

## Assignment 3

1. Write a program to sort an array, *arr*, consisting  $n$  numbers using the divide and conquer approach - Use only merge sort. (1) The divide step should split the array into two (nearly) equal sub-arrays. (2) The divide step should split the array into three (nearly) equal sub-arrays. Answer the following questions.
  - 1.1. (T) Write pseudocodes to design the algorithms for above mentioned computational problem. Both algorithms should sort the data by dividing them into two and three (nearly) equal sub-arrays respectively.
  - 1.2. (T) Analyze the time complexity of both algorithms (split the array into two and three sub-arrays) using the recursion tree method (Include the handwritten analysis of these algorithms as an image in the latex/word file. Make sure that the images/contents are readable.).
  - 1.3. (L) Provide the details of Hardware/Software you used to implement algorithms and to measure the time.
  - 1.4. (L) Submit the code (complete programs).
  - 1.5. (L) Measure the best-case time, average-case time and worst-case time of the above two algorithms for all ten files (Assignment 1). Plot a graph.
  - 1.6. (L) Compare the best-case performance of bubble sort, selection sort, insertion sort, and merge sort for all ten files. Plot a graph.
  - 1.7. (L) Compare the average-case performance of bubble sort, selection sort, insertion sort, and merge sort for all ten files. Plot a graph.
  - 1.8. (L) Compare the worst-case performance of bubble sort, selection sort, insertion sort, and merge sort for all ten files. Plot a graph.