

### Вариант №1:

Отсортируем по отдельности все три массива за  $O(n \log n)$ . Создадим три переменные - счетчика для каждого массива, изначально равные нулю. Будем увеличивать за каждый шаг алгоритма тот счетчик, что указывает на ячейку массива с наименьшим значением, при этом проверяя значения в ячейках на которые указывают счетчики на совпадение, соответственно инкрементируя переменную, отвечающую за подсчет количества совпадений. Также необходимо проверять чтобы наши переменные счетчики не выходили за пределы размера массивов, на которые они указывают. В случае, если переменная счетчик указывает на последнюю ячейку, необходимо увеличивать оставшиеся счетчики до тех пор пока все переменные не станут указывать на последние элементы своих массивов. Таким образом мы пройдем по каждому массиву всего один раз, хоть и с проверками и сравнениями, которые, выполняясь за константу, не изменят сложности подсчета одинаковых элементов в  $O(3n + C) = O(n)$ . Общая сложность алгоритма -  $O(3n \log n + n) = O(n \log n)$

### Вариант №2:

Выполним слияние трех массивов и отсортируем полученный массив размера  $O(3*n)$  за  $O(3*n \log n)$ . После этого линейным проходом по массиву можно будет проверять предыдущий и следующий элементы на совпадение с текущим и увеличивать переменную, отвечающую за подсчет совпадений в благоприятном случае. Из этого следует, что общая сложность сравнима с  $O(3n \log n + 3n) = O(n \log n + n) = O(n \log n)$  Минус этого варианта - необходимо создать новый массив :с