University of Windsor

School of Computer Science

COMP-2540 Data Structures and Algorithms - Winter 2023
Lab Assignment 4

Deadline: **March 13/15, 2023** (15 minutes prior to the end of the lab)

Regulations: This assignment must be done **individually**. Any similarity found between your code/answers and another student's, or a carbon-copy of an answer found in the web will be considered cheating. You must implement **your own** Heap and sorting algorithms and not use the built-in ones from the programming language you are using.

Objective: The aim of this assignment is to obtain hands on experience in implementing, testing and understanding the Heap Sort algorithm and the Heap ADT.

Tasks:

1. Design two algorithms in **pseudocode**: one for the insert operation and the other for the removeMin operation, both for a Heap implemented on an array H.
2. Using your favorite programming language, implement the Heap ADT on an array, including the following operations: insert, removeMin, min, size and isEmpty. Use the insert and removeMin algorithms of question #1.
3. Design the Heap-Sort algorithm in **pseudocode**, which uses the Heap ADT of #2. Your algorithm should sort an array of integers in **increasing** order.
4. Using your favorite programming language, implement the Heap-Sort algorithm you designed in question #3.
5. What are the worst and best-case time complexities of Insert, removeMin, and Heap-Sort, in O-notation? Why? Your answer must be based on the algorithms you implemented.
6. Generate an array A of n random integers. Sort array A using Heap-Sort. Do this for arrays of size n = 8, 16, 32, 64, …, 2^{20}. Keep track of the CPU time each algorithm takes to sort these arrays. Comment on the running times you obtained, and compare them to the complexities as discussed in class. Hint: place the CPU times you obtained in a table or a plot.
7. [**Optional] Do the same as #6 for Quicksort, Insertion-Sort and Merge-Sort, and compare the results with those of Heap-Sort.

Submission:

1. Your assignment must be submitted during the lab session in the section you are registered in. Any submission *on or 15 minutes* prior to the end of the lab session will **not** be accepted and a **zero** mark will be given. Late assignments will be given a **zero** mark. Submissions by email will **not** be accepted.
2. Provide the insert, removeMin and Heap-Sort algorithms **in pseudocode** (e.g., written on paper or Word). Code in a programming language like Java, C/C++ or any other will **not** be accepted.
3. Provide the source code for *all* your implementations.
4. Run your Heap-Sort algorithm during the lab and show they work for various sample inputs, including (more inputs may be included during the evaluation in the lab):
    a. 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
    b. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
    c. 1, 3, 5, 7, 9, 10, 8, 6, 4, 2
5. Explain how your Heap and sorting algorithms work. This will be asked when the lab assignment is being submitted and marks will be deducted if not clear how it works.