

Теория к задаче Футбольная команда

1 Алгоритм решения

Если игрок один, возвращаем его.

Иначе отсортируем массив игроков по возрастанию эффективности и назовем его `Players`. Создадим массив кумулятивных сумм массива `Players` и назовем его `Partial`.

Заведем два указателя `from` и `to`, сохраняющие следующий инвариант: они обозначают границы подмассива `Players`, из которого можно собрать самую большую команду, начинающуюся с `from`.

Указатель `from` проходит весь массив. Для каждого значения `from` находим такое значение `to`, чтобы сохранялся инвариант. По пути вычисляем эффективность команды как `Partial[to] - Partial[from]`, и запоминаем, если она превышает текущую, вместе с соответствующими указателями.

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

Возвращаем максимальную эффективность команды и саму команду.

2 Доказательство правильности алгоритма

Заметим, что оптимальная команда должна быть подмассивом внутри `Players`. Действительно, если команда содержит игроков `Players[i]`, `Players[k]`, но не содержит `Players[i+1]`, $i+1 < k$, то можно либо добавить игрока `Players[i+1]` в команду, если в команде есть двое слабее него, либо взять его вместо самого слабого, тем самым строго усилив команду.

Если игрок один, то возвращаем его. Иначе в оптимальной команде всегда как минимум два игрока.

Поскольку мы пробегаемся по всем оптимальным массивам размера как минимум 2, начинающимся во всех возможных местах, то мы найдем и глобально оптимальную команду.

3 Временная сложность — асимптотика

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

Создание массива `Partial` - $O(n)$

Прохождение указателей `to` и `from` по массиву - $O(n)$

Создание массива ответов - $O(n)$

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

Общая сложность - $O(n \log n)$

4 Затраты памяти — асимптотика

Для массива `Partial` - $O(n)$

Массив ответов можно перезаписать на месте Partial, все равно в итоге необходима дополнительная память $O(n)$.