1: [10 points] Clearly and briefly describe one advantage and one disadvantage of the worst case time complexity analysis.

2: [10 points] Sort the following growth rate classes in increasing order of time complexity: exponential, quadratic, logarithmic, cubic, and factorial classes.

3: [10 points] Prove that $\frac{1}{3}n^2 - 3n = \Theta(n^2)$.

4: [10 points] Mathematically derive the average time complexity of the sequential search algorithm where the searched item is sequentially compared to each element in the list. Show your derivation process step by step.
5: [10 points] True/False Questions. Just say T( rue) or F( alse) for each of the following questions.

(a) [2 points] 1, 306, 998, 999n³ ∈ O(n³).
(b) [2 points] 5n² + 10n ∈ Ω(n³).
(c) [2 points] 3n⁴ + 9n³ + 7n² + 0.5n ∈ o(n⁴).
(d) [2 points] n lg n ∈ O(n²).
(e) [2 points] n² ∈ Θ(n³).

6: [20 points] Answer the following questions.

(a) [10 points] Develop a divide-and-conquer algorithm that searches a sorted list of n integers to find an arbitrary integer x in the list. Especially, write an algorithm that divides a list into m smaller sublists of equal size where m > 2 and n = mᵏ for k > 0.

(b) [10 points] Analyze the time complexity of your algorithm in terms of Θ by writing recurrence equations and iteratively solving them.

7: [10 points] Answer the following questions.

(a) [2 points] Would you use a recursive divide-and-conquer algorithm to compute the nth Fibonacci number where n > 1? Simply say yes or no.
(b) [8 points] Why or why not? Clearly and briefly justify your answer in part (a).

8: [10 points] Answer the following questions.

(a) [5 points] Assume that there is a list of \( n \) natural numbers. The quicksort algorithm is used to sort these numbers and the first element in a list is used as the pivot for quicksort. What is the worst case input to quicksort in this case? Briefly describe it.

(b) [5 points] Write the recurrence equations for the worst case and analyze the time complexity by solving the equations.

9: [10 points] Sort a list of numbers \([10, 1, 7, 3, 5, 2]\) using the heapsort algorithm.

(a) [5 points] Draw a min-heap consisted of all the elements in the list.
(b) [5 points] Using the min-heap you built in part (a), perform heapsort. Draw figures to show all heapsort steps.