

CS3310 Project #1 (120 Points)  
Spring 2020  
Instructor: Tannaz R.Damavandi  
**Due Date: Friday – 03/06/2020 at 11:59 pm.**

### **Task #1 - Sorting Algorithms (20 Pts)**

Program **Merge Sort** and **Quick Sort** on your computer to sort a list of  $n$  elements. Carry out a complete test of your algorithms with  $n = 10,000, 20,000, 50,000, 100,000, \dots$  (up to the largest size of  $n$  that your computer can handle – You should stop the program for any method if it takes more than 10 minutes to run). Draw a graph to compare the growth rates of these two algorithms.

### **Task #2- Tower of Hanoi (50 Pts)**

**Problem definition:**

The famous Tower of Hanoi is a classic problem that involves 3 pegs and  $n$  disks of varying size stacked on top of each other from the largest to smallest. (from the bottom to the top). This problem requires you to stack all the disks in the same way but on a different peg following these rules:

- 1- Only one disk can be moved at a time.
  - 2- Only the upper disk can be moved at each step.
  - 3- No larger disk may be placed on the smaller one.
- a. Program the Tower of Hanoi problem. Carry out a complete test of your algorithm with  $n = 2, 4, 8, 16, 32, \dots$  (up to the largest size of  $n$  that your computer can handle – You should stop the program if it takes more than 10 minutes to run) and graph its growth rate.
  - b. Find the recurrence relation and time complexity of this algorithm.

### **Task #3- Matrix Multiplication (50 Pts)**

Program both **Classical** and **Strassen's Matrix Multiplication** that are covered in the class. Let the matrix size be  $n \times n$ . Carry out a complete test of your algorithms with  $n = 2, 4, 8, 16, 32, 64, 128, 256, \dots$  (up to the largest size of  $n$  that your computer can handle - You should stop the program for any method if it takes more than 10 minutes to run). Draw a graph to compare the growth rates of these two algorithms.

### **What to Submit?**

1. Java or Python source codes for each task (**Please comment your code properly**)
2. A detailed report and explanation together with graphs comparing your algorithms.
4. Readme.txt (Please describe how to run your code)
5. Please zip all documents as yourname\_project1.zip and submit it on blackboard.

*Discussion among students is encouraged, but I expect each student to hand in original work.*