Assignment 1 (10 points for each question)

1. (Textbook 1.2-3 page 14) What is the smallest value of *n* such that an algorithm whose running time is $100n^2$ runs faster than an algorithm whose running time is 2^n on the same machine?

2. (Textbook 2.1-1 page 22) Use Figure 2.2 as a model, illustrate the operation of INSERTION-SORT on the array A = {31, 41, 59, 26, 41, 58}.

3. (Textbook 2.3-1 page 37) Use Figure 2.4 as a model, illustrate the operation of merge sort on the array A = {3, 41, 52, 26, 38, 57, 9, 49}

4. Design an algorithm that, given a set S of n integers and another integer x, determines whether or not there exist two elements in S whose sum is exactly x. (Hint, you may sort S first). Please give the running time of your algorithm in terms of Θ notation.

5. Design an algorithm that, given a set S of n integers and another integer x, determines whether or not there exist k (n>k>2) elements in S whose sum is exactly x. Please give the running time of your algorithm in terms of Θ notation.

6. Prove that $lgn + e^{\frac{1}{n}} \in O(n^t)$

7. Express the function $\frac{n^2}{100} + 2.5n - 200nlgn$ in terms of Θ notation.

8. Use the recursion tree method to determine the asymptotic upper bounds for the Recurrence T(*n*) = 4T(*n*/3) + O(*n*)

9. Use the recursion tree method to determine the asymptotic upper bounds for the Recurrence T(*n*) = 2T(*n*/2) + O(lg*n*)

10. Given two arrays sorted in ascending order. Design an algorithm to merge them together without allocating extra array. Assume one array has enough buffer at the end to hold the other array. Please analyze the running time of your algorithm.