CS2400 Summer 2023 Project 2

Total points: 100

Due date: Monday, June 26, 2023

Purpose:

1. Master a binary heap and the array representation.
2. Understand the time complexity of heap operations.

"Please start working on this assignment as early as possible!"

Task Description:

In this project, you are going to build a **max-heap** using array representation in Java. In particular,

- In the file **"heap.java"**,
  - (20 pts) **Implement** two methods of building a max-heap.
    - **Using sequential insertions** (its time complexity: $O(nlogn)$, by successively applying the regular add method).
    - **Using the optimal method** (its time complexity: $O(n)$, the "smart" way that we learned in class).

    For both methods, your implementations need to keep track of how many swaps (swapping parent and child) are required to build a heap.
  - (20 pts) **Implement** the remove method of a max-heap.

- In the file **"testHeap.java"**,
  - (5 pts) **Load** a sequence of integers **from an input file**.
    - "data.txt": This file contains 100 integers (no duplicates, and positive numbers). Each line is an integer.
  - **Perform heap operations** and **Write** the results **into an output file**.
    - (5 pts) Create a max-heap using the **sequential insertions**, for those 100 integers.
    - (5 pts) Output the first 10 integers of your array, into the output file
    - (5 pts) Output the number of swaps performed, into the output file
    - (5 pts) Perform 10 removals on the heap
    - (5 pts) Output the first 10 integers of the resulting array, into the output file

    - (5 pts) Create a max-heap using **the optimal method**, for those 100 integers
    - (5 pts) Output the first 10 integers of your array, into the output file
    - (5 pts) Output the number of swaps performed, into the output file
    - (5 pts) Perform 10 removals on the heap
    - (5 pts) Output the first 10 integers in the resulting array, into the output file

- (5 pts) The final output file should use the format as shown below:

```
======================================================================
Heap built using sequential insertions: 100,94,99,77,93,98,61,68,76,84,...
Number of swaps in the heap creation: 480
Heap after 10 removals: 90,89,62,77,88,53,61,68,76,84,...

Heap built using optimal method: 100,95,99,79,94,98,63,71,78,87,...
Number of swaps in the heap creation: 96
Heap after 10 removals: 90,89,63,79,88,55,62,71,78,87,...
======================================================================
```

This project will be graded based on the quality of your program. Please note that **Java interface and generic data types are NOT required** in this project.

What to Submit?

1. Source codes, including “heap.java” and “testHeap.java”
   Note: Please test your source codes using the **Eclipse** IDE and see if the codes are executable. **Non-executable programs will result in a grade of zero.**
2. Input file (just the given “data.txt”)
3. Output file
4. Please zip all documents as yourname_p2.zip and submit it in Canvas.