1 Обязательная часть

2. ***Поиск точки***

   - $\sum_i [w_i(x_i - x^*)^2] \to min$
     Возьмем производную. Получим $\sum_i [w_i|x_i - x^*|] \to min$. Эту задачу мы уже решали на практике. Ответ - центр масс.
   - $\sum_i [w_i(|x_i - x^*| + |y_i - y^*|)] \to min$
     Эту задачу можно решать независимо по x и y. Далее решение для координаты x. Отсортируем точки и пойдем по ним слева направо. При этом поодерживаем сумму высеов левее нас и правее нас. Как только нашли такую точку, что разность этих сумм обратилась в ноль или сменила знак - нашли нашу x^*. То же самое делаем для y^*.
   - $max_i[w_i(|x_i - x^*| + |y_i - y^*|)] \to min$
     Будем действовать как в задаче 2e из практики (в разборе она 2b почему-то). Повернем координаты на 45 градусов. Наша новая метрика - $d(w_i(|x_i - x^*| + |y_i - y^*|))$. То есть "квадраты множества, находящиеся на расстоянии $max(w_i \cdot |x_i - x^*|, w_i \cdot |y_i - y^*|)$. Итак, новая функция - $max(w_i \cdot max(|x_i - x^*|, |y_i - y^*|)) \to min$.

     Эта задача тоже независима по координатам. Можем решить по каждой координате отдельно, а затем выбрать максимум из полученных результатов. А так как по одной координате уже умеем решать, то все хорошо. Осталось только вернуть координаты на место, а это мы умеем делать за линейное время. То есть время работы будет $\mathcal{O}(sort + n)$

3. ***Поиск двух точек***

   Пусть $q_1 < q_2$. Будем искать ответ в том случае, когда первые i точек ближе к q_1, а вторые к q_2. То есть нашу сумму можем разбить на две независимых. В таком случае искать точки для конкретного i за линейное время мы умеем с практики. При переходе от i к $i + 1$ обе точки q будут увеличиваться. Чтобы найти их новую позицию, вычтем из суммарного веса слева w_{i+1}, а справа добавим. И двигаем наши q, пока веса слева и справа не сравняются. Пробуем обновить уже имеющийся ответ (если стало меньше, то хорошо).

4. ***Поиск многих статистик***

   Пусть A - наш массив из n чисел, P - массив из m чисел. Создадим массив B размера n. Пробежимся по P и для каждого $x[i]$ запишем, сколько раз он встречаля в массиве P.

   Находим медиану в нашем B из всех существующих. За линию находим эту статистику. Затем выпускаемся от частей B слева и справа медианы. В каждой из этих частей нужно найти уже в 2 раза меньше статистик. Глубина этой рекурсии будет $\log m$.

   Работает за нужное время. m заполняем массив B. n - ищем стастику, $\log m$ - глубину рекурсии. В итоге $\mathcal{O}(m + n \log m)$.

5. ***Поиск статистики***

В детерминированном алгоритме для 5 элементов мы получали сложность $\mathcal{O}(n)$ следующим образом:

$$T(n) = T(\lceil \frac{n}{5} \rceil) + T(\lfloor \frac{7n}{10} \rfloor) + \mathcal{O}(n)$$

Что откуда бралось: $T(\lceil \frac{n}{5} \rceil)$ - рекурсивынй поиск медианы из $\lceil \frac{n}{5} \rceil$ уже найденных, $\mathcal{O}(n)$-разбиение на $\lceil \frac{n}{5} \rceil$ групп, поиск в них медианы сортировкой вставками, Partition массива по медиане медиан. Осталось разобраться с $T(\lfloor \frac{7n}{10} \rfloor)$.

Заметим, что не меньше половины медиан из $\lceil \frac{n}{5} \rceil$ будут больше или равны медиане медиан. То есть хоть половина из $\lceil \frac{n}{5} \rceil$ групп даст по три числа, больших медианы медиан. Получаем не менее $3 \cdot (\lceil \frac{1}{2} \lceil \frac{n}{5} \rceil \rceil) \geq \frac{3n}{10}$. То есть алгоритм, рекурсивно запускаемый, от одной из частей (больше медианы медиан или меньше), будет обрабатывать на больше $\frac{7n}{10}$ элементов.

Из рекурентного $T(n) = T(\lceil \frac{n}{5} \rceil) + T(\lfloor \frac{7n}{10} \rfloor) + \mathcal{O}(n)$ получаем время работы $\mathcal{O}(n)$.

- *7*

  Что изменилось? Количество элементов в группе. Что изменится для рекурентной формулы?

  $$\lceil \tfrac{n}{5} \rceil \to \lceil \tfrac{n}{7} \rceil$$
  $$3 \cdot (\lceil \tfrac{1}{2} \lceil \tfrac{n}{5} \rceil \rceil) \to 4 \cdot (\lceil \tfrac{1}{2} \lceil \tfrac{n}{7} \rceil \rceil) \geq \tfrac{5n}{7}$$

  Получаем $T(n) = T(\lceil \frac{n}{7} \rceil) + T(\lfloor \frac{5n}{7} \rfloor) + \mathcal{O}(n)$. Докажем, что сложность $\mathcal{O}(n)$.
  $cn \geq \mathcal{O}(n) + \lceil \frac{n}{7} \rceil + \lfloor \frac{5n}{7} \rfloor \geq \mathcal{O}(n) + \frac{n}{7} + \frac{5n}{7} - 1 = \mathcal{O}(n) + \frac{6n}{7} - 1$. Очевидно, что мы можем подобрать такое c, что неравенство будет выполняться.

- *3*

  Что изменилось? Количество элементов в группе. Что изменится для рекурентной формулы?

  $$\lceil \tfrac{n}{5} \rceil \to \lceil \tfrac{n}{3} \rceil$$
  $$3 \cdot (\lceil \tfrac{1}{2} \lceil \tfrac{n}{3} \rceil \rceil) \to 2 \cdot (\lceil \tfrac{1}{2} \lceil \tfrac{n}{3} \rceil \rceil) \geq \tfrac{2n}{3}$$

  Получаем $T(n) = T(\lceil \frac{n}{3} \rceil) + T(\lfloor \frac{2n}{3} \rfloor) + \mathcal{O}(n)$. Докажем, что сложность $\mathcal{O}(n \log n)$.

  $$c \cdot n \log n \geq \mathcal{O}(n) + \tfrac{n}{3} \log \tfrac{n}{3} + \tfrac{2n}{3} \log \tfrac{2n}{3} = \mathcal{O}(n) + \tfrac{n}{3} (\log \tfrac{n}{3} + 2 \log \tfrac{2n}{3}) =$$
  $$\mathcal{O}(n) + n \log n + (\tfrac{2}{3} - \log 3)$$

  Очевидно, что можем подобрать такой c, что неравенство будет выполняться. Оценка $\mathcal{O}(n)$ была бы неверна, так как сумма коэффицентов перед n не меньше единицы $(\frac{1}{3} + \frac{2}{3})$.