Практика по алгоритмам #6

1. QuickSort

Посчитаем точно матожидание числа сравнений, которое делает Quicksort на перестановке из n элементов. Сумма гармонического ряда.

2. Задачи про бинарный поиск

1. Найти корень вещественного числа с использованием стандартных арифметических операций.
2. Найти расстояние от точки до прямой в 3D (в nD).
3. Корни многочлена
   a) Найти корень многочлена нечетной степени за $\mathcal{O}(\log M)$.
   b) Зная все корни производной, найти все вещественные корни многочлена за $\mathcal{O}(n \log M)$.
   c) Найти все вещественные корни многочлена за $\mathcal{O}(n^2 \log M)$.
4. Бинарный поиск на массиве
   a) Выразить `upper_bound` для целых чисел через `lower_bound`.
   b) Докажите, что нельзя сделать и `lower_bound`, и `upper_bound` одновременно, используя в худшем случае меньше чем $2\log_2 n + \mathcal{O}(1)$ сравнений?
   c) Сделать предподсчет за $\mathcal{O}(n \log n)$, чтобы за $\mathcal{O}(\log n)$ отвечать на запрос "сколько раз число x встречается на отрезке $[l..r]$"?

3. Задачи про поиск точки

Каждую из предложенных задач можно решить за время $\mathcal{O}(n) + \mathcal{O}(sort)$, тем не менее решения за линию от n на полилогарифм тоже приветствуются. Для разминки предлагается продифференцировать e^x и найти за $\mathcal{O}(n)$ площадь пересечения n прямоугольников со сторонами параллельными осям координат.

1. Даны n точек на прямой x_i. Найти точку x^*:
   a) $\sum_i |x_i - x^*| \to \min$
   b) $\sum_i (x_i - x^*)^2 \to \min$
   c) $\max_i |x_i - x^*| \to \min$
   d) $\max_i (x_i - x^*)^2 \to \min$
2. Даны n точек на плоскости (x_i, y_i). Найти точку (x^*, y^*):
   a) $\sum_i \Big[\max(|x_i - x^*|, |y_i - y^*|)\Big] \to \min$
   b) $\sum_i \Big[|x_i - x^*| + |y_i - y^*|\Big] \to \min$
   c) $\sum_i \Big[(x_i - x^*)^2 + (y_i - y^*)^2\Big] \to \min$
   d) $\max_i \Big[\max(|x_i - x^*|, |y_i - y^*|)\Big] \to \min$
   e) $\max_i \Big[|x_i - x^*| + |y_i - y^*|\Big] \to \min$
   f) $\max_i \Big[(x_i - x^*)^2 + (y_i - y^*)^2\Big] \to \min$
3. На прямой расположено n точек $p_1, p_2, \ldots, p_n$. Каждая точка имеет вес $w_i \geq 0$. Требуется найти точку q: $\sum_i \Big[w_i \cdot |p_i - q|\Big] \to \min$.

4. Домашнее задание

4.1. Обязательная часть

1. (1) Предложите алгоритм на основе бинарного поиска для поиска остовного дерева графа, в котором максимальный вес ребра минимален.
2. (1) Даны n точек на прямой x_i с весами $w_i \geq 0$. Найти точку x^*:
   $\sum_i \Big[w_i(x_i - x^*)^2\Big] \to \min$
3. (1) Даны n точек на плоскости (x_i, y_i) с весами $w_i \geq 0$. Найти точку (x^*, y^*):
   $\sum_i \Big[w_i(|x_i - x^*| + |y_i - y^*|)\Big] \to \min$
4. (2) Даны n точек на плоскости (x_i, y_i) с весами $w_i \geq 0$. Найти точку (x^*, y^*):
   $\max_i \Big[w_i(|x_i - x^*| + |y_i - y^*|)\Big] \to \min$. Дополнительный балл можно заработать, решив эту задачу $\mathcal{O}(sort + n)$.
5. (3) На прямой расположено n точек p_1, p_2, $\ldots$, p_n. Каждая точка имеет вес $w_i \geq 0$. Выбрать две точки q_1, q_2: $\sum_i \Big[w_i \cdot \min(|p_i - q_1|, |p_i - q_2|)\Big] \to \min$.
6. (3) Вариация на тему `Pairing heap`: докажите, что если в `DeleteMin` вместо процедуры `Pairing` список $[a_1, \ldots, a_n]$ произвольным образом разбить на пары $[(a_{i_1}, a_{j_1}), \ldots, (a_{i_{n/2}}, a_{j_{n/2}})]$, то амортизированное время работы операции `DeleteMin` будет $\mathcal{O}(\sqrt{n})$.

4.2. Дополнительная часть

1. (6) Есть n отрезков на окружности. Выбрать максимальное по размеру множество, покрывающее каждую точку не более чем 2 раза.
   a) (3 из 6) $\mathcal{O}(n^3)$
   b) (4 из 6) $o(n^3)$
   c) (6 из 6) $\mathcal{O}(n \cdot Poly(\log n))$
2. (4) Даны n точек на плоскости (x_i, y_i) с весами $w_i \geq 0$. Найти точку (x^*, y^*):
   $\max_i \Big[w_i((x_i - x^*)^2 + (y_i - y^*)^2)\Big] \to \min$.
   Требуется решение за линию на полилогарифм.
3. (5) На прямой расположено n точек p_1, p_2, $\ldots$, p_n. Каждая точка имеет вес $w_i \geq 0$. Выбрать три точки q_1, q_2, q_3: $\sum_i \Big[w_i \cdot \min(|p_i - q_1|, |p_i - q_2|, |p_i - q_3|)\Big] \to \min$.
   Требуется решение за линию на полилогарифм.