《匹配计数》解题报告

吴雨洋

1 题目大意

1.1 题目描述

给定正整数 n 以及正整数序列 $a_1, a_2, \cdots, a_n$，其中 a_i 表示第 i 个点的颜色。求同时满足如下条件的大小为 n 的简单无向图 G 的个数：

1. 边之间没有公共端点。即 G 是一匹配。

2. 对于任一条边它的两个端点的颜色相同。

3. 对两条不同的边 $e_1 = (u_1, v_1)(u_1 < v_1)$ 与 $e_2 = (u_2, v_2)(u_2 < v_2)$，若 $u_1 < u_2 < v_1 < v_2$ 或 $u_2 < u_1 < v_2 < v_1$ 则称 e_1 与 e_2 相交。满足 e_1 与 e_2 相交的无序对 (e_1, e_2) 有偶数个。

由于答案可能很大，对 998244353 取模。每个数据点有 T 组数据。

1.2 数据范围

对于数据点 $1 \sim 2$，$n \le 13$。

对于数据点 $3 \sim 4$，a_i 是全部相同的。

对于数据点 $5 \sim 10$，$a_i \le 10$。

对于数据点 $11 \sim 14$，$n \le 300$。

对于数据点 $15 \sim 20$，$n \le 2000$。

对 100% 的数据，$T \le 5, n \le 2000, a_i \le n$。

1.3 时空限制

时间限制：1 秒

空间限制：512 MB

2 解题过程

2.1 暴力

引理 2.1. 大小为 n 的完全无向图的匹配数可以 $O(n)$ 求解。

证明. 枚举连边数 k，然后我们考虑将每个点都分别放入 A, B, C 三个集合中的一个，使得 $|A| = |B| = k, |C| = n - 2k$，然后我们只需对 A, B 间的完美匹配计数。而对于一个真正的方案，它会被这个方法计算 2^k 次。所以方案数是

$$\sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{n!}{k!(x-2k)!2^k}$$

□

当 $n = 13$ 时，这个值是 568504，因此直接搜索即可获得 10 分。

2.2 初步分析

我们先忽略掉限制 3，然后来考虑这个问题。对于每种颜色，我们求出只考虑这个颜色的点的匹配数，然后将所有求出的方案数相乘即可。由引理 2.1 这部分可以 $O(n)$ 解决。

接下来考虑如何处理限制 3。对于所有满足限制 1 和限制 2 的图，我们设 $(-1)^t$ 为这个图的权值，这里 t 是满足 e_1, e_2 相交的无序对 (e_1, e_2) 的个数，我们也把它叫做这个图的交点数。

容易发现所有满足限制 1 和限制 2 的图的权值和与忽略限制 3 的方案数求和再除以 2 就是答案了。

引理 2.2. 对正整数 n，所有大小为 $2n$ 的完美匹配的权值和为 1。

证明. 考虑归纳，当 $n = 1$ 时这是 $(-1)^0 = 1$。

假设 $n = m$ 时成立，则当 $n = m + 1$ 时，考虑第 1 个点的邻居在剩下的点中的排名。若排名为 $k(k = 0, 1, \cdots, 2m)$，则它与其它边产生的交点个数的奇偶性与 k 的奇偶性相同。

有 $m + 1$ 个排名为偶，m 个排名为奇，因此此时的值是 $((m + 1) - m) \times 1 = 1$。 □

由引理 2.2 可以解决第 2 档部分分，结合暴力可获得 20 分。

引理 2.3. 边 (u, v) 上的交点数的奇偶与 $\{u + 1, u + 2, \cdots, v - 1\}$ 间有边连接的点数的奇偶是相同的。

证明. 令 $S = \{u + 1, u + 2, \cdots, v - 1\}$。

对一条端点都在 S 中或端点都不在 S 中的边，它不会对这两个数的奇偶造成影响。

对一条恰一个端点在 S 中的边，它会同时改变这两个数的奇偶性。 □

考虑只指定序列中有边连接的点。每种颜色被指定的点恰偶数个。

考虑对于所有匹配，由引理 2.3 不同颜色之间产生的交点与将被指定的点按序取出后的逆序对数奇偶性相同，是不由匹配的样子而改变的。

所以可以通过乘法分配律来只考虑颜色相同的点的贡献，最后乘在一起即可。而对于颜色相同的点，由引理 2.2 所有方案的贡献的和是 1。

所以所有方案的权值之和就是 -1 的 “把指定的点按序取出后的逆序对数” 次方。

可以设计一个状压 DP 的算法，记录选出的点中每种颜色出现次数的奇偶。设 a_i 的最大值为 A，则时间复杂度为 $O(nA2^A)$，结合之前的算法可以获得 50 分。

2.3 问题的一般化与解决方案

考虑这样一个问题：给简单无向图 G，若 i,j 间有边则 $G_{i,j}$ 为 1 否则为 0，求

$$\sum_{x_1,x_2,\cdots,x_n\in\{0,1\}}(-1)^{\sum_{1\leq i<j\leq n}G_{i,j}x_ix_j}$$

其中 x_i 表示第 i 个点选不选。

可以换一个说法，对 G 的每一个子集求 -1 的导出子图边数次方的和。

我们考虑选中一个点 u。若它的邻居中有奇数个被选的点，则选它与不选它对答案的贡献是互为相反数的。因此，我们可以将问题转化：删除节点 u，并强制它的邻居中一定要选偶数个点，然后将这个答案乘 2。

设 V 是 u 的邻居集合，v 是 u 的某个邻居。设 v 的邻居集合为 S。注意到一个方案对答案的贡献可以拆成删除 v 后的贡献乘上与 v 有关的贡献，而又由于 $\sum_{q\in V}x_q\equiv 0 \pmod 2$，所以这是

$$(-1)^{\sum_{p\in S}x_vx_p}=(-1)^{\sum_{p\in S}\sum_{q\in V-\{v\}}x_qx_p}$$

所以问题可以转化成对每个 $p\in S,q\in V-\{v\}$ 在 p,q 中加一条边，然后将 v 删除。

注意到在这个操作做完后，某些点会出现自环。所以真正的算法比上面的讨论要略复杂一些。这里只给出最后的结果：

1. 选中一个点 u。

1.1. 若 u 没有邻居，则若 u 有自环答案就是 0，否则删除 u，答案乘 2。

1.2. 若 u 有邻居，答案乘 2，然后删除 u。接下来选中 u 的一个邻居 v。设 u 的邻居除去 v 后的集合为 V，设 v 的邻居除去 u 后的集合为 S。

1.2.1. 若 v 有自环，

1.2.1.1. 对 V 中每个点 t 给 t 连一个自环。

1.2.1.2. 若 u 有自环就将答案乘 -1。

1.2.2. 对 S 中每个点 t，

1.2.2.1. 对每个 $p\in V$，在 t,p 之间连一条边，要注意此时连的边可能是自环。

1.2.2.2. 若 u 有自环则给 t 连一个自环。

2. 重复执行上述过程，直到所有节点都被删除。

普通的实现可以做到 $O(n^3)$。

而注意到复杂度瓶颈在于对 S 中每个点 t 与 V 中每个点 p 在 t,p 间连边。但如果对邻接矩阵的每一行开一个 bitset，则这实际上就是一个 bitset 异或另一个 bitset。

所以时间复杂度可以做到 $O(\frac{n^3}{\omega})$。

对于原题，我们令 $G_{i,j}=[a_i>a_j]$，然后又限制了每种颜色的点要选偶数个。但由上述算法，我们发现这个限制是容易做到的。所以可以做到 $O(\frac{n^3}{\omega})$ 的复杂度，可以获得 100 分。