Algorithm 2019 Spring

Homework 1

(Chapter 1~ Chapter 4)

P.S. $\lg n = \log_2 n$

1. **(10pts) Explain why the statement, "The running time of algorithm A is at least O(n^2)." is meaningless.**

2. **(10pts) Come up with a real-world problem in which only the best solution will do. Then come up with one in which a solution that is "approximately" the best is good enough.**

3. **(10pts) Suppose we are comparing implementations of insertion sort and merge sort on the same machine. For input of size n, insertion sort runs in $8n^2$ steps, while merge sort runs in $64n\lg n$ steps. For which values of n does insertion sort beat merge sort?**

4. **(10pts) Prove $\lg(n!) = \Theta(n \lg n)$.**

5. **(10pts) Partition the following functions by their asymptotic order. (That is, f and g are in the same partition if, and only if, $f \in \Theta(g)$.) Then list them from the lowest asymptotic order to highest asymptotic order.**

$$n,\ 2^n,\ n\lg n,\ n^3,\ n^2,\ 7n^5 - n^3 + n,\ n^2 + \lg n,\ e^n,$$
$$\sqrt{n},\ 2^{n-1},\ \lg\lg n,\ \lg n,\ \lg^2 n,\ n!,\ n^{1+\varepsilon}(0 < \varepsilon < 1)$$

6. **(10pts) Given a sequence of numbers a1, a2, a3, a4, a5, in which condition will the insertion sort has the most and the least swap to sort the sequence to ascending order?**

7. **(10pts) Answer "true" or "false" first, then explain the reason.**
   (a) (5pts) $2^{n+2} = O(2^n)$
   (b) (5pts) $2^{2n} = O\ (2^n)$

8. (20pts) Give asymptotic upper and lower bounds for T(n) in each of the following recurrences. Assume that T(n) is constant for sufficiently small n. Make your bounds as tight as possible, and justify your answers.

a) (5pts) $T(n) = T(9n/10) + n$ (Use Master Theorem)

b) (5pts) $T(n) = 4T(n/2) + n^2$ (Use Master Theorem)

c) (5pts) $T(n) = T(\sqrt{n}) + 1, T(2) = 1$ (Use substitution method)

d) (5pts) $T(n) = T(n/4) + T(3n/4) + n$ (Use recursion-tree method)

9. (10pts) Given the definition of Fibonacci number:

$$F_n = F_{n-1} + F_{n-2}, \text{ for } n \geqq 2$$

$$F_0 = 0, F_1 = 1$$

(1) Write a *Pseudocode* to compute F_n with *recursive* method. Please analyze your time complexity and give the answer with Asymptotic notation.

(2) Write a *Pseudocode* to compute F_n with *iterative (loop)* method. Please analyze your time complexity and give the answer with Asymptotic notation.

(Bonus 10pts)：

(This is a bonus question. You can choose to answer it or not. However, the highest score is still 100pts.)

Give asymptotic tight bounds for T(n) in the following recurrences.

$$T(n) = 27T(n/3) + \Theta(n^3/\lg n)$$