

Week 4 Graded Problem - Quiz 2

Topic: Heapsort

1. In Week 3, you covered the topic of the [Heapsort](#) where you learned its main strength when compared to quicksort and mergesort: it is an in-place sorting algorithm with $O(N \log N)$ worst-case complexity, which makes it an efficient sorting algorithm in terms of both time and space. One of the applications of heapsort is [finding \$k\$ largest \(or smallest\) elements in an array](#). You will use this heap property to solve this week's problem.

Problem Definition

The evil twins of TAs Alperen and Selma (**La'peren and Es'mal**) have abducted them and locked them up in a place from which they will not be able to escape before the end of the semester. The poor BBM202 and BBM204 students have been completely fooled by these malicious TA impostors, and have no idea what terrible injustice is going to befall them. La'peren and Es'mal are very biased and cruel, and they want to punish the struggling students, as well as to favor the ones who are doing well. But they want to do this in a sneaky way, so that the students will not easily detect this deception.

La'peren wants to punish the struggling students by lowering the grades of the top N students with the lowest grades, while Es'mal wants to favor the top M students with the highest grades by increasing their grades even more. The numbers N and M vary according to the day of the week (because they are grumpier during weekdays, especially on Mondays), and can be calculated as follows:

$$\begin{aligned} N &= 8 - \text{day_of_the_week} \\ M &= \text{day_of_the_week} \end{aligned}$$

Here, *day_of_the_week* is simply the number corresponding to the day of the week, e.g., Monday = 1, Sunday = 7, etc.

The evil impostor TAs decided to employ BBM202 students to dig their own grave by coding a Java program that will automatize this process.

Important note: after manipulation, any grade can only take values in the range $[0, 100]$.

For this problem you are expected to complete:

- (a) **maxHeapify** and **minHeapify** functions in *Heap.java* (like [sink\(\)](#) function given on page 19 of the course slides, **but starting from index 0**).
- (b) **rewardStudents** and **punishStudents** functions in *BiasedGrading.java*.

You MUST use this starter code.

Input

The input file name and the amount of points to add/subtract will be given as command line arguments to your Main class in the following format:

```
java Main <inputfile> <points to add or subtract (double)>
```

For example, to run your program on *grades.csv* and add/subtract 2.5 points, the following run command will be issued:

```
java Main grades.csv 2.5
```

The input file is a CSV file that contains lines of randomly generated pairs of student numbers and their corresponding grades delimited by a comma. The sample input file is as follows:

```
21597968,15.79
21999062,36.07
21867000,18.23
21844170,43.83
21686842,1.81
21548740,8.34
21576007,65.29
21777374,20.29
21803220,76.30
```

Output

You are expected to output the altered list of student-grade pairs (delimited by a comma), to the STDOUT. For example, for the given sample run command and the sample *grades.csv*, when run on a Thursday, your output should be:

```
21597968,13.29
21999062,38.57
21867000,15.73
21844170,46.33
21686842,0.0
21548740,5.84
21576007,67.79
21777374,20.29
21803220,78.8
```

Important Rules and Grading Policy

- **You MUST use this starter code.** Do not change any function signatures in the given starter codes, those are given to ensure that you pass the unit tests in autograding. Only complete the TODOs.
- Grading policy:
 - Implement **maxHeapify** function by completing TODOs in *Heap.java*: 25%.
 - Implement **minHeapify** function by completing TODOs in *Heap.java*: 25%.
 - Implement **punishStudents** function by completing TODOs in *BiasedGrading.java*: 25%.
 - Implement **rewardsStudents** function by completing TODOs in *BiasedGrading.java*: 25%.
- Test your codes using the [autograding platform](#) and finally submit them via submit.cs.hacettepe.edu.tr using the same format given below:
 - **<studentID>.zip**
 - * **Main.java**
 - * **Heap.java**
 - * **BiasedGrading.java**
 - * **StudentGrade.java**