

# Programming assignment (60 points)

---

First, you should implement a function that creates an array of "n" random generated integers. Then, you should implement MergeSort, HeapSort, QuickSort, and InsertionSort. Your implementation should be based on the book's pseudo-code.

In each of the following questions, you should PLOT your results in a two dimensional space. You should use a different figure (plot) for each question. In each plot (figure), axis-x should represent the size of the array "n" and axis-y should represent the running time. The axis-x should be in the range of [100, 1500]. You should not increase the size "n" by a single unit for each experiment. Initialize n to 100 and for each experiment increase the size of the array "n" by 100 integers.

Therefore, for each of the above algorithms you should have 15 dots in your plot. Each dot is a point with coordinates (x,y) that represents the running time (y) of each algorithm with respect to the size (x) of the random generated array. For each dot (experiment), you should collect the results of at least 20 "runs" and report the average. As an example, run MergeSort with 20 different input arrays that have size equal to 100. Then, take the average "av" of these 20 "runs" and represent it as a point with coordinates (100 , av) into the two dimensional plot.

If your results are not what you are expecting then you should do more "runs" or increase the arrays' size range. If your results still contradict the theory then you have accomplished to degrade the performance of at least one algorithm.

## Questions:

---

(20 Points) Plot (in the same figure) the running time of each algorithm as the size of the array increases. The running time of each algorithm should be plotted as a continuous function.

(20 Points) Convert recursive HeapSort into iterative Heapsort (as you were asked in problem 2 of HW3) and plot their running times. Did you expect these results? Why?

(20 Points) Plot the running time of the QuickSort algorithm when it sorts already sorted arrays vs when it sorts arrays sorted in decreasing order. Do the same for InsertionSort and HeapSort.

Briefly explain your results.

For this programming assignment you should use C++ or Java . Please state in your report which function did you use to time the running time of the algorithms. You should submit a pdf report and your code (implementation) through blackboard. The report should include the plots and your answers to each question.