

Spring 2023 – CS 303 Algorithms and Data Structures  
Lab 4

**Notes:**

- Implement the algorithm and analyze the results using the give input files
- **Deliverables:** Report.pdf file and your code file (please do not send a zip file. If you have more than one class in your code, then submit each file separately through Canvas.)
- Homework report must follow the guidelines provided in the sample report uploaded in Canvas

**Objectives:**

- Implement heap sort using a max-heap
- Compare the performance of insertion sort, merge sort, and heap sort

**Problems**

1. Implement a method to sort a given array using the heap sort algorithm. You can use the algorithm from the textbook (see page 2).
2. Write a driver program to test the heap sort algorithm for the arrays of varying lengths provided in Canvas.
3. Compare the execution time of heap sort with insertion sort implemented in Lab-2 and merge sort implemented in Lab-3. Make sure you use the same array to compare the performance. Use a table or plot to summarize the results and document your observations and analysis in the report.

HEAPSORT( $A$ )

```
1 BUILD-MAX-HEAP( $A$ )
2 for  $i = A.length$  downto 2
3     exchange  $A[1]$  with  $A[i]$ 
4      $A.heap-size = A.heap-size - 1$ 
5     MAX-HEAPIFY( $A, 1$ )
```

BUILD-MAX-HEAP( $A$ )

```
1  $A.heap-size = A.length$ 
2 for  $i = \lfloor A.length/2 \rfloor$  downto 1
3     MAX-HEAPIFY( $A, i$ )
```

MAX-HEAPIFY( $A, i$ )

```
1  $l = \text{LEFT}(i)$ 
2  $r = \text{RIGHT}(i)$ 
3 if  $l \leq A.heap-size$  and  $A[l] > A[i]$ 
4      $largest = l$ 
5 else  $largest = i$ 
6 if  $r \leq A.heap-size$  and  $A[r] > A[largest]$ 
7      $largest = r$ 
8 if  $largest \neq i$ 
9     exchange  $A[i]$  with  $A[largest]$ 
10    MAX-HEAPIFY( $A, largest$ )
```

## Spring 2023 – CS 303 Algorithms and Data Structures

### Lab 4

Note: The above pseudo code assumes that the array indexing is starting from 1. If you are using a programming language that uses array indexing starting from 0, you have to modify the pseudo code accordingly.

#### Submission:

- You are required to submit a written report (.pdf) and your project file (.zip) to the Canvas
- Homework report must follow the guidelines provided in the sample report uploaded in Canvas. Please include the screenshot of your code and outputs of your code at the end of your report.
- Do not forget to submit Independent Completion Form
- When you create the code to read the file use relative path instead of absolute path

#### DATA

1000, 2500,5000,10000,25000,50000,100000,250000,500000. (not 1000000 this time)

#### Grading Rubric

Coding	Implementing Algorithms	20 points
	Producing Correct Outputs	20 points
Report	Explaining the algorithms used	10 points
	Displaying the output with a graph or table	20 points
	Comparing the outputs and discussing the time complexity of algorithms	20 points
	Correct submissions of the files (ICF, Code.zip, report.pdf)	10 points