Теория к задаче Футбольная команда

1 Алгоритм решения

Если игроков два или меньше, возвращаем всех игроков.

Иначе отсортируем массив игроков по возрастанию эффективности и назовем его Players.

Заведем два указателя from и to, инициализируем from началом Players, to - позицией третьего игрока. Заводим переменную sum и кладем в нее сумму эффективностей игроков в интервале [from, to). Заводим копии этих переменных для сохранения оптимальной команды.

Указатель from проходит в цикле весь массив. В конце каждого цикла сохраняется следующий инвариант: [from, to) обозначает границы подмассива Players, из которого можно собрать самую большую команду, начинающуюся с from, а sum хранит сумму эффективностей игроков в этом интервале.

Это достигается за счет того, что мы во вложенном цикле увеличиваем to, пока эффективность игрока на позиции to не больше суммы эффективностей игроков на позициях from и from+1, и при увеличении to обновляем sum.

Если sum превысила текущий максимум, запоминаем ее вместе с соответствующими указателями. Цикл заканчивается, инкрементируем from и обновляем sum.

Находим номера игроков, которые составляют оптимальную команду, и сортируем их.

Возвращаем максимальную эффективность команды и саму команду.

2 Доказательство правильности алгоритма

Если игроков два или меньше, они образуют тривиальную оптимальную команду. Иначе в оптимальной команде всегда не меньше 2 игроков.

Если эффективность to не больше суммы эффективностей from и from+1, то из интервала [from, to] можно составить команду. Действительно, поскольку игроки упорядочены по неубыванию эффективностей, для любых различных игроков a, b, c из интервала [from, to] верно следующее:

$$a.eff \leqslant to.eff \leqslant from.eff + (from + 1).eff \leqslant b.eff + c.eff$$

Заметим, что команду с максимальной эффективностью можно найти среди подмассивов внутри Players. Действительно, если команда содержит игроков Players[i], Players[k], но не содержит Players[i+1], i+1 < k, то можно взять Players[i+1] вместо самого слабого. Если игроков было двое, это очевидно. Иначе

$$\exists j : Players[i + 1].eff \leqslant Players[k].eff \leqslant Players[i].eff + Players[j].eff$$

$$\forall t : Players[t].eff \leqslant Players[k].eff \leqslant$$

$$\leqslant Players[i].eff + Players[j].eff \leqslant Players[i+1].eff + Players[j].eff$$

Повторяя это преобразование, мы не ослабляем команду и в итоге приведем ее к подмассиву. Значит, для любой команды определенной эффективности существует команда-подмассив не меньшей эффективности. Пусть этот подмассив начинается в from. Поскольку в соответствующем цикле был найден подмассив-команда максимальной длины, начинающийся в from, то найденная оптимальная команда имеет эффективность, не меньшую любой другой команды.

3 Временная сложность — асимптотика

Сортировка массива Players - сложность $O(n \log n)$

В вложенном цикле каждый раз инкрементируется to, значит всего проходов по вложенному циклу не более n. В основном цикле инкрементируется $from$, аналогично проходов не более n, все операции элементарны - $O(n)$

Создание массива ответов - $O(n)$

Сортировка массива ответов - $O(n \log n)$

Общая сложность - $O(n \log n)$

4 Затраты памяти — асимптотика

Если создавать копию массива для вывода или создавать структуру с полем для позиции и эффективности и сортировать in-place и перезаписывать входные данные, все равно в итоге необходима дополнительная память O(n).