

# Домашнее задание № 3

## Задача 1

Имеется двумерный массив значений, которые можно сравнивать по величине:

`Comparable[][]`. Будем говорить, что этот массив представляет собой совокупность “строк”, причем эти строки могут быть разной длины. Элементы в каждой строке упорядочены по неубыванию значений. Пример такого массива:

```
new Integer[][] {  
    { 5, 7, 10, 23 },  
    { 2, 12 },  
    { 6, 7, 7, 9 },  
    { 1 }  
}
```

Требуется написать функцию, которая, получая такой двумерный массив в качестве аргумента, возвращает список, который содержит все элементы исходного массива, упорядоченные по неубыванию. Для описанного выше примера результатом работы функции должен быть список `[ 1, 2, 5, 6, 7, 7, 7, 9, 10, 12, 23 ]`

Предполагается, что исходный массив может быть достаточно большим. Эффективность работы функции должна быть не хуже, чем  $O(M \log N)$ , где  $M$  - общее количество элементов, а  $N$  - число строк.

*Подсказки:*

1. Сначала собрать все элементы в один массив, а затем отсортировать его - очевидный, но не очень эффективный алгоритм. Скорость его работы будет  $O(M \log M)$ , что хуже, чем требуется.
2. Используйте двоичную кучу.

## Задача 2

Некоторая структура данных, для которой определен итератор, содержит  $n$  элементов, которые можно сравнивать друг с другом по значению. Тип такой структуры можно определить как `Iterable<T>`, при этом `T` реализует интерфейс `Comparable<T>`. Требуется написать функцию `<T extends Comparable<T>> List<T> first(Iterable<T> array, int m)` которая выдает список из  $m$  наименьших элементов этой структуры. Число элементов в структуре  $n$  заведомо (и намного) больше  $m$ . В свою очередь,  $m$  положительно, причем тоже может быть довольно велико. Эффективность работы функции должна быть не хуже, чем  $O(n \log m)$ . Например, если имелся список целых

```
List<Integer> array = Arrays.asList(3, 8, 12, 4, 2, 6, 9, 13, 11);
```

то вызов функции `mins(array, 4)` должен привести к выдаче списка из элементов `[3, 4, 2, 6]` в некотором порядке (не обязательно в том, в каком они появлялись в исходном массиве).

*Указание.* Подумайте, как в этой задаче можно использовать двоичную кучу.