

## Introduction

In this lab you will compare the running times and RAM usage of quicksort, 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 to Canvas.

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

## Problem Statement

Given an array of integers sort the numbers in this array in ascending order.

## Assignment

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

<pre>PARTITION(A, p, r) 1  x = A[r] 2  i = p - 1 3  for j = p to r - 1 4      if A[j] ≤ x 5          i = i + 1 6          exchange A[i] with A[j] 7  exchange A[i + 1] with A[r] 8  return i + 1</pre>	<pre>RANDOMIZED-PARTITION(A, p, r) 1  i = RANDOM(p, r) 2  exchange A[r] with A[i] 3  return PARTITION(A, p, r)  RANDOMIZED-QUICKSORT(A, p, r) 1  if p &lt; r 2      q = RANDOMIZED-PARTITION(A, p, r) 3      RANDOMIZED-QUICKSORT(A, p, q - 1) 4      RANDOMIZED-QUICKSORT(A, q + 1, r)</pre>
--	---

(b) Test your algorithm by choosing an array of size 10. Initialize your array by random numbers from 0 to 99. Make sure your program sorts the array 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 initialize the values in your array by random numbers from 0 to 99. Compute the running times of quick sort, heap sort and merge sort in nanoseconds for each of these input sizes and include them to the table below. Write a for loop that performs these operations automatically. Do not run them one at a time.

## COMP 301 Analysis of Algorithms

Instructor: Zafer Aydın

## Lab Assignment 10

Input size	Quick sort running time	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 quick sort, heap sort and merge sort for this input size. Open a terminal window and type top. Find the processes for the sorting algorithm you executed and record the RAM usage in MEM column. Include the RAM usage of these algorithms into the table below. Compare and comment on the RAM usage of these sorting algorithms.

Input size	Quick sort RAM	Heap sort RAM	Merge sort RAM
67108864			