

# Major Assignment Set 01

## Searching and Sorting Algorithms

This set of assignments should give students a comprehensive understanding of various searching and sorting algorithms and their implementations in Python.

### **Submission Guidelines:**

- All the assignment questions should be present in a single Python script.
- Write a separate Python methods for each assignment. These methods should be callable from external sources too.
- Provide algorithms/flowcharts/pseudo codes for each assignment.
- Include comments in the code to explain the logic and steps.
- Include identical test cases and report the results for each algorithm. The test cases should contain an unordered list of at least 10 numbers.
- Provide a report analysing the time complexity and performance of each algorithm.

### Assignment 1: Linear Search

1. **Objective:** Implement and test the linear search algorithm.
2. **Tasks:**
  - Develop algorithm/flowchart/pseudo code for the above.
  - Write a Python function to perform linear search on a list.
  - Test the function with different lists and target values.
  - Analyze the time complexity of the algorithm.

## Assignment 2: Binary Search

1. **Objective:** Implement and test the binary search algorithm.
2. **Tasks:**
  - Develop algorithm/flowchart/pseudo code for the above.
  - Write a Python function to perform binary search on a sorted list.
  - Test the function with different sorted lists and target values.
  - Analyze the time complexity of the algorithm.
  - Compare the performance with linear search.

### Assignment 3: Bubble Sort

1. **Objective:** Implement and test the bubble sort algorithm.
2. **Tasks:**
  - Develop algorithm/flowchart/pseudo code for the above.
  - Write a Python function to perform bubble sort on a list.
  - Test the function with different lists.
  - Analyze the time complexity of the algorithm.

#### **Assignment 4: Selection Sort**

1. **Objective:** Implement and test the selection sort algorithm.
2. **Tasks:**
  - Develop algorithm/flowchart/pseudo code for the above.
  - Write a Python function to perform selection sort on a list.
  - Test the function with different lists.
  - Analyze the time complexity of the algorithm.

### Assignment 5: Insertion Sort

1. **Objective:** Implement and test the insertion sort algorithm.
2. **Tasks:**
  - Develop algorithm/flowchart/pseudo code for the above.
  - Write a Python function to perform insertion sort on a list.
  - Test the function with different lists.
  - Analyze the time complexity of the algorithm.

### Assignment 6: Merge Sort

1. **Objective:** Implement and test the merge sort algorithm.
2. **Tasks:**
  - Develop algorithm/flowchart/pseudo code for the above.
  - Write a Python function to perform merge sort on a list.
  - Test the function with different lists.
  - Analyze the time complexity of the algorithm.
  - Compare the performance with bubble sort, selection sort, and insertion sort.

### Assignment 7: Quick Sort

1. **Objective:** Implement and test the quick sort algorithm.
2. **Tasks:**
  - Develop algorithm/flowchart/pseudo code for the above.
  - Write a Python function to perform quick sort on a list.
  - Test the function with different lists.
  - Analyze the time complexity of the algorithm.
  - Compare the performance with merge sort.