

# Алгоритмы и модели вычислений, группы 774 — 775,

## задание 7

---

### Задание 7

#### Сортировки:

1. Предложите алгоритм, который принимает на вход  $N$  длин отрезков  $a_1, a_2, \dots, a_N$  и находит поднабор с максимальной суммарной длиной такой, что из любых трех отрезков этого поднабора можно сложить треугольник (возможно вырожденный). Время работы  $O(N \log N)$ .
2. В некоторой одномерной деревне расположены в точках  $a_1, a_2, \dots, a_N$  расположены магазины, которые начинают работать одновременно, а заканчивают спустя  $b_1, b_2, \dots, b_n$  минут. Вы можете преодолеть расстояние  $S$  в минуту. Сможете ли вы успеть обойти все магазины в любом порядке до того, как они закроются? Утром перед открытием вы можете подготовиться и к открытию прийти в нужную точку. Вход алгоритма:  $N$ , числа  $a_i, b_i$ , скорость  $S$ . Выход: "YES" если обойти магазины возможно и "NO" иначе. Время работы  $O(N \log N)$ .
3. Инверсией в последовательности  $a_1, a_2, \dots, a_N$  называется такая пара индексов  $i < j$ , что  $a_i > a_j$ . Предложите алгоритм, позволяющий найти число инверсий за время  $O(N \log N)$ .

#### Кучи:

1. Докажите, что описанная на семинаре процедура построения кучи снизу вверх действительно работает за линейное время в худшем случае.
2. Пусть задана бинарная куча из  $N$  элементов (в корне минимум), предложите алгоритм, который за время  $O(K \log K)$  найдет  $K$  наименьших элементов из нее.
3. Предложите структуру данных, которая для заранее фиксированного числа  $k$ , не зависящего от входа, умеет выполнять следующие операции:
  - $Insert(x)$  - вставляет в структуру элемент  $x$  за время  $O(\log N)$ .
  - $Erase(x)$  - удаляет элемент  $x$  за время  $O(\log N)$ , считаем, что этот запрос всегда корректен, т.е. нужный элемент действительно лежит в структуре.
  - $GetKth()$  - возвращает  $k$  порядковую статистику от элементов, лежащих в структура, либо сообщает, что таких элементов меньше  $k$ . Время работы  $O(\log N)$ , но возможно и  $O(1)$ .

Во всех оценках сложности  $N$  - текущее количество элементов в структуре. Также выполнено следующее условие: "если некоторое число  $x$  было вставлено раньше числа  $y$ , то и удалено оно будет раньше".  
*Подсказка: используйте 2 кучи и еще что-то*

#### Графы (DFS, BFS):

1. Предложите  $O(V + E)$  алгоритм поиска расстояний от заданной вершины ориентированного взвешенного графа до всех остальных, если веса ребер могут принимать значения 0 или 1.

2. Предложите  $O(V + E)$  алгоритм поиска цикла в ориентированном невзвешенном графе.
3. Предложите  $O(V^2 + VE)$  алгоритм поиска кратчайшего цикла в невзвешенном неориентированном графе без петель. Если ответов несколько – верните любой, если ответов нет – сообщите об этом.
4. \* Предложите как можно более эффективный алгоритм, который для неориентированного, взвешенного графа без отрицательных циклов найдет кратчайший простой путь между двумя заданными вершинами, если ребра графа могут иметь вес  $-1$  или  $1$ .

### Графы (Алгоритм Дейкстры):

Будем считать, что алгоритм Дейкстры внутри себя использует хранилище, основанное на бинарной куче и имеет сложность  $O(E \log V)$ . Предложение имеет под собой ту логику, что реализация на фибоначчевой куче имеет большую константу в асимптотике и практически не используется на практике.

1. Пусть в не ориентированном взвешенном графе, ребра которого имеют положительную длину, выделены  $T$  вершин. Предложите  $O(V + E \log V)$  алгоритм, который найдет длину кратчайшего из расстояний между различными вершинами из множества  $T$ .
2. Пусть в неориентированном взвешенном графе, ребра которого имеют положительную длину, вершины покрашены в желтый, красный, синий и зеленый цвета. Предложите  $O(E \log V)$  алгоритм, который для каждой пары цветов найдет длину кратчайшего пути, начинающегося в вершине первого цвета и заканчивающегося в вершине второго цвета.
3. \* Пусть лабиринт представляет собой взвешенный неориентированный граф с ребрами положительной длины. Вы находитесь в некоторой точке и можете перемещаться по ребрам со скоростью, не превосходящей 2 единиц в секунду. Так же на вашей карте отмечены несколько невидимых для вас привидений, изначально расположенных в вершинах графа и способных произвольным образом перемещаться со скоростью не более 1 единицы в секунду (но может и медленнее, как захочет). Встретившись с привидением в любой точке лабиринта, возможно даже на ребре при движении навстречу, вы погибаете.

На месте вашего старта стоит неподвижная карта мародеров, показывающая текущее положение привидений, а так же схему лабиринта, включающую в себя указание позиции карты и выхода, расположенного в некоторой вершине. Предложите, как вам взглянуть на карту и определить, сможете ли вы гарантированно выйти из лабиринта. Если ответ положительный, он должен так же включать в себя путь до выхода. Алгоритм должен иметь сложность  $O(V + E \log V)$ , чтобы вам не пришлось стоять у карты слишком долго в поисках маршрута.