Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа №3**по курсу «Алгоритмы и структуры данных»

Выполнил:

Студент группы P3212  
Анищенко Анатолий Алексеевич

Преподаватель:

Косяков М.С.

г. Санкт-Петербург  
2020 г.

**Задача №1067 «Disk Tree»**

**Пояснение к примененному алгоритму:**

Никакого алгоритма здесь нет, просто банальная реализация поставленной задачи. Реализуем структуру с полями: имя и указатели на дочерние каталоги.

Для вывода на надо будет просто рекурсивно обойти полученное дерево в глубину.

**Задача №1494 «Монобильярд»**

**Пояснение к примененному алгоритму:**

Попробуем положить шары в стек так же, как их забивал бы Чичиков (таким образом на вершине стека окажется последний шар).

Если текущий шар больше, чем максимальный на данный момент, то считаем, что все в порядке, потому что именно в этом порядке шары попадают в лунку. (Если же он больше на несколько позиций, то, возможно, ревизор не подходил за шарами в момент, пока Чичиков забивал шары с номерами между максимальным и текущим, там собрался еще один стек шаров. Чтобы это проверить, сохраним все шары, которые должны были попасть в лунку в этот промежуток)

Если текущий шар меньше, то надо проверить, соответствует ли этот шар ожидаемому стеку, который мы сохранили выше. Если не соответствует, то Чичиков жульничает.

**Задача №1521 «Военные учения 2»**

**Пояснение к примененному алгоритму:**

Воспользуемся структурой «дерево отрезков», с помощью которого будем хранить сколько солдат осталось на конкретном отрезке. Таким образом мы сможем находить следующего за O(log n) . Удаление происходит за O(log n), т.к. нам нужно пройтись до корня.

Суммарное время работы O(n \* log n)

**Задача №1628 «Белые полосы»**

**Пояснение к примененному алгоритму:**

Добавим фиктивные чёрные точки вначале и конце каждой строчки/столбца.

Отсортируем все точки с левого верхнего угла до правого нижнего: сначала строки, потом столбцы. Каждую полосу, длиной больше 1 добавим в результат, каждую полосу, длиной 1 сохраним на будущее в отдельный вектор. Аналогично отсортируем все точки снова, только теперь сначала столбцы, потом строки. Теперь у нас есть количество вертикальных и горизонтальных полос. Осталось проверить только полосы 1х1.

Если полоса 1х1 встречается в векторе 2 раза, значит - квадрат 1х1 максимален и по вертикали, и по горизонтали => его следует добавить в результат.

Асимптотика работы: O(k \* log k)

**Задача №1650 «Миллиардеры»**

**Пояснение к примененному алгоритму:**

Смоделируем ситуацию, представленную в задаче.

Для решения задачи нам понадобится 2 структуры:

Отсортированное по возрастанию множество городов, в котором храним актуальное на данный день состояние городов. При каждом перелете миллиардера изменяем общую сумму денег в городе и обновляем наше множество.

Во второй структуре данных будем хранить данные об имени миллиардера и городе, в котором он сейчас находится. Это удобнее всего сделать при помощи map, где ключом будет имя миллиардера, а значением структура, хранящая количество денег, которые находятся в городе и сколько дней они там находятся.

Все миллиардеры и города хранятся в map, для того чтобы изменения какой-либо информации совершалось за логарифм.

Асимптотика работы: O(k \* log k)

**Исходный код:**

<https://github.com/AnatoliiAnishchenko/Algo_and_DS/tree/master/Part3>