# Домашнее задание №2

Д.А. Першин

29 октября 2014 г.

## 1 Словесное описание алгоритма

При решении данной задачи будем использовать алгоритм быстрой сортировки (временная слжность - $O(n \log n)$, память - $O(\log n)$) и алгоритм бинарного поиска (сложность - $O(\log n)$). Входной массив будет состоять из пар чисел $e_i$ и $i$ (необходим для восстановления начальных индексов после сортировки) - эффективность футболиста и входной индекс.

Отсортируем входной массив методом быстрой сортировки, назовем его $a'$. Получим массив, в котором эффективность всех футболистов отсортирована в порядке возрастания. Создадим вспомогательный массив $s$, где $s_i = s_{i-1} + a'_i$. Для суммы эффективностей каждой пары игроков $a'_k$,$a'_{k+1}$ найдем наиболее эффективного футболиста, не нарушающего условие сплоченности $a'_j$,

$$a'_j = \max_{0 \le i < n;\ a'_j \le (a'_k + a'_{k+1})} a'_i$$

Создадим впомогательный массив $r$, в который будем записывать суммарную эффективность игроков из $[a'_k, ..., a'_j]$, а в массив $p$ запишем индекс $k$ (для восстановления комманды игроков). Найдем в массиве $r$ максимальное значение $r_m$. Это и есть наибольшая суммарная эффективность, удовлетворяющая условию сплоченности, а $a'_{p_m} ... \ a'_m$ - искомая команда игроков. Восстановить исходные индексы не состаляет труда, так как во входном массиве хранятся изначальные индексы игроков до сортировки.

**Алгоритм:**

1. Запишем во входной массив $a$ пару чисел: $e_i$ - эффективность, $i$ - начальный индекс.

2. Отсортируем входной массив по значению $e$ в порядке возрастания, назавем его $a'$.

3. Создадим вспомогательный массив $s$ , где $s_i = s_{i-1} + a'_i$.

4. Для каждай пары $a'_k$,$a'_{k+1}$:

   - найдем $a'_j$, такой что $a'_j = \max\limits_{0 \le i < n;\ a'_j \le (a'_k + a'_{k+1})} a'_i$;
   - найдем $r_j = s_j - s_k$ - максимальная эффективность команды, включающая игрока $a'_j$;
   - $p_j = k$;

5. Найдем в массиве $r$ максимальное значение $r_m$ - наибольшая суммарная эффективность, удовлетворяющая условию сплоченности.

6. Из $a'_{p_m} ... \ a'_m$ найдем исходные индексы $i$, записанные на шаге 1 и запишем их в массив $I$.

7. Отсортируем массив $I$ в порядке возрастания, получим результирующий массив индексов.

# 2 Доказательство корректнсти

Предположим, что найденный набор игроков $[a'_k, a'_j]$ не является максимально эффективным, но в таком случае существует другой игрок $a'_m$ более эффективный, чем $a'_j$, но тогда $a'_j$ не является максимальным, что протеворечит ранее описаным условиям

$$a'_j = \max_{0 \le i < n;\ a'_j \le (a'_k + a'_{k+1})} a'_i$$

Таким образом набор игроков $[a'_k, a'_j]$ является набором, имеющим наибольшую суммарную эффективность, удовлетворяющая условию сплоченности.

# 3 Асимптотические оценки

В результате получаем сложность по памяти $O(n)$, так как мы используем 4 массива длиной $n$ ($a$, $s$, $p$ и $r$). Сложность по времени равна $O(n \log n)$, так как мы используем алгоритм быстрой сортировки - $O(n \log n)$ для сортировки входного массива, алгоритм бинарного поиска - $O(\log n)$ для поиска верхней границы в массиве $a'$ для каждой пары элементов, и того $O(n \log n)$. Поиск максимума в массиве $r$ и восстановление результирующей последоветльности индексов из массива $a'$ выполняется не более чем за $O(n)$.