

NOIP 模拟赛 解题报告

BAJim_H

0 总体情况

0.1 难度分布

难度顺序大概是 $1 < 2 = 3$

第二题偏向套路，第三题偏向思维

0.2 得分分预测

300 : 1~3 人

250+ : 5~8 人

200+ : 10 人以上

150+ : 大众分

1 Dulu

1.1 题目大意

给出一个 n 个数的序列 a ，每次操作可以将一个数变成整个序列的值的异或

求最少需要多少次才能将 a 变成目标序列 b

无法完成输出 -1

1.2 得分预测

AC : 10 人以上

70+ : 15 人以上

50+ : 25 人以上

30+ : 绝大多数

1.3 题解

考虑操作的本质

只要按位稍微分析一下，就可以发现，题目相当于一开始手里抓着整个序列 a 的异或，一次操作可以将手上的数与序列中的某个数换过来

知道操作的本质就简单许多

如果能够完成，方便起见将 a 的异或和 b 的异或分别加到序列末，排序后两个序列显然完全相同，这样就把 -1 判掉了

只有 $a[i] \neq b[i]$ 的位置我们是需要调整的

那么将 a, b 离散化， $a[i]$ 与 $b[i]$ 连边

考虑统计答案

将一个联通块内换完所需要的次数明显就是联通块大小。

而联通块之间跳需要 1 的代价

此时由于我们一开始手上抓着的是 a 的异或和

如果它刚好在某一个联通块里面就不用考虑了, 否则就必须将它自己看做一个大小为 0 的联通块

联通块可以用并查集维护

1.4 题目来源

AtCoder Grand Contest 016D Xor Replace

https://agc016.contest.atcoder.jp/tasks/agc016_d

2 Travel

2.1 题目大意

给出一棵树, 每个节点有两个值 A_i, D_i , 表示这个节点能给它的 $0 \sim D_i$ 级祖先的 F 值贡献 A_i 。然后给每条边一个出现概率, 每次询问某个节点的联通块的 F 值的和的平方的期望

2.2 得分预测

AC : 6~8 人

70+ : 10 人左右

50+ : 15 人以上

30+ : 绝大多数

2.3 题解

2.3.1 20%

暴力求 F, 枚举每条边是否出现暴力算出答案

2.3.2 40%

考虑如何求 F

链剖? 并不需要

对于每个国家在它自己打上一个+的标记, 在它的 $D[i]+1$ 级祖先打上-的标记, 就可以直接子树求和了。查询某个位置的 $D[i]+1$ 级祖先可以在 DFS 的时候维护一个栈, 然后对于每个点直接访问栈中相应位置即可, 这也是线性的。因此这样就能线性的求出 F

考虑如何计算和的平方, 我们发现和的平方拆开来就是随便选两个的乘积 (有序) 的和。

顺着这个思路, 我们不妨以询问的国家作为根重新 DFS 一遍这棵树, 考虑 DP, 主要问题就是求两个块以一条出现概率为 p 的边合并时的答案。不妨令原本的块为 a , 并入的块为 b , 根据期望的线性可加性, 那么答案是 $p(a+b)^2 + (1-p)a^2 = a^2 + p(2ab + b^2)$

这样对于每个点 i , 维护以它为根的子树中期望和的平方 $G[i]$, 以及期望和 $H[i]$, 按照上面从儿子合并到父亲即可。一直合并到根就是当前询问点的答案, 这样一次询问的复杂度是 $O(N)$ 的

时间复杂度 $O(NQ)$, 空间复杂度 $O(N)$

2.3.3 50%

因为边只有 0 或 1，那么对于每个联通块直接统计答案即可。结合前面的算法可以得到 50 分

2.3.4 70%

一条链的话，可以提前预处理好前缀合并好的与后缀合并好的期望和的平方与期望和，在每个点直接合并就好了。

2.3.5 100%

其实并不需要对于每次询问都重新 DFS 转移，我们不妨仍然以 1 为根，那么对于一个询问的国家 i ，我们只需要知道 i 的子树部分的 G 和 H ，以及 i 以外的部分的期望和以及期望平方和，这可以由 i 的父亲的这两个值以及 i 的其他兄弟的 G 和 H 转移而来。如果合并以后再拆出来就不好做了，我们可以在 i 的父亲将 i 的兄弟全部记录下来，记录好前缀合并好的 G 和 H 以及后缀合并好的 G 和 H ，递归进 i 的时候合并前后缀与父亲过来的即可。

这样一来我们发现 Q 并没有什么用…，纯是迷惑作用

时间复杂度 $O(N+Q)$ ，空间复杂度 $O(N)$

由于这是 NOIP 模拟，我并没有将数据范围开到必须线性来做，有其他的带 \log 一类的做法欢迎分享。

毕竟是原创题，难免有一些不足还请各位见谅！

2.4 题目来源

原创

3 VanUSee

3.1 题目大意

给定两个串 S 和 T ， $|S| \geq |T|$ 。

两个人博弈，每一轮操作，两人先后可以删掉 S 的第一位或最后一位。

当操作以后的串的长度等于 $|T|$ 时，游戏停止。

如果停止时 S 剩下的串 $= T$ ，则后手获胜，否则先手获胜。

问两人都采取最优策略下谁赢

3.2 得分预测

AC：1~3 人

30：15 人以上…

3.3 题解

考虑简化表示状态

设左边已经删掉了 L 个字符，右边已经删掉了 R 个字符

那么用 $R-L$ 来表示当前状态，可以用 KMP 求出有哪些目标状态

一开始 $R-L$ 为 0，每一次操作可以让它 $+1$ 或者 -1 ，双方轮流操作，总共 $|S|-|T|$ 次操

作，双方都用最优策略看最后是否能到达目标状态

显然所有的目标状态奇偶性相同

当 $|S|-|T|$ 为奇数时，最后一次操作是先手做，它肯定往不是目标状态走，那么一个位置在最后一次操作前是目标状态当且仅当它+1 它-1 都是目标状态

现在就全部转化成 $|S|-|T|$ 为偶数的情况

假如 0 是目标状态，那么显然后手会赢，因为无论先手往哪里走后手都可以把他拉回来

如果 0 不是，并且 -2 和 2 不全是，那么先手一定会朝不是的那一边走，后手无论如何都没有办法将它拉回到是的那一边了

因此后手会赢当且仅当 0 是目标状态或者 -2 和 2 都是目标状态

否则都是先手赢

3.4 题目来源

牛客练习赛#17 F 玩游戏

<https://www.nowcoder.com/acm/contest/109#question>

4 Tips

第一题你只要看出了操作的本质，后面的就不太难了，因此“不要被事物的表面现象所迷惑”。

第二题，说不定你会做，当然也很有可能你不会做…

第三题是不是很有趣？

题目用到的知识点分别是并查集、树形 DP、KMP

是不是简单又自然，非常的 NOIP？

为了照顾写挂选手，特意没有采用捆绑数据，是不是很良心？

多谢大家 3.5h 的资瓷！