

**Symbiosis Institute of Technology**

**Faculty of Engineering**

**CSE- Academic Year 2023-24**

**Data Structures – Lab Batch 2022-26**

|  |  |
| --- | --- |
| **Lab Assignment No:- 1,2,3** | |
|  | |
| **Name of Student** | Manya Khullar |
| **PRN No.** | 22070122112 |
| **Batch** | B1 |
| **Class** | CS |
| **Academic Year & Semester** | 2022-2026  SEM 3 |
| **Date of Submission** | 29/8/23 |
|  | |
| **Title of Assignment:** | A. Implement following searching algorithm: Linear search with multiple occurrences  B. Implement following searching algorithms in menu:  1. Binary search with iteration  2. Binary search with recursion |
| **Theory:** | 1. Prepare table for following searching and sorting algorithms for their best case, average case and worst case time complexities.   Linear search, binary search, bubble sort, Insertion sort, selection sort, merge sort, quick sort.   1. Discuss on Best case and Worst case time complexities of   Linear search, binary search, bubble sort, Insertion sort, selection sort, merge sort, quick sort.  1.   |  |  |  |  | | --- | --- | --- | --- | |  | Best Case | Average Case | Worst Case | | Linear Search | 0(1) | 0(n) | 0(n) | | Binary Search | 0(1) | 0(logn) | 0(logn) | | Bubble Sort | 0(n) | 0(n^2) | 0(n^2) | | Insertion Sort | 0(n) | 0(n^2) | 0(n^2) | | Selection Sort | 0(n^2) | 0(n^2) | 0(n^2) | | Merge Sort | 0(nlogn) | 0(nlogn) | 0(nlogn) | | Quick Sort | 0(nlogn) | 0(nlogn) | 0(n^2) |   2.  Linear Search  - Best Case: O (1) - Element found at the beginning.  - Worst Case: O (n) - Element not found or at the end.  Binary Search  - Best Case: O (1) - Element in the middle.  - Worst Case: O (log n) - Element not found even after the search space is divided.  Bubble Sort  - Best Case: O(n) - Already sorted.  - Worst Case: O(n^2) - Requiring swaps for reversed order.  Insertion Sort  - Best Case: O(n) - Already sorted.  - Worst Case: O(n^2) - Shifting elements in reverse order.  Selection Sort  - Best Case: O(n^2) - Same as worst case.  - Worst Case: O(n^2) - Scanning for minimum each time.  Merge Sort  - Best Case: O (n log n) - Balanced division.  - Worst Case: O (n log n) -Uneven divisions  Quick Sort  - Best Case: O (n log n) - Balanced partitioning.  - Worst Case: O(n^2) - Highly unbalanced. |
| **Source Code/Algorithm/Flow Chart:** | A.      B. |
| **Output Screenshots (if applicable)** | A.   B. |
| **Conclusion** | Thus we have studied different sorting algorithms and their time complexities. |