

abc418_d Substr Swap 题解

题目大意

给你两个长度为 n 的字符串 s 和 t 。

接下来有 m 次操作，每次操作给你两个整数 l 和 r ，要求将 $s[l..r]$ 和 $t[l..r]$ 替换。

输出 m 次操作结束后的字符串 s 。

解题思路

首先，我们可以开一个 a 数组。

并且令 a_i 的含义如下：

- $a_i = 1$ 表示 s_i 和 t_i 交换了奇数次；
- $a_i = 0$ 表示 s_i 和 t_i 交换了偶数次。

因为：如果 s_i 和 t_i 交换了偶数次，相当于没交换；交换了奇数次，相当于交换了一次。

所以对于每次操作的 l 和 r ，相当于对所有 $l \leq i \leq r$ 令 a_i 异或上 1。

但是如果每次都暴力处理的话，时间复杂度会来到 $O(n \cdot m)$ 。

所以考虑 **差分**。

定义 $c_i = a_{i-1} \oplus a_i$ （这里 \oplus 是异或符号）。

然后我们会发现， a_i 就是 c_1, c_2, \dots, c_i 的异或和。即：

$$a_i = c_1 \oplus c_2 \oplus \dots \oplus c_i$$

然后，对于每次操作的 l 和 r ，原问题是

- 将 a_l, a_{l+1}, \dots, a_r 都异或上 1

上述操作等价于

- 将 c_l 和 c_{r+1} 异或上 1

这样，单词操作的时间复杂度就降到了 $O(1)$ 。

最后，对 c 数组跑一下前缀异或和就能够得到对应的 a 数组。

然后根据 a_i 的值输出对应的第 i 个字符即可：

- $a_i = 1$ ：输出 t_i
- $a_i = 0$ ：输出 s_i

时间复杂度 $O(n + m)$ 。