20zr省选线上集训day2

> 洪华敦 北京大学

A. 【20省选集训day2】调兵遣将

首先考虑如何计算总的合法方案数量

一个比较大力的方法是动态规划，令 $f[L][R]$ 表示 $1...R$ 的合法方案数量，且满足 $L...R$ 是一个兵团

那么有 $f[L][R]=\sum_{x\le y<L} f[x][y]$，且 $gcd(w_{x...y})=gcd(w_{L...R})$

容易得到这样一个 $O(n^4)$ 的算法

考虑在本题中如何计算答案

我们取个补，变成计算有几种方案不包含该士兵

类似 $f[L][R]$ 我们整一个 $g[L][R]$，表示的是 $L...n$ 的合法方案数

那么我们可以枚举一个方案中相邻的两个兵团 $(L,R),(x,y)$，这两个兵团相邻的方案数是 $f[L][R]g[x][y]$，且 $R+1...x-1$ 不在其中

于是我们就得到了一个 $O(n^4)$ 的算法

优化

我们考虑如何去掉 gcd 的限制

令 $gcd(l,r)$ 表示 $gcd(w_{l...r})$

一个经典的结论是，当我们固定 r 后，$p_l=gcd(l,r)$ 可以分成 $O(\log(N))$ 个连续的段，使得每一段里的数相同

于是我们可以得到 $O(n\log N)$ 个形如 R,l_1,l_2,w 的段，表示对于 $L\in[l_1,l_2]$，有 $gcd(L,R)=w$

我们将 w 相同的段放在一起去做，考虑怎么去计算总的方案数

令 $f[x]$ 表示 $1...x$ 中合法的方案数

那么对于一个段 R,l_1,l_2，我们转移有 $f[R...n]+=\sum_{L=l_1}^{l_2} f[L-1]$

这显然是可以用数据结构维护的，对于一个 w，如果段数为 K，那么复杂度就是 $O(K\log n)$，总的段数之和是 $O(n\log n)$

所以我们可以在 $O(n\log^2 n)$ 的时间内计算总方案了

标准算法

我们借鉴一下 $O(n^4)$，一个比较自然的思路就是：

- 枚举 w，取出它的 K 个段进行 DP

- 对于 x，算出 $1...x-1$ 的方案数和 $x+1...n$ 的方案数，就可以加到 x 的答案上

我们类似 $f[x]$ 维护一个 $g[x]$，表示 $x \ldots n$ 的答案，同样要取出类似 (L, r_1, r_2, w) 的段

那么就是要对 $x \in [1, n]$ 都令 $ans_x = ans_x + f[x-1]g[x+1]$

我们可以发现，对于 $f[x]$ 来说，如果 x 不是这 K 个段中的某个 R，那一定有 $f[x] = f[x-1]$

同理如果 x 不是某个 L，则 $g[x] = g[x+1]$

所以我们把这若干段的端点，也就是 (R, l_1, l_2, w) 的 R 和 (L, r_1, r_2, w) 的 L 全拿出来离散化一下，然后每个空隙的贡献一起算就行了

时间复杂度：$O(n \log^2 n)$

B. 【20省选集训day2】一掷千金

SG 值的计算

这是经典的翻棋子类游戏，我们有以下两个结论：

- 我们可以把一个局面看成若干个独立的游戏，每个游戏中只有一个棋子是白色的，我们设这类局面叫 $W(x)$，其中 x 是白色的棋子
- 当我们把 x 的颜色翻转时，可以视为新加了一个 $W(x)$，因为颜色翻转其实是把颜色异或上 1，而两个 $W(x)$ 会抵消，符合异或的性质

所以问题就变成了计算每个白色棋子的 SG 值，然后异或起来就可以得到答案

对于 $W(x)$，设它的坐标为 (i, j)，我们打表可以发现 $SG(W(x)) = lowbit(max(i, j))$

所以相当于对这些矩形的并，求里面每个点 (x, y) 的 $lowbit(max(x, y))$ 的异或值

我们考虑类似矩形面积并去计算这个东西，枚举 x，用线段树维护 $lowbit(y)$ 的异或和

考虑计算 x 的答案，首先算一下 $y < x$ 时的覆盖长度，这一类的贡献是 $lowbit(x)$，之后相当于计算 $y > x$ 时的 $lowbit(y)$ 的异或值

我们就用线段树维护，其中遇到的一个问题是给定 l, r 要计算 $lowbit(y)(y \in [l, r])$ 的异或值

我们建线段树时对 $[0, 2^{30})$ 建，这样我们需要计算时所有的 l, r 都满足他是形如 $a2^b \ldots (a+1)2^b - 1$ 的形式

这种情况下，如果 $lowbit < 2^{b-1}$ 的话，例如是 2^x，我们把 $x+1...b-1$ 位翻转后得到一个同样 $lowbit$ 是 2^x 的数和他抵消

所以答案就是 $lowbit(a2^b)\ xor\ 2^{b-1}$

时间复杂度就是矩形面积并的时间复杂度：$O(n \log n)$

C. 【20省选集训day2】树拓扑序

令 $dp[x][pos][w]$ 表示，子树 x 有几个合法的序列满足 w 在第 pos 个位置

那么我们合并两个子树计算逆序对贡献时，一个想法是枚举右边子树的 pos, w，然后枚举合并完序列后，w 前面有 c 个左子树的点，这样的话产生的逆序对个数就是 $\sum_{i\le c,j>w} dp[left][i][j] + \sum_{i>c,j<w} dp[left][i][j]$

这个东西可以二维前缀和处理出来

然后这样合并的方案数就是 $dp[right][pos][w]C_{c+pos-1}^{c}C_{sz[right]-pos+sz[left]-c}^{sz[left]-c}$

更新 dp 数组其实也是类似的方法

这样的总复杂度是 $O(n^3)$ 的，因为枚举 pos 和 c 这两步总复杂度是 $O(n^2)$ 的，因为是 $sz[left]sz[right]$ 的。然后枚举 w 有 $O(n)$ 的复杂度

时间复杂度：$O(n^3)$