Футбольная команда

Чесаков Д. Г.

5 октября 2019 г.

1 Условие задачи

Вы хотите набрать футбольную команду. У каждого игрока своя эффективность, она описывается одним целым числом. Чем больше число, тем больше эффективность футболиста. Обязательным условием для любой команды является сплоченность. Если один из игроков играет сильно лучше всех остальных, его будут недолюбливать, и команда распадется. Поэтому эффективность любого игрока команды не должна превышать сумму эффективностей любых двух других игроков. Ваша задача — набрать команду, которая будет удовлетворять условию сплоченности, и при этом иметь наибольшую суммарную эффективность.

1.1 Формат ввода

В первой строке входа задано число n ($1 \leq n \leq 100000$). Во второй строке — эффективности каждого из n игроков — положительные целые числа, не превосходящие $2^{31} - 1$.

1.2 Формат вывода

Выведите две строки. В первую запишите наибольшую возможную сумму эффективностей игроков в команде. Во вторую строку выведите номера всех игроков, которых нужно взять в команду, в порядке возрастания.

2 Алгоритм решения

Сначала мы берем данный нам массив из n чисел, копируем его, а затем сортируем по возрастанию. Пусть отсортированный массив будет A, а изначальный B. Теперь ставим 2 указателя в отсортированном массиве l и r. l на его первый элемент и r на второй. Мы задаем 4 переменные – L, R, Max, Sum. $L = A[0], R = A[1], Sum = Max = A[0] + A[1]$. Дальше мы двигаем r вправо до тех пор пока условие, что элемент, на который указывает r меньше или равна суммы элементов A$[l]$ + A$[l+1]$, не перестанет выполняться. Когда условие перестает выполняться мы двигаем указатель l вправо до тех пор, пока условие снова не начнет выполняться, а дальше снова двигаем указатель r и так далее. Также на каждом шаге, мы либо добавляем к S элемент A$[r]$, либо вычитаем из S элемент A$[l]$. Если же условие A$[l]$ + A$[l$+1$]$ $\leq$ A$[r]$ выполнено, то сравниваем Sum и Max, и когда Sum > Max, обновляем Max до значения Sum, и L и R до значений A$[l]$ и A$[r]$. Так делаем до тех пор, пока указатель r не дойдет до конца массива и l не станет таким, что условие станет выполняться.

Теперь проходимся по неотсортированной копии B с первого элемента и до последнего, а если элемент B$[i]$ $\geq$ L и B$[i]$ $\leq$ R, то выводим его индекс i.

3 Доказательство правильности

Единственное условие, которое мы имеем на нашу команду (кроме того, чтобы эффективность была максимальной), это то, что ни один из игроков не имеет эффективность больше, чем суммарная эффективность любых двух других игроков. Это условие эквивалентно условию, чтобы максимальная эффективность игрока в команде не была больше суммы двух минимальных эффективностей. Отсюда следует, что игроки у которых эффективности попадают на промежуток между минимальной и максимальной эффективностью игроков команды, точно в нее попадут, если в команде уже хотя бы 3 игрока. Это достаточно понятно, пусть у нас уже сформирована команда, в которой игрок с максимальной эффективностью имеет эффективность x_{max}, а игроки с двумя минимальными x_{min1} и x_{min2}, тогда для игрока, имеющего эффективность x, меньшую или равную, чем x_{max} и большую или равную, чем x_{min2}, будет выполняться условие $x \leq x_{min1} + x_{min2}$, поскольку по условию уже сформированной команды $x_{max} \leq x_{min1} + x_{min2}$, а $x \leq x_{max}$. Также, если у игрока со второй наименьшей эффективностью индекс i, то у игрока с наименьшей эффективностью индекс будет $i - 1$, поскольку мы максимизируем суммарную эффективность, а наш массив отсортирован по возрастанию. Ведь если это не так и у него индекс $k < i-1$, то очевидно, что мы можем заменить минимальный элемент x_k на x_{i-1} и наша суммарная эффективность как минимум не уменьшится, поскольку $x_k \leq x_{i-1}$. Значит, если мы отсортировали наш массив по эффективностям, то искомая команда будет идти подряд в массиве без прерываний. Теперь наша задача свелась к тому, чтобы на отсортированном массиве найти подпоследовательность с максимальной суммой, идущих подряд элементов, где максимальный элемент не больше, чем сумма двух минимальных. Тогда мы пользуемся методом двух указателей. Двигая правый указатель мы увеличиваем сумму, двигая левый – уменьшаем. Нам нет смысла двигать левый указатель, пока условие выполняется, так как это только уменьшит сумму и максимум мы получить не сможем. Так как последовательность у нас неубывающая, то для левых указателей нет смысла перебирать все позиции правого указателя, это только уменьшит сумму, можно начинать с позиции, на которой мы остановились для предыдущего левого указателя. Таким образом мы проходим по всем кандидатам на подпоследовательность с максимальной суммой в массиве и среди них выбираем максимум.

Как уже было сказано выше, у нас попадают игроки всех эффективностей между максимальной и минимальной, так что пройдясь по неотсортированному массиву, когда мы смотрим на каждый элемент, нам не надо проверять есть ли он в подпоследовательности напрямую, а мы можем просто сравнить его с границами. А значит, пройдясь по всему массиву, мы выпишем индексы всех элементов, которые отвечают двум этим неравенствам, а значит входят в комаду. Что и требовалось доказать.

4 Временная сложность – ассимптотика

Наиболее затратной частью здесь является сортировка, которая отнимет у нас $O(n \times log(n))$ времени, также мы проходимся по стеку методом двух указателей и затем еще раз, когда выписываем индексы, что занимает у нас $O(n)$, а значит в результате мы тратим $O(n \times log(n))$.

5 Затраты памяти – ассимптотика

Применяя сортировку слиянием нам требуется $O(n)$ памяти, столько же нам требуется на оба массива, так что мы тратим $O(n)$ памяти по ассимптотике.