

DATA STRUCTURES & ALGORITHMS

#03

Sorting

|  |
| --- |
| Student Name: |
| Roll Number: Section: |
| Work submitted on: |

|  |  |  |  |
| --- | --- | --- | --- |
| **Maximum Marks** | **Performance** | **Viva** | **Total** |
| **Marks Obtained** |  |  |  |
| **Remarks (if any)** |  | | |
|  | | | |
| **Experiment evaluated by** | | | |
| Instructor Name: | | | |
| Signature: | | | |

|  |
| --- |
| Sorting Related Tasks |

**Task 1**

Write a program to generate a list of random integers and then implement each of the three sorting algorithms to sort the list. Compare the run-time complexity of each algorithm and note any differences.

**Task 2:**

Modify your implementation of the bubble sort algorithm to include an early termination condition that stops the algorithm if the list is already sorted. This should help to reduce the algorithm's run-time complexity in cases where the list is already sorted or nearly sorted.

**Task 3:**

Modify your implementation of the selection sort algorithm to implement an optimized version that swaps elements only when necessary. This should help to reduce the number of swaps required by the algorithm and improve its performance.

**Task 4:**

Implement a hybrid sorting algorithm that uses a combination of bubble sort, selection sort, and insertion sort to sort a list. Experiment with different combinations of these algorithms and observe how the run-time complexity of the hybrid algorithm compares to that of each individual algorithm.