Summer 2018 Selection Contest 3 Solution

By mstczuo @ SYSU April 5, 2018

A. Temperature

一个集合 S 合法当且仅当对任一不在集合中的元素都存在一个子集其均值属于某个区间：

$$\lfloor \frac{\sum_{j\in R_i} t_j}{|R_i|} \rfloor \in [t_i - x_i, t_i + x_i] \ \ \forall i \notin S$$

考虑对问题解构：一个集合对元素 i 合法当且仅当 $i \in S$ 或者 S 是某个 R_i 的超集，而一个合法的集合对所有元素都合法。

枚举集合 S，如果 S 的均值属于 $[t_i - x_i, t_i + x_i]$ 这个区间就说明它对 i 合法，然后用 DP 将其传递到其超集：dp[S] 表示 S 能使得哪些元素合法, f(S) 也是一个元素集合，表示 S 的均值属于哪些元素。

dp[S] = $\{\cup_{S' \subsetneq S} dp[S']\} \cup f(S)$

$f(S)$ 可以暴力求，$dp[\cdot]$ 可以枚举 S 少一个元素的子集，这样就可以做到总复杂度 $O(n2^n)$ 了。

B. Primary Mathematics II

题目名字可以翻译成“小学生数学题”，说明这不是一道很难的题。

考虑对公式进行分块。（假设 n 是 m 的倍数）

$$\sum_{i=0}^{n}[k\cdot i\%p \in [A,B]] = \sum_{j=0}^{\lfloor n/m \rfloor}\sum_{i=0}^{m}[k\cdot(j\cdot m + i)\%p \in [A,B]]$$

暴力求出数组 X = $\{k \cdot i\%p, i \in [0, m-1]\}$ 并对其排序。考虑上面的式子，$k(jm+i) = kjm + ki$, 前面的部分是和 i 无关的变量，于是对于每个 j ，实际上我们要求 $Y_j = \{(x+d)\%p | x \in X, d = kjm\}$ 中有多少数在 [A,B]中。这一步可以用二分求出。

总的复杂度是 $(m + n/m)\log m$ ，取 $m = \sqrt{n}$ 即可。

C.Let it go

考虑对一个区间如何求解。很容易想到一个 DP，$f[i][j]$ 表示前 i 个字符能包含 "2018"/"2019" 的前缀长度最多为 j 且不包含 "2013"~ "2017" 的情况下最少删除的数字个数。这是一个5个状态的DP，每个数字 d 都可以提取出一个 5x5 的转移矩阵 $A^{(d)}$ ，然后定义一种特殊的矩阵乘法：$C = A \times B, C_{i,j} = \min_k A_{i,k} + B_{k,j}$。

那么问题就变成了对于 [l, r] 这个区间的矩阵的乘法。这样一来就很容易用数据结构维护了，例如线段树。复杂度为 $O(m^3(n+Q)\log n)$, 这里 m=5 表示状态数.

D.Permutation Counting

容斥原理，计算出有至少 i 个不合法位置的方案数 F_i 然后答案就是 $\sum_i (-1)^i F_i (n-i)!$ 。

F_i 的值有很多方法计算，这里讲两个方法：

(1) 将数字按照 %k 的值分组，对于每一组数 $A_i = a_0 + ik, a_0 \leq k$, $dp[i][j][p1][p2]$ 表示考虑了 $a_0 \dots a_i$ 这些位置上的数字，(p1, p2) 表示是否使用了 a_i, a_{i+1} 的情况下有 j 个不合法位置的方案数。

(2) 考虑将其看成一个匹配模型， X_i 和 Y_j 可以匹配当且仅当 $|i-j| \neq k$ 。考虑数组 $A_i = a_0 + ik, a_0 \leq k$, $< X_{a_i}, Y_{a_{i+1}} >$ 和 $< Y_{a_i}, X_{a_{i+1}} >$ 是不合法的匹配。容易观察出它们构成两条链：$X_{a_0}, Y_{a_1}, X_{a_2}, Y_{a_3} \dots$ 和 $Y_{a_0}, X_{a_1}, Y_{a_2}, X_{a_3} \dots$ ， 于是问题变成了在 $2n$ 个点构成若干链上选不相邻的边的方案数。

E. Matrix Plus

用十字链表维护节点之间的关系，在网格的外面多加一圈表示头尾。

如果用坐标表示节点， 节点 $(i,j), 1 \leq i \leq n, 1 \leq j \leq m$ 表示矩阵一开始位置 (i,j) 的元素 ，节点 $(0,j), (m+1,j)$ 表示行的头尾；节点(i, 0), (i, m+1) 表示列的头尾，定义 (0, 0) 为初始节点。

对链表上的边定义两种标记：

- 数值累加标记，相比前一个节点，当前节点的数值应增加多少。
- 位置变换标记，相比于前一个节点，当前节点四个方向的指针作了怎样的变换。实际上是两个布尔值，即左右是否进行了交换，上下是否进行了交换。

这两类标记相互之间没有影响，可以累加。

定义一个节点的状态为从初始节点出发，到达这个节点的路径中边的标记的累加值，在维护的过程中始终保证从初始节点出发经过任意的路径到达某个节点都可以得到相同的状态。

当我们要对一个矩形做某种变换的时候，不管是加减还是位置交换，很容易发现需要改变的指针只有进入这个矩形或者退出这个矩形的指针，矩形内的相对关系是不变的。所以这个时候只需要取出矩形最外面一圈节点，计算出新的状态，然后更新外面进入矩形和离开矩形的边的标记即可。这样复杂度为 $O(Q(n+m))$

这道题可以将加减改成乘除，将位置变换改成旋转90度。但是加减和乘除不能同时出现，因为乘除会影响加减的标记，不能只更新跨越矩形内外的指针，还需要更新矩形内部的指针，这样复杂度就不对了。