炼石计划 9 月 10 日 NOIP 模拟赛 #2 题解

第一题 数位

考虑 DP，设 $dp[i][0/1]$ 表示 $[1,i]$ 正确划分的方案数，其中最后一段是否合法。

记 $pre[i]=pre[i+1]+10^{n-i}\cdot s[i]$，那么区间 $[j,i]$ 合法当且仅当 $(pre[j]-pre[i+1])/10^{n-i}\equiv 0 \pmod D$

当 $\gcd(D,10)=1$ 时，相当于是 $pre[j]\equiv pre[i+1] \pmod D$，直接开个桶就好。

当 $\gcd(D,10)\neq 1$ 时，设 $D=2^x5^ym$，那么相当于要求 $\%2^x,\%5^y,\%m$ 同时为 0。对于固定的 i，在 $j\leq i-20$ 以后 $s[j]$ 的贡献 $\%2^x,\%5^y$ 都是 0 了，只用考虑 $\%m$ 的限制，此时同样可以对 $\%m$ 开桶。对于 $j>i-20$ 直接暴力即可。

时间复杂度 $\mathcal{O}(T\cdot 20\cdot n)$。

第二题 乘法

法 1

首先可能的幸运数字就不多，直接暴搜可以通过 $B\leq 11$ 的情况。注意在暴搜时记录当前还剩下哪些 0 到 $B-1$ 的没用过，只搜这其中的数字，并优先搜索较大的。一旦搜到 2 个 $\%n=0$ 的直接退出。如果 n 太大（例如当前所有数按从大到小放进去仍然 $<n$）就不继续搜了。

下面考虑 $B=12$ 的情况。其实部分分已经给出了很强的提示，对于 $n\leq 2000$ 我们可以采用 DP，记录当前 0 到 $B-1$ 没用过的数是哪些（2^B 种），以及当前 $\%n$ 是多少。对于 $n>2000$ 可以从大往小枚举 n 的倍数依次判断，注意这在答案为 -1 时非常慢，~~但答案是 -1 的通常 n 比较大~~，加个卡时就可以通过本题了。

当然如果有高明的搜索技巧，可能也可以直接飞过去！

法 2

来讲讲正常的做法。

枚举答案在 B 进制下的位数，然后折半搜索。即枚举左半边选哪些数，右半边选哪些数，然后再各自 permutation 计算 $\%n$ 的结果，最后排个序归并一下。

这种做法在不同数据下耗时差不多，可以轻松通过。

第三题 循环

首先所有边都保留肯定是合法的。考虑删掉一条边，那么所有 (a_i,b_i) 的路径方向都已经确定了。我们枚举删哪条边，然后把 P 个区间求个区间并即可，时间复杂度 $\mathcal{O}(nP\log n)$ 或 $\mathcal{O}(nP)$。

我们发现，从删掉 $(i,i+1)$ 移动到删掉 $(i+1,i+2)$ 产生的影响并不多，严格来说均摊变化量是 $\mathcal{O}(P)$ 的。开一个线段树，相当于每次做区间 $+1/-1$，查询全局 >0 的位置权值之和。

第四题 轰炸

其实每一步操作是独立的。假设一个位置 (x,y) 距离某个 cannon 的曼哈顿距离最小值为 $dis[x][y]$，那么第 D 步可选的位置是满足 $dis \le D$ 的 (x,y) 个数，再减去 D（之前已经填过的）。这是因为之前已经填过的，必然在这一步可选的范围之内。

而当 $D > n+m$ 时，$dis \le D$ 的 (x,y) 个数必然是 $n \times m$。因此可与分母上的 $(n \cdot m)!$ 消掉后面大部分项。剩下的问题就是求 $dis = D$ 的位置个数。n,m 较小时可以跑多源最短路，但本题 $n \times m$ 较大，K 反而较小。

我们考虑先计算出所有包含 K 个起点的行上所有点的 dis，也就是最多 $K \times m$ 个位置的 dis。这是好算的，我们枚举行 i，然后枚举起点 (x_1,y_1)，此时 $|x_1 - i|$ 是确定的，我们不需要关心 x_1 和 i 的大小关系。我们只要把 y_1 排个序，然后相应按顺序枚举 $1 \le j \le m$，跑个扫描线，就可以把 $|y_1 - j|$ 拆开计算出 $dis[i][j]$ 了。这一步的复杂度是 $\mathcal{O}(K \times m)$。

然后考虑计算相邻两个关键行之间的 dis。枚举 i 表示考虑行 $i-1$ 与 i 之间的间隔，以及列 j，我们要求这一竖段的 dis。容易发现，左右两端点分别是 $dis[i-1][j], dis[i][j]$，然后左端点向中间 +1，右端点向中间 +1，在最中间碰到。也就是:

$$dis[?][j] = \min(dis[i-1][j] + ? - lsh[i-1], dis[i][j] + lsh[i] - ?)$$

这里 $lsh[i]$ 表示实际的行号，~~lsh 就是离散化的缩写~~。注意需要特判 $i = 0, K$ 等情况。

于是可以 $\mathcal{O}(1)$ 计算出 dis 的上升和下降段分别是多少，在最终的结果数组里打一个差分标记即可。

时间复杂度 $\mathcal{O}(K \times m + n + m)$。

如果把 dis 数组都开出来的话访问太慢，开滚动会快很多，也就是一边求 dis 一边更新答案。