19CSP-S模拟赛十联测day6

A. Digit

等价于：找出所有 x 满足：对于所有整数序列 $a_{0..n-1}$ 有：

$\sum_{i=0}^{n-1} a_i P^i \ mod\ x = 0$ 等价于 $\sum_{i=0}^{n-1} a_i \ mod\ x = 0$

可以发现一个必要条件是 $P \ mod\ x = 1$，令 a_1 以外的所有 a 都为 0 就可以证明

然后可以发现 $P \ mod\ x = 1$ 意味着 $P^i \ mod\ x = 1$，所以这个条件也是充分的

所以答案就是 $P - 1$ 的约数个数

B. Gcd

暴力算法

$f[i][j][k]$ 表示考虑完 $b[1...i]$，$gcd = j$，不同的位置有 k 个的方案数

暴力转移是 $O(n^4)$ 的，稍微整一点优化就能到 $O(n^3 \log n)$ 了，例如添加了新的数后 gcd 肯定会变成约数，我们可以预处理出 $g[i][j]$ 表示有几个数 x 满足 $gcd(i, x) = j$，因为 $j|i$，所以复杂度变成了 $O(n^3 \log n)$

标准做法

假设 $f(d)$ 是我们要求的答案

令 $g(T) = \sum_{T|d} f(d)$

$g(d)$ 相当于是把第二个条件改成 $d|gcd(b[1...n])$ 的答案，也就是对于每个 i 有 $d|b[i]$

如果能求出 $g(d)$ 的话，根据莫比乌斯反演，有 $f(d) = \sum_{d|T} g(T)\mu(T/d)$，就可以在 $O(n \log n)$ 的时间内算出答案

k=0

当 $k = 0$ 时，相当于没有条件 3 的限制，那么 $g(d) = (\lfloor \frac{m}{d} \rfloor)^n$，因为每个 $b[i]$ 只要是 d 的倍数就行了

k!=0

一个大致的想法就是：有些 $b[i]$ 的选项个数仍然是 m/d，但有些就是 $m/d - 1$，因为不能等于 $a[i]$

考虑如果恰好有 k 个 i 满足 $a[i] = b[i]$，那么这些 i 必须要满足 $d|a[i]$

我们求出 $cnt[d]$，表示有几个 i 满足 $d|a[i]$，那么我们的方案数可以这样计算：

首先从 $cnt[d]$ 中选出 k 个，令它们满足 $a[i] = b[i]$，那么满足 $d|a[i]$ 的那些里面剩下 $cnt[d] - k$ 个的方案数是 $m/d - 1$，而不满足 $d|a[i]$ 的 $n - cnt[d]$ 个的方案数是 m/d

所以方案数是 $C_{cnt[d]}^{k}(m/d-1)^{cnt[d]-k}(m/d)^{n-cnt[d]}$

而题目要求的是至少有 k 个满足 $a[i]\neq b[i]$，转化后就是至多有 k 个满足 $a[i]=b[i]$

观察方案数的式子可以发现：必须有 $k\le cnt[d]$，不然方案数就必然为 0，所以我们暴力枚举 k 的话总枚举次数就是 $\sum_{d=1}^{m}cnt[d]$，大概 $O(m^{1.5})$ 级别，实际上严重不满

然后我们把快速幂给优化掉，时间复杂度就变成 $O(m^{1.5})$ 了，可以通过本题

C. Numbers

长短不同的情况

这个价值其实比的就是字典序大小，且长的序列的价值严格高于短的序列

序列长短不同这个情况我们可以单独计算，大概只要对每个序列计算出长度为 i 的不同子序列有几个就行了：

核心优化

令 $f[i][j]$ 表示现在长度为 j，且末尾是 i 的不同子序列有几个

由于有子序列不同这个限制，所以我们在转移的时候，对于同一个数字我们需要选后面第一个，这样就可以避免算重的问题，即每个子序列都刚好在最前面出现的地方被算到。

一种暴力是直接枚举 k，然后判断一下 $a[i+1...k-1]$ 中是否还有 $a[k]$，这样就是 $O(n^3)$ 的

还有一种暴力是枚举下一个数是 y，然后预处理出 $a[i+1...n]$ 中第一个 y 的位置，然后转移过去，是 $O(n^2m)$ 的

我们可以思考一下：若 $f[i][j]$ 可以转移到 $f[k][j+1]$，那么充要条件就是 $a[i+1...k-1]$ 中没有 $a[k]$

所以令 $p[i]$ 表示 $a[1...i-1]$ 中最后一个 $a[i]$ 的位置，那么我们就有：

$f[k][j+1]=\sum_{i=p[k]}^{k-1}f[i][j]$

这个优化是非常自然的，之后就预处理出 p，然后边算 f 边维护前缀和，就可以 $O(n^2)$ 计算了

大致思路

接下来考虑长短相同的，那这两个子序列肯定先是有一个前缀完全相同，然后在某一位有 $S_i>T_i$ 后，后面就可以乱放了

我们可以对这三种情况都开一个 dp 数组，分别为 $f[i][j],g[i][j],h[i][j]$

$f[i][j]$ 的转移跟上面的情况类似，就是要求 $a[i]=b[j]$，有：

$f[i][j]=\sum_{i_1=p_a[i]}^{i-1}\sum_{j_1=p_b[i]}^{j-1}f[i_1][j_1]$

整个二维前缀和就好了！

$g[i][j]$ 就是要求 $a[i]>b[j]$，且有：

$g[i][j]=\sum_{i_1=p_a[i]}^{i-1}\sum_{j_1=p_b[i]}^{j-1}f[i_1][j_1]$

$h[i][j]$ 什么都不要求，且从 g 和 h 转移过来，方程和上面两个类似，都能二维前缀和

时间复杂度：$O(n^2)$