

NOI 2025 Day1 C. 数字树

1kri

2025 年 7 月 14 日

问题 (数字树)

- 给定一棵完全二叉树，每次给两个叶子写上新的数字，然后求有多少个 *DFS* 序是优美的。
- 一个 *DFS* 序是优美的，当且仅当它遍历到的所有叶子数字构成的序列，能通过若干次消除相邻相同数字的方式删空。

数据范围： $n \leq 2 \times 10^5$ 。

欢迎吐槽。

- 先考虑所有叶子填完之后怎么求出答案。

- 先考虑所有叶子填完之后怎么求出答案。
- 由于保证有解，不妨假设一开始的树的先序遍历就是优美的。

- 先考虑所有叶子填完之后怎么求出答案。
- 由于保证有解，不妨假设一开始的树的先序遍历就是优美的。
- 将 DFS 序一一对应到：翻转一些子树后的先序遍历。
 - 翻转指对子树内每个点交换左右儿子。

- 先考虑所有叶子填完之后怎么求出答案。
- 由于保证有解，不妨假设一开始的树的先序遍历就是优美的。
- 将 DFS 序一一对应到：翻转一些子树后的先序遍历。
 - 翻转指对子树内每个点交换左右儿子。
- 这里先直接给出结论：对于每个点的 DFS 序区间，把它们按照区间中出现恰一次的数的集合分成若干等价类，答案就是 2^{n-c} ，其中 c 为出现恰一次的数的集合大小 ≥ 2 的等价类个数。

- 先考虑特殊性质 A，即每次操作选择的两个叶子位于 1 的不同子树内。

- 先考虑特殊性质 A，即每次操作选择的两个叶子位于 1 的不同子树内。
- 因为所有的点对都是经过根的，所以翻转了左边的一个子树，也即翻转了一段 DFS 序区间，就需要对应地翻转右边的完全一样的区间。

- 先考虑特殊性质 A，即每次操作选择的两个叶子位于 1 的不同子树内。
- 因为所有的点对都是经过根的，所以翻转了左边的一个子树，也即翻转了一段 DFS 序区间，就需要对应地翻转右边的完全一样的区间。
- 由于所有区间形成了若干等价类，因此上述限制等价于：每个等价类必须恰好选择偶数个区间翻转。

- 现在推广到一般情况：考虑根的两个子树，它们中有一些跨越根的和完全在内部的数字对。

- 现在推广到一般情况：考虑根的两个子树，它们中有一些跨越根的和完全在内部的数字对。
- 对于任何一个点的子树，分析时只保留恰出现一次的数字，而忽略其它的。如果只有跨过根的数字，那么交由对于特殊性质 A 的分析。

- 现在推广到一般情况：考虑根的两个子树，它们中有一些跨越根的和完全在内部的数字对。
- 对于任何一个点的子树，分析时只保留恰出现一次的数字，而忽略其它的。如果只有跨过根的数字，那么交由对于特殊性质 A 的分析。
- 一个子树如果既包含跨越根的数字，又打乱了一些完全在内部的数字对，就一定要翻转偶数次。这个结论可以反证：找到最上面的翻转了奇数次的子树，没有其它子树能“救”它，就一定不合法。

- 现在推广到一般情况：考虑根的两个子树，它们中有一些跨越根的和完全在内部的数字对。
- 对于任何一个点的子树，分析时只保留恰出现一次的数字，而忽略其它的。如果只有跨过根的数字，那么交由对于特殊性质 A 的分析。
- 一个子树如果既包含跨越根的数字，又打乱了一些完全在内部的数字对，就一定要翻转偶数次。这个结论可以反证：找到最上面的翻转了奇数次的子树，没有其它子树能“救”它，就一定不合法。
- 我们保留了只有完全在内部的数字对的情况，因此可以递归进根的两个子树各自分析。综合以上分析，我们发现恰得到之前陈述的结论。

- 现在考虑具体实现。

- 现在考虑具体实现。
- 对于每个点记录一个长度为 n 的 01 串，表示数 i 在这个点的子树内是否出现恰一次。

- 现在考虑具体实现。
- 对于每个点记录一个长度为 n 的 01 串，表示数 i 在这个点的子树内是否出现恰一次。
- 对于操作 i 次后的等价类个数，其实就是这 $2n - 1$ 个长度为 n 的 01 串，有多少个不同的 i -前缀。

- 现在考虑具体实现。
- 对于每个点记录一个长度为 n 的 01 串，表示数 i 在这个点的子树内是否出现恰一次。
- 对于操作 i 次后的等价类个数，其实就是这 $2n - 1$ 个长度为 n 的 01 串，有多少个不同的 i -前缀。
- 那么直接用线段树合并维护 01 串的哈希值，然后按字典序排序。

- 现在考虑具体实现。
- 对于每个点记录一个长度为 n 的 01 串，表示数 i 在这个点的子树内是否出现恰一次。
- 对于操作 i 次后的等价类个数，其实就是这 $2^n - 1$ 个长度为 n 的 01 串，有多少个不同的 i -前缀。
- 那么直接用线段树合并维护 01 串的哈希值，然后按字典序排序。
- 对于第 i 大的 01 串，设其第二个 1 的位置为 p ，与第 $i - 1$ 大的 01 串的最长公共前缀为 l ，则该串会对所有 $\max(p, l)$ 后的询问产生贡献。

- 现在考虑具体实现。
- 对于每个点记录一个长度为 n 的 01 串，表示数 i 在这个点的子树内是否出现恰一次。
- 对于操作 i 次后的等价类个数，其实就是这 $2n - 1$ 个长度为 n 的 01 串，有多少个不同的 i -前缀。
- 那么直接用线段树合并维护 01 串的哈希值，然后按字典序排序。
- 对于第 i 大的 01 串，设其第二个 1 的位置为 p ，与第 $i - 1$ 大的 01 串的最长公共前缀为 l ，则该串会对所有 $\max(p, l)$ 后的询问产生贡献。
- 时间复杂度 $O(n \log^2 n)$ ，期望得分：100。

- 考虑优化最后的字典序排序。

- 考虑优化最后的字典序排序。
- 考虑将线段树合并中所有的 $O(n \log n)$ 个结点排序。

- 考虑优化最后的字典序排序。
- 考虑将线段树合并中所有的 $O(n \log n)$ 个结点排序。
- 以每个结点的左右儿子的字典序为第一、二关键字，进行基数排序即可。

- 考虑优化最后的字典序排序。
- 考虑将线段树合并中所有的 $O(n \log n)$ 个结点排序。
- 以每个结点的左右儿子的字典序为第一、二关键字，进行基数排序即可。
- 为了方便实现，我们把线段树长度补成二的次幂，从线段树的叶子（区间长度为一）开始做，每次给一层排序。

- 考虑优化最后的字典序排序。
- 考虑将线段树合并中所有的 $O(n \log n)$ 个结点排序。
- 以每个结点的左右儿子的字典序为第一、二关键字，进行基数排序即可。
- 为了方便实现，我们把线段树长度补成二的次幂，从线段树的叶子（区间长度为一）开始做，每次给一层排序。
- 时间复杂度 $O(n \log n)$ ，期望得分：100。