Assignment 1 (10 points for each question)

1. Exercise 2.3-1: Using Figure 2.4 as a model, illustrate the operation of merge sort on the array A = {3, 41, 52, 26, 38, 57, 9, 49}

2. Exercise 2.3-6: Observe that the while loop of lines 5 – 7 of the INSERTION-SORT procedure in Section 2.1 uses a linear search to scan (backward) through the sorted subarray A[1…j-1]. Can we use a binary search instead of a linear search to improve the overall worst-case running time of insertion sort to $\Theta(n\lg n)$?

3. For the MERGE function, the sizes of the L and R arrays are one element longer than n_1 and n_2, respectively. Can you rewrite the merge function with the size of L and R exactly equal to n_1 and n_2?

4. Prove that $e^{\frac{1}{n}} \in O(n^t)$ ($t > 0$)

5. Express the function $\frac{n^3}{100} - 50n - 100lgn$ in terms of Θ notation.

6. Exercise 3.1-6 Prove that the running time of an algorithm is $\Theta(\mathrm{g}(n))$ if and only if its worst-case running time is O(g(n)) and best-case running time is $\Omega(g(n))$.

7. Which is asymptotically larger: lg$n$ or $\sqrt{n}$? Please explain your reason.

8. Prove that $n^{lgc} \in \Omega(c^{lgn})$, where c is a constant and $c > 1$.

9. Use the definition of limits at infinity to prove $(lgx)^p \in o(x^p)$.

   Definition (limits at infinity): Let $f(x)$ be a function defined on $x > K$ for some K. Then we say that, $\lim_{x\to\infty} f(x) = L$ if for every number $\varepsilon > 0$ there is some number $M > 0$ such that $| f(x) – L| < \varepsilon$ whenever $x > M$