

# BITACM 俱乐部 2023 年暑假集训 Rating#4

## Solution

2023 年 8 月 12 日

### A 拼回文

法一：Manacher

很显然的一点是，可以先回文匹配前缀和后缀，直到找到一个  $s_l \neq s_r$  的位置，此时问题转化成了找  $s[l, r]$  这个字符串的最长回文前缀和最长回文后缀。于是可以对  $s[l, r]$  这部分做 Manacher 找到  $p_i = i$  和  $p_i + i = \text{length}(s[l, r])$  的位置，两者比较选择较长的一个即可。

法二：Hash

随机数据，不卡 hash，欢迎乱搞（

### B 拼回文签到

上一题的签到版。

### C Palindromic

cf 原题网址

洛谷题解

### D 大哥叫我出题

可以看这个的题解 <https://www.cnblogs.com/zxhl/p/7227643.html>

### E 你要进入……吗

由于赛前多加了一组比较大的数据而忘了开大空间，导致大家 MLE，而且由于有人较快过题而没有及时发现这个问题，给大家带来了较高的罚时，非常抱歉。

首先我们可以知道， $\bigoplus_{i=l}^r a_i = (\bigoplus_{i=1}^l a_i) \oplus (\bigoplus_{i=1}^r a_i)$ 。那么我们可以来维护一个前缀异或和数组  $sum$ 。

我们用  $l_i$  表示从左往右到第  $i$  位时的区间最大异或和， $r_i$  表示从右往左到第  $i$  位时的区间最大异或和，那么显然我们有  $l_i = \max\{sum_L \oplus sum_R, 1 \leq L < R \leq i\}$ ， $r_i$  同理。最后  $ans = \max\{ans, l_i + r_{i+1}\}$ 。

暴力跑的话显然难以通过，因此我们可以用字典树来维护。将  $sum$  数组插入字典树中，枚举  $a_i$  并查询字典树中与它异或的最大异或值即可。

## F 签到题

真的是签到题 [www](#)

可以用 manacher 搞出所有回文子串的长度（搞出每个位置为中心最长的回文串即可，其他的都是-2 的关系），子串作为物品，然后就是一维数量一维长度和的二维多重背包问题。

## G 重建海拉鲁

由于修改的是子串的后缀，也就是前缀是相同的，问题可以转化为对集合 A 和 B 的子串进行匹配，使其公共前缀的长度和最大，即

$$\sum_{i=1}^n lcp(A_i, B_i)$$

最大 SAM 只能处理后缀，可以用反串建立 SAM 处理。两个子串的最长公共后缀等于其在 parent tree（也有人称为后缀链接）上的最近公共祖先。记  $sum$  为当前公共前缀的长度和，当匹配至节点  $i$  时，有  $s1_i$  个 a 的 len 大于等于  $k$  的字串，有  $s2_i$  个 b 的 len 大于等于  $k$  的字串没有在  $i$  的儿子节点被匹配，有  $\min(s1_i, s2_i)$  对 a, b 字串的 LCA 为节点  $i$ ，更新  $sum$ ，剩下的部分则上传到父亲节点继续匹配。答案为长度为  $k$  的子串个数减去最大的公共前缀长度和，即

$$(k * (n - k + 1) - sum)$$

## H 呼呼飞车

结论题，可以发现对于两个操作序列，如果一个操作序列的操作翻转串（即 'W' 与 'S' 互换，'A' 与 'D' 互换，然后左右翻转）的前缀与另一个操作序列的后缀相同（可以发现这个关系是符合交换律的），则不存在符合题意的操作序列。然后只需要对一个操作序列做翻转后对二者做 KMP 匹配即可。（不知道为什么开的人不多，可能是题面太长了，但结论多造几个样例推一推就能感觉到了）

## I 领航者试炼

我们先对原串做一遍字符串哈希，取出每一个长度为  $K$  的串的哈希值，建立主席树。查询的时候，我们先判断区间长度是否比  $K$  大，然后再判断给定的串的哈希值是否出现过，然后再在对应区间的主席树里面查找是否存在即可。

## J 气球（贰）

这里提供后缀数组的解法。要求出每一个前缀本质不同的后缀的个数，那么我们可以把原序列倒过来，然后实际上就是对于每一个后缀求与其它后缀不重复的前缀个数，也即是后缀长度减去 height 值；求出某一个后缀对答案的贡献之后，他不应该停留在元序列中

对后续答案的求解产生影响，所以应该把它删除；在实现方式上，可以使用链表，与平衡树的操作有些类似