LMATRIX3 题解

by 吉如一

September 23, 2015

1 题目大意

现在有 m 个数组和一个整数 P，第 i 个数组用 A_i 表示，且由四个整数 F_i, C_i, D_i, L_i 生成，生成方式如下：

1. 数组 A_i 的长度为 L_i

2.$A_{i,0} = F_i$

3.$A_{i,j} = (A_{i,j-1} \times C_i + D_i) \mod P (1 \le j < L_i)$

接着，把这 n 个数组依次连接得到一个数组 B（即可以理解为 $B = \sum_{i=1}^{m} A_i$）。然后你需要对这个数组进行若干次操作，每一次操作你可以任意选择数组中的一个区间 $[L, R]$ 和一个整数 k 并把所有下标在 $[L, R]$ 中的数 A_i 变成 $(A_i + k) \mod P$。

问至少需要多少次操作才能把数组中的所有数都变成 0。

数据范围 $1 \le m \le 100$，$0 \le F_i, C_i, D_i < P \le 10$，$1 \le L_i \le 10^{15}$

2 第一步转化

因为题目中的操作时区间加，于是就可以考虑对数列进行差分操作（以下的加减法都在模 P 的意义下进行）。

设数组的长度为 n，令 $x_i = B_i - B_{i-1} (1 \le i < n)$（特殊地，令 $x_0 = B_0$，$x_n = -B_n$）。那么每一个操作就变成了在数组 x 中任选两个位置并把其中一个位置加上 k 另一个位置减去 k，数组 B 中的所有元素都变成 0 当且仅当数组 x 中的所有元素都变成了 0。

可以发现，问题等价成了：把数组 x 分成若干个和为 0 的子序列且每一个元素恰好出现了一次，最大化分成的子序列个数。证明如下：

假设已经知道了一个操作序列的所有 l, r 但是不知道 k，我们可以先建一张 $n+1$ 个标号为 0 到 n 的点。对于每一个 l, r，我们在点 l 到点 r 之间连一条无向边，这样最后得到了一张无向图。可以发现存在一个 k 值的选取方法使得这个操作序列是合法的当且仅当这张图每一个联通块的权值和为 0（这样在标 k 值的时候就可以选取一棵生成树然后从叶子递推上去得到每一条边的 k 值）。

因为把一个大小为 $size$ 的点集联通最少需要 $size - 1$ 条边，所以如果我们把这张图给划分成了 k 个和为 0 的点集，那么只需要 $n + 1 - k$ 个操作就好了。综上，问题就转化成了最大化划分成的和为 0 的子序列个数。

3 计算数组 x

因为问题已经被转化成了子序列，所以数字之间的相对顺序就不重要了。我们就可以用每一个数组出现了多少次来描述数组 x。因为模数 P 的范围非常小，所以即使每一个子数组的长度很大 10^{15}，但是数组的循环节不会超过 P。所以我们可以先处理出每一个子数组的循环节，然后将不在循环节的一段、第一个循环节、中间部分、最后一个循环节分开来计算得到数组 x。

4 划分子序列

接下来的讨论假定 $P=10$，因为其他的 P 的值都和这个类似且数据范围比它小。

首先 x 中为 0 的位置一定是划分成单独一个子序列最优，其次可以发现对于所有和为 P 的数对 19,28,37,46,55，肯定是要尽可能地把它们两两分到一组，用这种两元数对无论怎么调整都不会比它们单独放一起更优。

这样，数组中就至多只剩下了 5 种数（因为每一个数对最多只有一种剩下来）且其中有一种至多只剩下 1 次。接下来我们假定剩下来的数是 1 到 5。

可以发现剩下来这些数划分成的子序列一定满足和为 0 且不存在一个和为 0 的非空真子集。所以可以先预处理出所有可能的子序列，这样的子序列个数最多只有 30 个左右。我们对每一个子序列都设一个变量表示出现次数，那么就可以列出若干个约束条件，于是就变成了一个整数线性规划的问题。

我们可以尝试使用类似单纯形的方法，先随便求出一个解，然后逐步逼近最优解。逼近的方法如下：

每一次答案变得最优一定是把一些已经选取了的子序列合并然后分裂出更多的子序列。我们把一个分裂方法称为最简的当且仅当它不能被其他分裂方法线性表出。例如

(1, 3, 4, 4, 4, 4) + (1, 2, 2, 2, 3) –> (2, 4, 4) + (1, 2, 3, 4) + (1, 2, 3, 4)

就是一个最简的分裂方法，而

(1, 3, 4, 4, 4, 4) + (1, 2, 2, 2, 3) + (2, 3, 3, 3, 3, 3, 3) + (2, 2, 2, 2, 2) –> (2, 4, 4) + 2 (1, 2, 3, 4) + 3 (2, 2, 3, 3)

是一个分裂方法但不是一个最简的分裂方法，它可以被拆分成

(1, 3, 4, 4, 4, 4) + (1, 2, 2, 2, 3) –> (2, 4, 4) + (1, 2, 3, 4) + (1, 2, 3, 4) 和

(2, 3, 3, 3, 3, 3, 3) + (2, 2, 2, 2, 2) –> 3 (2, 2, 3, 3)。

可以发现最简的分裂方法是有限的，于是就可以尝试把所有这样的分裂方法都预处理出来，预处理可以用 DP 的方式进行，时间复杂度为 $O(lim^4 tot)$，其中 lim 为最简分裂方法中每一个数字出现次数的上界，tot 为可能的子序列个数。因为剩下的数只有 5 个，所以在 $P=10$ 时最多只有 2^4 种可能的情况，所以我们可以枚举所有的情况然后打表处理寻找一些性质。可以发现：1. 在所有可能的状态中最简的分裂方法至多只有 150 个左右。2. 在所有可能的状态中 lim 最大为 12。

所以我们就可以直接在程序中用 DP 求出所有最简的分裂方法。然后每一次对于当前选取的子序列，我们可以判断它是否满足某种分裂方法，如果满足，那么就把这种分裂方法进行到不能进行为止。这样在有限轮以后，一定可以到达最优解（因为每一次的答案都会减少）。

综上，我们解决了这个问题 OwO。