Бабушкин А.

1 Обязательные задачи

1.

a) Нашли медиану, сделали массив из элементов `|pos(x) - pos(median)|`, нашли там k-ую порядковую, выписали все элементы, которые меньше этой k-ой.
b) Нашли медиану, сделали массив из элементов `|x - median|`, нашли там k-ую порядковую, выписали все элементы, которые меньше этой k-ой.

2.

a) $f = \sum_i (w_i(x_i - x^*)^2), f' = 0$ при $x^* = \frac{w_i x_i}{n}, O(n)$.
b) По каждой из осей можно решать независимо, то есть у нас есть задача $\sum_i (w_i|x_i - x^*|) \to min$, а это просто найти взвешенную медиану, $O(n)$.
c) Повернём на $\frac{\pi}{4}$, теперь вместро суммы $|x_i - x^*| + |y_i + y^*|$ у нас $max(|x_i - x^*|, |y_i - y^*|) \to$ ответ на задачу – максимум из максимумов, то есть снова независмые задачи по каждой из координат. Надо решить задачу $max_i(w_i|x_i - x^*|) \to min$. Это можно представить как отрезки, расширяющиеся со скоростью w_i, и нам надо найти первый момент времени, когда их пересечение непусто. Давайте посмотрим, где будет находиться самая левая из правых границ отрезков в каждый момент времени. Это возврастающая функция и она кусочно-линейная, её можно построить за $O(sort{+}n)$, отсортировав точки по w_i и построив пересечение полуплоскостей. Аналогично строим для самой правой из левых границ. Нам нужен момент времени, когда самая правая из левых $\geq$ самой левой из правых $\to$ надо найти, где две кусочно-линейные функции пересекаются, это делается двумя указателями. Итого $O(sort + n)$.

3. Отсортируем массив. Теперь если мы поставим две точки куда-то, то одна будет отвечать за некоторый префикс этого массива, а другая – за оставшийся суффикс. Пусть мы решили, за какой префикс должна отвечать первая точка. Тогда нам надо её поставить во взвешенную медиану этого префикса, а вторую – во взвешенную медиану оставшегося суффикса. Если мы будем перебирать этот предполагаемый префикс слева направо, то эти медианы можно поддерживать, как и суммы $\to$ можно решить за $O(sort + n)$ с помощью трёх указателей.

2 Дополнительные задачи

1. При фиксированном y^* и движении x^* слева направо функция будет выпуклой вниз, и наоборот $\to$ давайте сделаем два вложенных тернарных поиска.

2. Пусть `p[i][j]` – позиция, куда лучше всего поставить последнюю точку, если мы хотим оптимально разбить префикс длины `i` с помощью `j` точек. Заметим, что `dp[i][j]` $\leq$ `dp[i + 1][j]` $\to$ можно сделать Divide-and-conquer, пересчитывая p по слоям (разбить с помощью j точек – это выбрать последнюю и потом

разбить оставшийся префикс с помощью j - 1 точки), решение за $O(nk \log n) = O(n \log n)$ в нашем случае.