Lab Assignment 3

CS 302 – Advanced Data Structures and File Processing

Problem 1

You are given an array A with n elements and an integer $k < n$. Each element in A is at most k away from its target position. For example, if $k = 2$, an element at index 7 in the sorted array can be at indexes 5, 6, 7, 8, and 9 in the given array. Implement an algorithm that uses a heap to sorts A in $\mathcal{O}(n \log k)$ time.

Problem 2

The company Magic Memories produces magic hard drives. Once a drive with capacity c is used for the first time, it magically creates a new drive with capacity $\lfloor c/2 \rfloor$. The newly created drive is as magical as the old one. You were able to obtain n such drives with different capacities. Unfortunately, you can only use k of them at once. Implement an algorithm that determines in $\mathcal{O}(n \log n)$ time the maximum capacity you can use.

Implementation

You are given two files: *Lab3.java* and *MinHeap.java* (which you can download from canvas). The file *Lab3.java* contains a class `Lab3` with the two functions `problem1` and `problem2`. Implement your solutions in the corresponding functions. **Do not make any changes outside of these two functions (e. g. by adding helper functions); such changes will be undone.** Do not output anything to the terminal. The file *MinHeap.java* implements a min-heap you can use for you solutions.

The program already implemented in the file `Lab3.java` randomly generates test cases. The seed of the random number generator is set to ensure the same test cases whenever to program is executed. Note that the purpose of the tests is for you to avoid major mistakes. **Passing all given tests *does not* imply that your algorithm is correct, especially that is has the expected runtime.**

Submission

For your submission, upload the file *Lab3.java* with your implementation to canvas.

This is an individual assignment. Therefore, a submission is required from each student.

Deadline: On Canvas.