

## Introduction

In this lab you will compare the running times of heap sort and merge sort algorithms. Submit your answers to the questions below in a text file (e.g. Word document). Name your file in name\_surname.docx format. Submit your solution document and Java codes as a folder in name\_surname format to TA through a USB flash disk.

You can use the code templates in `heap.java` in this lab.

## Problem Statement

Given an array of integers sort the numbers in this array in ascending order. You can implement heap by an array.

## Assignment

1. (a) Implement Java methods for the heap sort algorithm given below.

PARENT( $i$ )

1 **return**  $\lfloor i/2 \rfloor$

LEFT( $i$ )

1 **return**  $2i$

RIGHT( $i$ )

1 **return**  $2i + 1$

MAX-HEAPIFY( $A, i$ )

1  $l = \text{LEFT}(i)$

2  $r = \text{RIGHT}(i)$

3 **if**  $l \leq A.\text{heap-size}$  and  $A[l] > A[i]$

4      $\text{largest} = l$

5 **else**  $\text{largest} = i$

6 **if**  $r \leq A.\text{heap-size}$  and  $A[r] > A[\text{largest}]$

7      $\text{largest} = r$

8 **if**  $\text{largest} \neq i$

9     exchange  $A[i]$  with  $A[\text{largest}]$

10    MAX-HEAPIFY( $A, \text{largest}$ )

BUILD-MAX-HEAP( $A$ )

1  $A.\text{heap-size} = A.\text{length}$

2 **for**  $i = \lfloor A.\text{length}/2 \rfloor$  **downto** 1

3     MAX-HEAPIFY( $A, i$ )

HEAPSORT( $A$ )

1 BUILD-MAX-HEAP( $A$ )

2 **for**  $i = A.\text{length}$  **downto** 2

3     exchange  $A[1]$  with  $A[i]$

4      $A.\text{heap-size} = A.\text{heap-size} - 1$

5     MAX-HEAPIFY( $A, 1$ )

(b) Test your algorithm by choosing a heap of size 10 (implemented as an array). Initialize your heap by random numbers from 0 to 99. Make sure your program sorts arrays correctly. Include the output of your program for this sample input in your report.

(c) Choose input sizes in the table below, which are powers of 4, and randomly initialize

COMP 301 Analysis of Algorithms, Fall 2022  
Instructor: Zafer Aydın  
Lab Assignment 8

the key values in your array. Compute the running times of heap sort and merge sort in nanoseconds for each of these input sizes and include the table below in your report.

Input size	Heap sort running time	Merge sort running time
4		
64		
256		
1024		
4096		
16384		
65536		
262144		
1048576		
4194304		
16777216		
67108864		

(d) Set the input size to 67108864. Run the heap sort. Open a terminal window and type top. Find the process for heap sort and record the RAM usage in MEM column for this process. Include the RAM usage of heap sort in your report.

(e) Repeat part (d) for merge sort. Does merge sort use more RAM as compared to heap sort?