hiho 挑战赛 Solution

jcvb

October 30, 2018

1 XOR

这是一道非常容易的题目，只要分析每一位运算的所有情况就可以了。答案：如果 k 是奇数则无解，如果 $(k/2)$ and x 不为零也无解，否则答案就是 $2^{\mathrm{popcount}(x)}$，其中 $\mathrm{popcount}(x)$ 表示 x 的二进制表示中的 1 的个数。

2 GCD

这是一道比较基础的数论题。答案 (不要求 $i \le j$ 的情况）为

$$\sum_{i,j} \sum_{x|i,x|j} \mu(x) \sum_{y|a_i,y|a_j} \mu(y) = \sum_x \mu(x) \sum_y \mu(y) \left(\sum_k [y \mid a_{kx}] \right)^2$$

枚举 x 之后的计算量是 $\sum_k d(a_{kx})$（$d(n)$ 表示 n 的因子数目)。所以总的时间复杂度是 $\sum_i d(i)d(a_i) \le \sum_{i \le n} d(i)^2 \le O(n \log^2 n)$。

3 Dominos

先看数组 $rg[1..n], rb[1..n]$，简记 $row[i] = (rg[i], rb[i])$，表示每行分别有几个绿块和黑块。可以把它们拆成若干段，每一段 $i..j$ 可能有三种情况：

- $row(i) = \cdots = row(j) = (0, 0)$，即什么都没有。
- $row(i) = (1, 0), row(i + 1) = (1, 1), row(i + 2) = (1, 1), \ldots, row(j - 1) = (1, 1), row(j) = (0, 1)$，即 i 到 j 这些行放的都是上绿下黑的纵向骨牌（共 $j - i$ 张)。对称的情况类似（上黑下绿）
- $row(i) = \cdots = row(j) = (1, 1)$。这样子情况比较多，你可以把 $i..j$ 拆成若干小段，每段可以是两个纵向骨牌（一个上黑下绿一个上绿下黑)，也可以是一个横向骨牌。也就是相当于将 $j - i + 1$ 拆成 1 或 2 的小段，方案数可以用组合数计算。

对于所有 k 计算出有 k 个横向骨牌时的方案数（方案不唯一主要是由于有第三种情况)。然后对列数组 cg, cb 也做同样的处理。最后枚举横纵骨牌各有几个，把行和列的方案数乘起来，再乘以横纵骨牌分别全排列的数目（因为可以决定哪个每个纵向骨牌放在哪一列，每个横向骨牌放在哪一行)。

计算时可以使用分治 FFT 来加速，复杂度是 $O(n \log^2 n)$。

4 Items

用普通的 DP 做法，假设当前插入的物品重量是 a，转移式是 $DP[(x+a) \bmod M] \leftarrow DP[(x+a) \bmod M]$ or $DP[x]$，其中 DP 是一个 01 数组。这样是 nM 的复杂度。优化这个 DP 过程。在插入 a 的时候，拆成两部分更新，一部分是用 $X_1 := DP[0..M-1-a]$ 去更新 $Y_1 := DP[a..M-1]$，另一部分是用 $X_2 := DP[M-a..M-1]$ 去更新 $Y_2 := DP[0..a-1]$。用树状数组维护 DP 数组的 hash，可以用二分在 $\log^2 M$ 时间求出两个子串的 LCP。在用 X_i 子串去更新 Y_i 子串时，用 $(1+s_i)\log^2 M$ 时间可以求出所有 X_i 和 Y_i 不匹配的位置（假设有 s_i 个不匹配的位置）。这些不匹配位置 k 中，只有 $X_i[k]=1, Y_i[k]=0$ 的才可以进行有效更新，设有效更新有 t_i 个 $(t_i \le s_i)$。关键的观察是 $t_1+t_2=(s_1+s_2)/2$（这是因为 X_1X_2 和 Y_1Y_2 有相同数量的 0 和 1）。由于 DP 值只能由 0 变 1，总的有效更新数量 $\sum(t_1+t_2) \le M$，那么总的复杂度 $\sum((1+s_1+s_2)\log^2 n) = O((n+M)\log^2 M)$.