

1 (). На вход задачи поступают три отсортированных массива. Постройте алгоритм, находящий число уникальных элементов в объединении этих массивов.

▷ Установим три указателя на начало каждого из массивов и увеличим число уникальных элементов на нужное значение (если 3 уникальных то на три, если 2 на 2 и если все равны то на 1), в отдельных переменных будем хранить последнее число, записанное из каждого из массивов. Далее шаг следующий: после сдвига сначала оставляем только уникальные элементы на указателях, затем сравниваем каждое из чисел с соответствующим значением в переменной, и увеличиваем количество если не совпали. Если доходим до конца массива, соответствующий указатель останавливается. Итого, мы проходим каждый массив один раз. На каждом шаге делается число сравнений, не зависящих от длин массивов. Полученный алгоритм работает за $O(n)$. ◁

2 (). На вход задачи поступает массив a из n чисел. Постройте алгоритм, находящий число инверсий в массиве, то есть таких пар индексов i, j , что $i < j$ и $a[i] > a[j]$.

▷ ◁

3 (). На прямой задано n отрезков в виде пар концов (l_i, r_i) . Они могут поступать на вход в произвольном порядке. Постройте алгоритм, находящий **множество точек на прямой**, покрытое ровно $\lceil \frac{2n}{3} \rceil$ отрезками.

▷ ◁

4 (). На вход поступает число n и массив a размера $2n + 1$. Постройте алгоритм, находящий число s , минимизирующее сумму $\sum_{i=1}^{2n+1} |a_i - s|$

▷ Сперва заметим, что порядок, в котором стоят элементы массива только меняет местами слагаемые в сумме, то есть сама сумма от него не зависит. Поэтому далее считаем массив отсортированным по возрастанию. Поделим весь массив на пары (a_k, a_{2n-k}) , $k = 0, \dots, n-1$, т. е. так чтобы первый стоял с последним, второй с предпоследним и т. д. Далее рассмотрим конкретную пару (a_k, a_{2n-k}) . Если число s находится между a_k и a_{2n-k} то сумма $|s - a_k| + |s - a_{2n-k}|$ равна $a_{2n-k} - a_k$. Если же s лежит за пределами этого отрезка, сумма будет строго больше $a_{2n-k} - a_k$. Отсюда получаем, что для всех пар число s должно быть внутри образованных этими парами отрезков. Остается заметить, что a_n стоит в паре с самим собой, т. е. образует отрезок из одной точки, лежащий в пересечении всех других отрезков. Тем самым, для минимизации суммы необходимо и достаточно, чтобы s совпадало с a_n . Так как мы предполагали, что массив отсортирован по возрастанию, для исходного массива s должно быть равно $a_{(n)}$ — n -ой порядковой статистике. Алгоритм находящий n -ую порядковую статистику работает за $O(n)$. ◁