

SPb HSE, Магистры ВШЭ, осень 2024/25

Практика по алгоритмам #2

Сортировки, медианы, кучи

12 сентября

Собрано 12 сентября 2024 г. в 11:37

---

## Содержание

1. Сортировки, медианы, кучи

1

# Сортировки, медианы, кучи

## 1. Дедлайны

Есть несколько заданий,  $i$ -е нужно делать  $t_i$  времени.

Вам нужно сделать тах число заданий, успеть к общему для всех заданий дедлайну  $D$ .

## 2. Отрезки времени

Даны  $N$  отрезков времени  $[L_i, R_i]$ , проверить, пересекаются ли хотя бы какие-то из них.

## 3. Сортировка без памяти

Есть массив пирожков, содержащий пирожки трёх типов: 1, 2, 3.

Отсортируйте его по типу пирожков за  $\mathcal{O}(n)$  времени, используя  $\mathcal{O}(\log n)$  бит памяти.

## 4. Точки на прямой

Даны  $n$  точек  $x_i$  на прямой. Выберите точку  $x^*$ :  $\sum_i |x_i - x^*| \rightarrow \min$ . Легенда: хотим на шоссе построить магазин, чтобы среднее расстояние от поселений до него было поменьше.

## 5. Куча точек

А теперь точки добавляются. После каждого добавления выдавать оптимальную  $x^*$ .

## 6. Точки с весами на прямой

Даны  $n$  точек  $x_i$  на прямой, у точек есть веса  $w_i > 0$ . Выберите точку  $x^*$ :  $\sum_i w_i |x_i - x^*| \rightarrow \min$ . Легенда: та же, теперь учитываем, что у поселений разное население.

## 7. Куча пар точек

На прямой живут точки. Иногда добавляются/удаляются новые. Поддерживать пару ближайших.

## 8. Моделирование

$n$  человек бегут по прямой в одну сторону,  $i$ -й человек начинает в  $x_i$ , бежит со скоростью  $v_i$ , все скорости различны. Если  $i$  догоняет  $j$ , то  $j$  расстраивается и уходит. Для каждого человека узнайте, скольких он расстроит.

## 9. Ускорение SiftUp

Модифицируйте операцию SiftUp для бинарной кучи так, чтобы она по-прежнему работала за  $\mathcal{O}(\log n)$ , но при этом делала лишь  $\mathcal{O}(\log \log n)$  сравнений.

## 10. Статистика в бинарной куче

Дана бинарная min-куча. Найти  $k$ -ую статистику за: (a)  $\mathcal{O}(k \log n)$ ; (b)  $\mathcal{O}(k^2)$ ; (c)  $\mathcal{O}(k \log k)$ .

## 11. (\*) Anti-QuickSort test

Пусть в качестве разбивающего элемента всегда берётся

(a) первый элемент  $a[l]$ ; (b) средний элемент:  $a[\lfloor \frac{l+r}{2} \rfloor]$ .

Построить массив длины  $n$ , на котором QuickSort отработает за  $\Omega(n^2)$ .

## 12. (\*) Второй максимум

Найти второй максимум в массиве за  $n + \mathcal{O}(\log n)$  сравнений.