

# Atcoder Rugular Contest 127

[比赛链接](#)

## D - Sum of Min of Xor

### 题意

给定序列  $A, B$  求  $\sum_{1 \leq i < j \leq n} \min(a_i \oplus a_j, b_i \oplus b_j)$

### 题解

从高到低考虑每个位，当  $a_i \oplus a_j$  和  $b_i \oplus b_j$  第一个位出现不同时已经可以判断  $a_i \oplus a_j$  和  $b_i \oplus b_j$  的大小关系。

注意到这也等价于  $a_i \oplus b_i$  和  $a_j \oplus b_j$  的第一个位出现不同。

假设当前考虑到第  $k$  位，此时将所有下标  $i$  划分成  $S, T$  其中  $S$  中每个  $i$  满足  $a_i \oplus b_i$  第  $k$  位为  $0$ ， $T$  中每个  $i$  满足  $a_i \oplus b_i$  第  $k$  位为  $1$ 。

递归处理  $\{i, j\} \subseteq S$  或  $\{i, j\} \subseteq T$  的  $(i, j)$  对的贡献。现考虑如何处理  $i \in S, j \in T$  的  $(i, j)$  对贡献。

根据  $a_i$  的第  $k$  位是否为  $0$  可以将  $S$  分为  $S_0, S_1$  同理将  $T$  分为  $T_0, T_1$

不难发现，对  $i \in S_0, j \in T_0$  有  $\min(a_i \oplus a_j, b_i \oplus b_j) = a_i \oplus a_j$  这转化为一个  $O(n \log V)$  的经典问题。

其余的  $(S_0, T_1), (S_1, T_0), (S_1, T_1)$  也有类似的解法。算上分治，总复杂度为  $O(n \log^2 V + 2^V)$

需要注意的是，递归最后一层为  $k = -1$  即  $a_i \oplus b_i = a_j \oplus b_j$  此时有  $\min()$

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal\\_string:jxm2001:contest:arc\\_127&rev=1632663821](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:contest:arc_127&rev=1632663821)

Last update: 2021/09/26 21:43