Assignment: Divide and Conquer Approach & Sorting Algorithms

Objective:

The purpose of this assignment is to develop an in-depth understanding of sorting algorithms, particularly those based on the **Divide and Conquer** approach. You will explore, implement, and compare various sorting techniques such as Quick Sort, Merge Sort, Heap Sort, and Shell Sort.

Section 1: Theoretical Questions

1. Divide and Conquer Strategy:

(a) Explain the **Divide and Conquer** approach. How does it differ from other problem-solving techniques?
(b) Give two real-world examples where the Divide and Conquer approach is used apart from sorting algorithms.

2. Quick Sort:

(a) Explain the **working principle** of Quick Sort with an example.
(b) What is the worst-case time complexity of Quick Sort? Under what conditions does it occur?
(c) How can the choice of pivot affect the performance of Quick Sort?

3. Merge Sort:

(a) Explain the **working principle** of Merge Sort with an example.
(b) Why is Merge Sort considered a **stable sorting algorithm**?
(c) Compare the space complexity of Quick Sort and Merge Sort.

4. Heap Sort:

(a) Explain the **heap property** and how Heap Sort utilizes a heap to perform sorting.
(b) What are the advantages of Heap Sort over Quick Sort in terms of **worst-case performance**?
(c) Construct a **Max Heap** from the following array: [5, 3, 8, 4, 1, 2, 7] and then perform Heap Sort on it.

5. Shell Sort:

(a) Explain the **gap sequence** concept in Shell Sort.
(b) Why is Shell Sort considered an improvement over Insertion Sort?
(c) What are the challenges in selecting an optimal gap sequence?

6. Comparative Analysis:

(a) Fill in the following table comparing sorting algorithms:

Algorithm	Best Case	Average Case	Worst Case	Space Complexity	Stability
Quick Sort	?	?	?	?	?
Merge Sort	?	?	?	?	?
Heap Sort	?	?	?	?	?
Shell Sort	?	?	?	?	?

(b) Which sorting algorithm would you recommend for sorting a **very large dataset**? Justify your answer.