

Право на одиночество

Сначала проверим, не являются ли все числа уже различными. Если это так — ответ 0.

Теперь пусть $k = 0$. Тогда при замене поддерева на другое будет 2 идентичных поддерева, поэтому сделать все числа различными не получится.

Иначе сделаем так: сначала заменим поддерево левого сына корня на любой лист справа. Потом заменим поддерево правого сына корня на значение в левом сыне, проксоренном с k . Так как $k \neq 0$, значения будут различными. Таким образом, всегда есть конструкция за 2 операции, а следовательно ответ на задачу не больше 2.

Осталось проверить, возможно ли решить задачу за 1 операцию. Зафиксируем поддерево, которое мы хотим поменять. Его оптимально заменить на лист вне поддерева. Теперь задача звучит так: существует ли такое поддерево, что если рассмотреть все значения вне него, то они все различные и найдётся такое число x среди них, что $x \oplus k$ нет среди значений. Будем поддерживать множества значений и значений, проксоренных с k . Если в множествах все значения различны, а также они не равны, то такой x найдётся. Оба условия можно поддерживать при добавлении и удалении значения за $O(1)$, если предварительно сжать значения a_v и $a_v \oplus k$.

Теперь разберёмся, как перебирать поддеревья. Будем делать dfs и добавлять сначала все значения левого поддерева и спускаться в правое, потом удалять все значения левого поддерева, добавлять все значения правого поддерева и спускаться в левое дерево, затем удалять все значения правого поддерева. Поскольку у нас полное двоичное дерево на 2^n вершинах, такой dfs будет делать $O(n2^n)$ добавлений и удалений, которые работают за $O(1)$, следовательно время работы составляет $O(n2^n)$.