Необычная сортировка

Автор задачи и разработчик: Мария Жогова

Для решения **первых двух подзадач** достаточно было написать полный перебор всех возможных способов выбрать наборы позиций изменяемых символов. Отличие в том, что при $n \leqslant 3$ можно было обойтись условными операторами, а при $n \leqslant 10$ проще было написать перебор битовых масок от 0 до $2^n - 1$, кодирующих выбор индексов (i-й индекс выбран $\iff$ i-й бит в маске равен 1). А уже при фиксированном наборе индексов достаточно за время $\mathcal{O}(n)$ построить итоговый массив и проверить его на неубывание. Получили решение, работающее за $\mathcal{O}(2^n \cdot n)$ времени.

Массив в **третьей подзадаче** является перестановкой чисел от 1 до n. Чтобы не перебирать целиком все способы выбрать позиции изменяемых элементов, попробуем найти элементы, для которых знак определен однозначно. Логично посмотреть на максимальный элемент $a_i = \max(a)$, так как со знаком плюс он больше любого другого числа с любым знаком, и наоборот, $-a_i$ меньше любого другого возможного значения в массиве. Таким образом, если $\max(a) = n$ стоит на первом месте, его надо взять со знаком минус, если на последнем — с плюсом, иначе массив невозможно отсортировать.

Удалим максимальный элемент с конца, получим массив длины $n-1$, являющийся перестановкой чисел от 1 до $n-1$, для которого можно повторить аналогичное рассуждение. Чтобы удаление работало быстро, можно поддерживать левую и правую границу актуальной части массива и двигать из за $\mathcal{O}(1)$. Единственное отличие возникнет, когда останется массив только из числа 1 — оно и первое, и последнее, и его можно взять с любым знаком. Таким образом, ответ равен либо 2, либо 0, если на каком-то шаге очередной максимум оказался не с краю массива.

Решение **четвертой подзадачи** полностью повторяет решение третьей за тем исключением, что надо быстро находить максимум на оставшейся части массива. Одним из способов сделать это является сжатие значений в отрезок $[1; n]$ с помощью `map`, тем самым полностью сведя эту подзадачу к третьей.

Разберем решение **пятой подзадачи**. Заметим, что от умножения на -1 элементы, равные нулю, не меняются, а в неубывающем массиве перед нулем идут неположительные элементы, а после — неотрицательные. Поэтому если в массиве между двумя нулевыми элементами a_i и a_j есть ненулевой a_k, массив невозможно отсортировать описанным образом, так как будет выполнено либо $a_k > a_j = 0$, либо $0 = a_i > a_k$. Таким образом, если в массиве есть $a_i = 0$, то все нулевые элементы должны образовывать непрерывный отрезок индексов, все ненулевые элементы слева от него обязаны стать отрицательными и все ненулевые элементы справа обязаны стать положительными.

Решение подгруппы, соответственно, заключается в проверке того, что нулевые элементы образуют один отрезок, что можно сделать за время $\mathcal{O}(n)$, и проверке, что оставшиеся элементы с нужными знаками неубывают. Остается посчитать ответ. Но для каждого ненулевого элемента есть ровно один способ задать ему знак, а для каждого нулевого можно независимо от других выбрать, умножать его на -1 или нет — массив от этого не изменится. Ответ будет равен либо 0, если необходимые условия не выполнены, либо $2^{\texttt{count(0)}}$.

В **шестой подзадаче** гарантировалось, что каждое число в массиве имеет ровно два вхождения в него. Обозначим позиции первого и второго вхождения числа x за l_x и r_x, соответственно. Тогда если $l_x \neq r_x - 1$, между ними есть число, не равное x. Но тогда нельзя взять вхождения x с одинаковым знаком, так как по неубыванию все элементы между ними были бы обязаны быть равны x. Поэтому x на позиции l_x обязательно надо взять со знаком минус, на на r_x — с плюсом.

Расставим во всех таких парах нужные знаки, проверим, что нет положительных чисел, идущих перед отрицательными. Рассмотрим оставшиеся пары, для которых l_x и r_x стоят рядом в массиве. Несложно заметить, что если среди них минимальный модуль имеет x_0, то все пары слева обязаны иметь знак минус, а все справа — знак плюс. Остается только проверить, что все расставленные знаки дают невозрастание. В самой же паре (l_{x_0}, r_{x_0}) можно выбрать знаки прозвольным способом из $(-,-)$, $(-,+)$ и $(+,+)$, поэтому ответ — либо 3, либо 0.

Возвращаясь к третьей и четвертой подзадачам, заметим, что альтернативно максимальным элементам можно было смотреть на минимальный, и применить к нему рассуждение, аналогичное рассуждению про 0 в пятой подзадаче или про минимальную пару в шестой.

Из этого можно получить **полное решение**. Найдем в массиве число с минимальным абсолютным значением и проверим, что элементы с таким же абсолютным значением занимают в массиве непрерывный отрезок. Все элементы слева должны быть отрицательными, все справа — положительными. А сам отрезок k минимумов можно упорядочить 2^k, если это нули, и за $k+1$, если не нули. Для этого надо выбрать префикс, который будет отрицательным, и все оставшиеся элементы будут положительными. Время работы решения — $\mathcal{O}(n)$.