

The 2nd yuzusoft cup Stage 2: Zhanjiang Tutorial

朝武・芳乃

Mar 29th, 2024

1 杜老师 (dls)

假设 $n = R$ 。

我们可以把第 i 个质数 p_i 视作一个二进制位，一个数 n 对应一个二进制数 $f(n)$ ，其中 $f(n)$ 的第 i 位为 1 当且仅当 $V_{p_i}(n)$ 是奇数 ($V_p(n)$ 表示正整数 n 中含 p 因子的数量)。

那么一个集合 S 是合法的当且仅当: $\text{XOR}_{s \in S} f(s) = 0$ ，这个部分可以通过线性基解决，时间复杂度是 $\mathcal{O}(\frac{n \times \pi(n)}{w})$

这样子显然过不去，我们根据 2022 联合省选 D2T1 的套路，我们把 $> \sqrt{n}$ 的质因子单独处理 (这样的质因子只有一个)，复杂度就能做到 $\mathcal{O}(\frac{n \times \pi(\sqrt{n})}{w})$ 。

我们发现合适不能消掉 n 这个大头，我们充分开发人类智慧，发现 $R - L + 1$ 太大的时候 (大概是 $2\sqrt{n}$ 的时候) 每个质数位的线性基都会被塞满，我们只需要统计 $L \sim R$ 这个区间出现的质因子个数。

复杂度可以做到 $\mathcal{O}(\frac{\sqrt{n} \times \pi(\sqrt{n})}{w})$ ，随便过。

2 连通图计数 (count)

如果 $m = n - 1$ ，这个玩意就是 prufer 计数。

如果 $m = n$ ，也就是基环树，我们按照类似圆方树那样对环建一个方点，那么环上点断开连通块数量又变成了度数，直接上 prufer 即可。

如果 $m = n + 1$ ，分为两种情况：

- 分成两个环，我们还是可以 $m = n$ 那样做，不同的是我们需要枚举两个方点的度数；
- 否则，我们 > 1 的连通分量是唯一的，我们对这个双连通分量建方点，我们只需要知道这个联通分量里面有多少种方案即可。具体的，这个双连通分量里面必然会存在两个“汇点” u, v ，会存在三条连接 u, v 的路径。我们只需要枚举其中一条路径的大小，通过排列组合算出 2,3 路径的方案数，然后答案除以 6 即可。

时间复杂度： $\mathcal{O}(n)$ 。

3 原神攻略 (genshin)

显然我们只关心长度为 $k, k+1, \dots, 2k-1$ 的子串 (因为一个 $\geq 2k$ 的合法串可以拆分成若干个长度为 k 的串和一个长度介于 $[k, 2k-1]$ 之间的串)。

考虑 dp, f_i 表示 $s[1, i]$ 能否被表示, f_i 可能从 $f[i-2k+1, i-k]$ 当中转移过来, 也就是我们只需要关心前 $2k$ 个值, 我们就可以做到类似于矩阵那样做了。

由于这个矩阵乘积是 OR 乘积。我们可以通过 int 压位做到 $\mathcal{O}(k^2)$ 。

对于修改操作 $t[l, r]$, 我们可以先对 s 建一个 SAM, 然后对 $t[l-2k, r+2k]$ 直接跑 SAM 求这一段的 lcs 即可 (因为我们只关心其与 $2k$ 的 min 关系)。

我们进行动态 dp, 一共会进行 $L + 4qk$ 个点, 但是我们只会进行 $\mathcal{O}(q)$ 次操作, 所以 pushup 操作只会进行 $\mathcal{O}(q \log m + L + 4kq)$ 次, 是能够接受的。

时间复杂度: $\mathcal{O}(k^2(q \log m + L + kq))$ 。稍微卡卡常数就过了。

4 醒来 (wake)

首先考虑对 $0 \sim 2^n - 1$ 的情况。不难发现构造是极其容易的，大概可以使用分治。我们尝试去加强这个结论，下面说明我们可以做到构造一个以 0 开头， k 结尾的合法序列 ($k \neq 0$)

证明如下：

- 如果 $k < 2^{n-1}$ ，构造 $[p_0, p_1, \dots, p_{2^{n-1}-1}]$ 为 $0 \sim 2^{n-1} - 1$ ，且以 0 开头， k 结尾的数列。那么，我们只需要往 $(p_0, p_1), (p_2, p_3), \dots$ 中间插入左右两个数加上 2^{n-1} 即可满足。
- 如果 $k \geq 2^{n-1}$ ，构造一个以 0 开头，1 的 $0 \sim 2^{n-1} - 1$ 的排列，然后构造以 1 开头， $k - 2^{n-1}$ 的 $0 \sim 2^{n-1} - 1$ 的排列，把后面排列每个元素异或 $2^{n-1} + 1$ 再拼接起来即可。

这说明任意不同的数作为开头，结尾都是可行的。

然后考虑 $0 \sim x$ 这种情况，如果 $\text{popcount}(x)$ 为奇数就把 x 染黑，否则染白色。那么我们说明：我们可以找到一个以 x 开头的合法序列，当且仅当 x 的颜色的数量不少于 x 的反色的数量。

证明如下：假设我们需要以 x 为开头，并且假设 $n = 2^k + r$ ($r < 2^k$)， $p(n)$ 表示 $\text{popcount}(n)$ 的奇偶性，那么：

- 如果 $x < 2^k$ ，那么先随便构造一个以 $f(r)$ 为结尾的 $0 \sim r - 1$ 的合法排列。然后构造一个以 $f(r)$ 开头， x 结尾的 $0 \sim 2^k - 1$ 的合法排列，把前面的排列集体加上 2^k 即可。
- 如果 $x \geq 2^k$ ，那么构造一个以 $x - 2^k$ 作为结尾的 $0 \sim r - 1$ 的合法排列，假设开头是 t ，构造一个 t 结尾的 $0 \sim 2^k - 1$ 的合法排列即可，前面的加上 2^k ，把后面的拼接上前面的即可。

最后考虑区间的情况，我们把 $l \sim r$ 之间所有数二进制下的 lcp 去掉，这样子就必然存在一个 k 使得 $l < 2^k \leq r$ 。

我们不难发现如果满足一下两种条件之一是不可能存在合法序列的：

- 黑白数字相差 ≥ 2 ；
- $l + 2^k > r$ (这样子 $[l, 2^k - 1]$ 和 $[2^k, r]$ 不连通)。

如果这两个条件都不满足，我们总可以构造一下两对合法排列之一：

- 以 l 结尾， $[l, 2^k - 1]$ 的排列拼接 $l + 2^k$ 开头， $[2^k, r]$ 的排列；
- 以 $r - 2^k$ 结尾， $[l, 2^k - 1]$ 的排列凭借 r 开头， $[2^k, r]$ 的排列。

这些构造总能转成 $l = 0$ 的形式。

于是这题就做完了，时间复杂度 $T(n) = T(n/2) + \mathcal{O}(n) = \mathcal{O}(n)$ 。

5 寻宝 (treasure)

你发现一个元素不会取超过 $\mathcal{O}(\sqrt{a_n})$ 次（否则直接取一次 n 即可）。这样子，我们直接 dp 即可。可以用一下斜率优化或者单调性优化。

6 狂氣沈殿 (crazy)

最简单的 ynoi。

考虑分块，每个点往上跳 k 次的过程可以直接树剖（因为总共只会经过 $\mathcal{O}(\log n)$ 次非重边）。散块直接暴力，关键是怎么维护整块的答案。

对每个块维护有效点集合 S ，维护一个是否访问的 bool 数组 `flag`。对于整块集体向上跳的操作，只需要对每个 $s \in S$ 往上跳。如果 `flag[fa[s]]` 已经访问就不加入，否则把 `flag[fa[s]]` 设为 true 然后加入 `fa[s]`。这样子所有时刻 S 集合的大小总和是 $\mathcal{O}(n + q)$ 的，足以通过。

时间复杂度： $\mathcal{O}((n + q)\sqrt{n})$ 。