Homework 1 (5pt. + extra 1pt.)

Submission instruction:

Submit one single pdf file for this homework including both coding problems and analysis problems.

For coding problems, copy and paste your codes. Report your results.

For analysis problems, either type or hand-write and scan.

Question 1 (2 pt.) Coding: write programs of insertion sort, and mergesort. Find the input size n, that mergesort starts to beat insertion sort in terms of the worst-case running time. You can use clock_t function (or other time function for higher precision) to obtain running time. You need to set your input such that it results in the worst-case running time. Report running time of each algorithm for representative input sizes n. And show how you find the n that mergesort starts to beat insertion sort.

Question 2 (1pt.) You are given with an array {10, 5, 7, 9, 8, 3}. Show the arrangement of the array for each iteration during insertion sort. You are given with the same array. Show the arrangement of the array for each iteration of the Partition subroutine of quicksort and the result of Partition subroutine.

Question 3 (1pt.) True or False

$$n + 3 \in \Omega(n)$$

$$n + 3 \in O(n^2)$$

$$n + 3 \in \Theta(n^2)$$

$$2^{n+1} \in O(n + 1)$$

$$2^{n+1} \in \Theta(2^n)$$

Question 4 (1pt.) Using the master method, determine T(n) for the following recurrance.

$$T(n) = 8T\left(\frac{n}{2}\right) + n$$

$$T(n) = 8T\left(\frac{n}{2}\right) + n^2$$

$$T(n) = 8T\left(\frac{n}{2}\right) + n^3$$

$$T(n) = 8T\left(\frac{n}{2}\right) + n^4$$

Question 5 (extra 1pt.) Draw recursion tree for $T(n) = 8T\left(\frac{n}{2}\right) + n$. And prove the obtained T(n) by substitution method.