Разбор задачи «Оптимальное перестроение»

Решение 1. Наивное. Чтобы решить задачу при маленьких n, достаточно было перебрать значение x, после этого построить новый ряд из рыб по правилам, указанным в условии, после чего для каждого из получившихся рядов перебрать все пары рыб в нем и проверить, не образуют ли они пару, где более сильная рыба стоит раньше менее сильной.

Асимптотика времени работы данного решения — $\mathcal{O}(n^3)$, этого достаточно для получения 40 баллов.

Ключевое замечание. Пусть в исходной перестановке была пара позиций i, j, для которой выполнялось $i < j$ и $a_i > a_j$. Пусть также Аквамен отдал команду перестроения и назвал число x. Заметим, что эта пара рыб будет стоять относительно друг друга **не** в таком же порядке после перестроения тогда и только тогда, когда выполняется неравенство $a_i \geqslant x \geqslant a_j$ (так как в этом случае первая рыба с номером j встанет до рыбы x, а рыба с номером i — после).

Таким образом, чтобы посчитать ответ для каждого x, надо посчитать количество пар номеров i, j, для которых верно, что $i < j$ и либо $x > a_i > a_j$, либо $a_i > a_j > x$. Пусть a^{-1} — обратная перестановка к перестановке a, то есть такая последовательность из n различных чисел от 1 до n, для которой верно $a^{-1}_{a_i} = i$. Теперь введем новые обозначения: $i' = a_i$, $j' = a_j$. Тогда $i = a^{-1}_{i'}$, $j = a^{-1}_{j'}$. Таким образом, нам надо посчитать количество пар i', j', где $a^{-1}_{i'} < a^{-1}_{j'}$, а также либо $i' > j' > x$, либо $x > i' > j'$. Заметим, что это количество инверсий на префиксе до числа x и на суффиксе после числа x в перестановке a^{-1}. Инверсией мы называем пару индексов x, y, где $x < y$ и $a^{-1}_x > a^{-1}_y$.

Теперь в перестановке a^{-1} посчитаем для каждого элемента количество инверсий, которые он образует с элементами после него и до него. Таким образом, для каждого x ответ — это сумма этих чисел на префиксе до числа x и после числа x.

Существует множество способов считать для каждого элемента количество инверсий, которые он образует. Если делать это наивно, с помощью цикла, получается решение с асимптотикой времени работы $\mathcal{O}(n^2)$, которое получает 60 баллов. Если применить одну из структур данных с запросами на отрезке, таких как корневая декомпозиция, дерево отрезков или дерево Фенвика, можно было получить решение с асимптотикой времени работы до $\mathcal{O}(n \log n)$, и получить от 80 до 100 баллов.