum[k]}) * \mathrm{pos[k]}$$

因此可以通过 $O(nm^2)$ 的复杂度得到 dp[n][m]。 可以在 dp 过程中顺便记录第 j 个黑洞的收集数量和前一个状态的最少容量，进行比较更新即可。

E. 珂朵莉与 GCD

从一个点 r 开始往左端的 gcd 最多只会变化 logv 段
先预处理出每个点到一端的所有 gcd 变化的位置
然后把询问离线
扫描线右端点，维护所有左端点的答案即可
总复杂度 O(mlognlogv + nlogv)

F .Masha 与老鼠

这题最大的问题是∑ cj 太大，我们先把这个数降下来。 首先做两遍贪心，第一遍贪心，从左往右，每只老鼠进入它左边最近的一个有容量的老鼠洞，第二遍贪心，从右往左，每只老鼠进入它右边最近的一个有容量的老鼠洞。 那么，最优的决策一定是从每只老鼠都只进入这两次贪心中选择的位置。这样，∑ cj就可以降到 2n级别。 现在问题变成了，有n只老鼠和m个老鼠洞。每个老鼠洞容量均为 1。问每只老鼠都进洞的最小代价。

先考虑一个暴力 DP，把老鼠和老鼠洞放在一起安装坐标排序后，用$f[i][j]$表示已经决策完前i个老鼠和老鼠洞，前i个老鼠和老鼠洞中，选择的老鼠的个数-选择的老鼠洞的个数为j的最小代价。(对于一组匹配xi和pj，贡献为$-min(xi, pj)$和$max(xi, pj)$)。

对于$j > 0$ 的情况，此时老鼠比老鼠洞要多，那么如果第i个元素是老鼠，这只老鼠只能和坐标比它大的洞匹配，这只老鼠的贡献是$-xi$，从$f[i-1][j-1]$转移过来。如果第i个元素是老鼠洞，那么它肯定会和 $1...i-1$ 中没有匹配的老鼠进行匹配，贡献是pi，从$f[i-1][j+1]$转移过来。

对于$j < 0$ 的情况，此时老鼠洞比老鼠要多，那么如果第i个元素是老鼠，这只老鼠会和它之前的老鼠洞匹配，这只老鼠的贡献是xi，从$f[i-1][j-1]$转移过来。如果第i个元素是老鼠洞，那么它肯定会和$i+1...n+m$中的老鼠进行匹配，贡献是$-pi$， 从$f[i-1][j+1]$转移过来。

对于$j = 0$ 的情况，如果第i个元素是老鼠，那么它会从$f[i-1][j-1]$转移过来，贡献是xi，如果第i个元素是洞，那么此时就有两种转移，一种是$f[i-1][j+1]$转移过来，贡献是pi，一种是从$f[i-1][j]$转移过来，代价为 0(即不需要这个老鼠洞)。

暴力 DP 复杂度是$O(n+m)^2$的，但是我们分析后发现，$f[i][j]$只有$j = 0$ 才会需要决策，对于$j != 0$ 的状态，我们只需要把j全部+1 或−1，然后全体代价都加上一个值，不需要进行决策。所以我们就用两个栈维护，一个栈维护$j > 0$，一个栈维护$j \leqslant 0$。然后 再维护一个整体加法标记就可以了。