[Sqrt5 Round 2] 题解

T1

题外话

题目背景里说的那位同学是谁？没错，就是我(`WCK`)。

测试点 $1 \sim 3$

暴力枚举 a, b 的值，然后直接判断。

时间复杂度 $O(n^2)$，获得 `12pts`。

测试点 $4 \sim 9$

我也不知道这部分怎么做。

测试点 $10 \sim 18$

这一部分留给懂得一点数位 `DP`，但不会卡上/下界的同学。

正解

首先注意到 $\text{减法} = \text{异或} - \text{退位}$，所以 $a \oplus b$ 一定不会小于 $a - b$ 。我们只要拿总数减去 $a \oplus b = a - b$ 的个数，剩下的即为 $a \oplus b > a - b$ 的方案数。

我们设 a_i, b_i, L_i, R_i 分别表示 a, b, L, R 在 **二进制** 下 **从低到高数** 第 i 位（最低位为第 1 位）。

引理

$a \oplus b = a - b$ 当且仅当不存在 i ，使得 $a_i = 0$ 且 $b_i = 1$。

证明

因为 $\text{减法} = \text{异或} - \text{退位}$，因此只要不发生退位，$a \oplus b$ 就会等于 $a - b$。不难发现，$a - b$ 会发生退位当且仅当存在 $a_i = 0, b_i = 1$ 的情况。因此排除掉这种情况即可。

因此，如果 $a_i = 1$，b_i 可以为 0 或 1；如果 $a_i = 0$，b_i 必为 0。

不难发现，此时 a 在二进制下的每一位都大于等于 b，因此 $a \ge b$ 一定成立。那么我们只需要拿 $a \ge b$ 的情况数减去 $a \oplus b = a - b$ 的情况数，剩下的就是 $a \ge b$ 且 $a \oplus b > a - b$ 的方案数。

根据上述引理，我们可以数位 `DP` 求 $a \oplus b = a - b$ 的方案数。

设 $dp_{i,0/1,0/1}$ 表示考虑到了第 i 位（**从高位到低位考虑**），是否卡住了上界，是否卡住了下界的方案数。

`Q`: 什么叫“卡住了上界”?

`A`: 因为我们要求的 a, b 不仅要满足上述引理，还要在 $[L, R]$ 这个范围之间。考虑一个问题：假设 $R=2$，即二进制下的 10。如果 $a_2=1$，则 a_1 必须为 0 （否则 a 就会大于 R）。然而，如果 $a_2=0$，则 a_1 既可以等于 0，也可以等于 1。这是为什么呢？原来，$R_2=1$，而 a_2 也等于 1，这种情况我们称之为“卡住了上界”，此时 a_1 的取值不能超过 R_1 的值。但是当 $a_2=1$ 时，并没有“卡住上界”，此时 a 已经小于 R 了，所以 a_1 的取值无限制。设 R 在二进制下有 n 位，则第 i 位“卡住了上界”的含义是：a 的第 $i \sim n$ 位与 R 的第 $i \sim n$ 位相等，此时 a_{i-1} 的取值不能超过 R_i 的值；否则，a_{i-1} 的取值无限制。“卡住了上界”是用来限制 a_i 和 b_i 的取值范围的。“卡住下界”同理。

`Q` 为什么要从高位到低位 `DP`?

`A`: 因为这样可以很方便地判断是否“卡住上下界”。

`Q`: 题目中有两个变量 a 和 b，为什么不需要分别记录 a 和 b 是否“卡住上下界”，只需要记录一个呢?

`A`: 由引理可知，在 `DP` 的过程中，我们保证了 $a \geq b$，因此如果 b 卡住了上界，a 一定卡住上界；如果 a 卡住了下界，b 一定也卡住了下界。所以只需要记录一个就行了。

考虑转移。我们可以枚举 a 和 b 第 i 位填的是 0 还是 1，保证 $a_i \geq b_i$，然后直接从第 i 位转移到第 $i-1$ 位即可。

`Q`: 后两维（卡上下界）怎么转移?

`A`: 如果第 $i+1$ 位“卡住了上界”，并且 $a_i=R_i$，则第 i 位“卡住了上界”。卡住下界的转移方式同上。

初值：$dp_{62,1,1}=1$；目标：$dp_{0,0/1,0/1}$。

`Q`: 62 是什么?

`A`: 因为 L 和 R 都不超过 $\sqrt{5} \times 10^{18}$，小于 2^{61}，因此不会超过 61 位。从第 62 位开始，可以不用判断第 61 位是否卡住了上下界（原因见下文）。

`Q`: 为什么你 dp 的初值的后两维都是 1?

`A`: 由于 a, b, L, R 不超过 61 为，因此 $a_{62}=b_{62}=L_{62}=R_{62}=0$，此时一定卡住了上下界。

`Q`: 目标状态为什么是 $dp_{0,0/1,0/1}$，不是 $dp_{1,0/1,0/1}$?

`A`: 因为最低位也要做决策（枚举 a_1 和 b_1 的值），此时 dp_1 会转移到 dp_0，因此目标状态为 $dp_{0,0/1,0/1}$。

上述题解可能难以理解（尤其对于不会数位 DP 的人而言），如果实在不能理解，可以问我，也可以找我看代码，对着代码理解会容易得多。

问题解决，时间复杂度 $O(\log R)$ 。

T2

本题实际上就是 `CF1994F`。

注意到本题特别像欧拉回路，只不过本题的边分为 **必经边** 和 **可行边。**

测试点 $1 \sim 6$

爆搜/状压`DP`。

测试点 $7 \sim 9$

我也不会做。

测试点 $10 \sim 14$

所有边都要经过，直接对原图跑一边欧拉回路即可。

正解

注意到题目中的一个关键条件：所有的必经边都是连通的。

连通图存在欧拉回路的充要条件是：所有点的度数均为偶数。我们可以只保留必经边，求出每个点的度数。

然后考虑去掉必经边，只保留可行边。

显然，我们要选出一些可行边变成必经边，把奇度点变成偶度点。

那么，我们将奇度点两两配对，并将两个配对的点之间的路径上的边全部设为必经边，再跑欧拉回路，不就行了？

但是这样会有一个问题：一条边可能多条路径上，会重复经过，不符合题意！

怎么办呢？

比如，我们假设 A 和 B 配对，C 和 D 配对。假设 A 到 B 的路径为 $A \to x \to y \to B$，C 到 D 的路径为 $C \to x \to y \to D$，这时 $x \to y$ 这一段路径会被经过两次。

那我们能不能改成 $A \to x \to C$，$B \to y \to D$ 呢？当然可以。此时 $x \to y$ 这段路径上的边不会被设为必经边（因为没有经过）。

仔细观察上述流程，我们发现，对于每一对匹配点，我们相当于把它们之间的路径上的边 **翻转**（必经边变为可行边，可行边变为必经边）。

于是问题被转化为了：将奇度点两两配对，并将两个匹配点之间的路径翻转。

考虑如何快速实现这一操作。我们可以求出原图（只保留可行边）的 **生成森林**，这样问题被转化为了树上路径翻转。

你当然可以写树剖，但是可以利用一个技巧（树上差分？）避开树剖。

如果你要翻转 x 到 y 这段路径，你可以在点 x 和点 y 分别打一个懒标记，表示将 x 到根以及 y 到根的路径上的边都翻转。然后你神奇地发现，从根到 $\mathrm{LCA}(x,y)$ 的路径相当于被翻转了两次，所以相当于什么都没做，真正翻转的只有 $x \to y$ 的路径。

最后对生成森林进行一次 `dfs`，统计每个结点的子树中有多少个翻转标记，如果为奇数，则翻转它与它父亲之间的边。再跑欧拉回路即可，时间复杂度 $O(n)$。

T3

测试点 $1 \sim 4$

留给不会 `Manacher` 算法的同学。

测试点 $5 \sim 8$

用 `Manacher` 算法预处理每个点的回文半径，询问时暴力计算。

时间复杂度 $O(n^2)$。

测试点 $9 \sim 16$

首先用 `Manacher` 算法求出以每个点为回文中心的回文半径，记为 p_i 。

容易发现，询问区间为 $[l,r]$ 时，点 i 真实的回文半径为 $\min(i-l+1, r-i+1, p_i)$ 。

所以答案即为

$$\max_{i=l}^{r} \min(i-l+1, r-i+1, p_i)$$

首先考虑一个简单的分讨。设 $mid = \frac{l+r}{2}$，则 $l \le mid$ 时，$i-l+1$ 一定小于 $r-i+1$；当 $l > mid$ 时，$i-l+1$ 一定大于 $r-i+1$ 。

于是我们可以将答案拆为两部分：

一部分是

$$\max_{i=l}^{mid} \min(i-l+1, p_i)$$

一部分是

$$\max_{i=mid+1}^{r} \min(r-i+1, p_i)$$

两部分答案取 $\max$，就是答案。

由于两部分是类似的，因此我们只考虑第一部分，第二部分同理。

仔细观察一下这个式子：

$$\min(i-l+1, p_i)$$

不难发现，随着 l 的增大，$i-l+1$ 的值会逐渐变小。当 l 小于某一个值时，p_i 小于 $i-l+1$；当 l 大于某一个值时，$i-l+1$ 就会小于 p_i。我们把这个值记为 i 的 **临界值**，用 b_i 表示。显然 $b_i = i+1-p_i$。

那么这部分答案又可以拆成两部分，一部分是

$$\max_{i=l}^{mid} i-l+1$$

$$\text{s.t. } b_i \le l$$

一部分是

$$\max_{i=l}^{mid} p_i$$

$$\text{s.t. } b_i > l$$

对于第一部分，考虑对于每一个 i，在二维平面中建立一个点 (b_i, i)，权值为 $i+1$，则这部分的答案即为所有横坐标 $\le l$，纵坐标 $\in [l, mid]$ 的点的权值最大值减 l。这是一个二维数点问题，可以用离线扫描线等方法解决。

第二部分类似，只不过每个点的权值变成了 p_i，并且最后不需要减 l。

时间复杂度 $O(n \log n)$，但是需要离线。

正解

与上述做法类似，只是将离线扫描线替换为主席树，可在线解决，时间复杂度 $O(n \log n)$。

T4

本题实际上就是`ABC214G`。

测试点 $1 \sim 6$

暴力/状压`DP`。

测试点 $7 \sim 9$

不知道。

测试点 $10 \sim 12$

注意到 $p_i = q_i$ 时，此问题和 **错排问题** 等价，直接使用错排公式即可：

$f_i = (i-1) \times (f_{i-1} + f_{i-2})$

其中 f_i 表示的是长度为 i 的错排个数。初值为 $f_1 = 0, f_2 = 1$。

当然也可以用二项式反演等方式计算错排个数，时间复杂度 $O(n)$。

正解

由于 $r_i \neq p_i$ 且 $r_i \neq q_i$ 这样的不等关系不好处理，而处理相等关系则会容易得多。于是考虑二项式反演（容斥），设 f_i 表示钦定 i 位不合法（满足 $r_i = p_i$ 或 $r_i = q_i$）的方案数。

最后我们要求恰好有 0 位不合法的方案数，显然可以二项式反演，最终答案即为

$$\sum_{i=0}^{n} (-1)^i \binom{i}{0} f_i = \sum_{i=0}^{n} (-1)^i f_i$$

考虑如何计算 f。考虑建图，对于每一个 i，将 p_i 向 q_i 连边，得到一张图。不难发现每个点的出度均为 1，因此这张图一定是由若干个环组成的，我们把它们称为置换环。

考虑在置换环上的每一个点都填一个数，这些数对应着排列 r。容易发现，如果第 i 个位置不合法，要么 r_i 被填在了结点 p_i 上（$r_i = p_i$），要么 r_i 被填在了结点 p_i 连向的结点上（$r_i = q_i$）。并且，如果选择将 r_i 填在 p_i 指向的结点上，那么 p_i 指向的结点就不能选择将 r 填在自己的位置上（否则 p_i 指向的结点就填了两个不同的值，就冲突了）。

可以用`DP`求出每个置换环的方案数，再合并起来。设 $dp_{i,j,0/1}$ 表示考虑到置换环上的第 i 个位置，已经有 j 个位置不合法，且第 i 位是否选择填 p_i 指向的数。

转移是简单的，枚举当前位置不钦定/r_i 填在结点 i 上/r_i 填在结点 i 指向的结点上，直接转移即可。时间复杂度 $O(n^2)$。

注意，由于是环结构，如果最后一个位置选择了填在指向的结点上，那么第一个位置就不能填自己了。于是我们可以执行两次 `DP`，第一次 `DP` 时 r_1 不能填在结点 p_1 上，第二次 `DP` 时 r_1 必须填在结点 p_1 上，最后把答案加起来。

注意特判掉长度为 1 的置换环。

然后对每个置换环的答案做一次背包，把它们合并起来，得到数组 g，其中 g_i 表示钦定 i 个位置不合法，**且不考虑其它位置的填法**的方案数。对于一个长度为 len 的置换环，这一部分的复杂度为 $O(n \times len)$，因为所有置换环的长度之和为 n，所以总复杂度为 $O(n^2)$。

最后求 f_i，显然 $f_i = g_i \times (n-i)!$，因为有 i 个位置是钦定不合法的，剩下 $n-i$ 个位置可以随便填，因此要乘上 $(n-i)!$。

问题解决，时间复杂度 $O(n^2)$。