Assignment no. 7

Title: Searching and Sorting

- Part 1
  1. Implement various sorting algorithms e.g., bubble sort, optimized bubble sort, selection sort, insertion sort, merge sort, quick sort and heap sort.
  2. Sample inputs of different types and sizes are added in a zip file available in the link. Sample file reading code is also given in the zip.
  3. Compare their performance on different input types (sorted, random, almost sorted)
  4. Compare their performance on different input sizes (10, 100, 1000, 10000).
  5. *Analyze the above performances (time taken, number of comparisons, no of swaps, space required, etc.) and present the results visually (e.g., tables, plot as graphs) in the lab report*. See this link for reference.
- Part 2
  1. Implement linear search using both iterative and recursive approach.
  2. Implement binary search using both iterative and recursive approach.
  3. (*Optional*)Implement interpolation search.
- Part 3
  1. Student Grade Sorting: Write a program that sorts a list of student records based on their grades. Allow users to choose whether to sort by ascending or descending grades.
  2. Online Marketplace Product Listings: Design a system for sorting and displaying products on an online marketplace. Use sorting algorithms to arrange products based on factors like popularity, price, and ratings.
  3. Top-N Elements: Implement a program that finds the top N elements from a large dataset. Use a suitable sorting algorithm to efficiently retrieve the highest elements.
  4. Phonebook Application: Create a phonebook application that allows users to add, delete, and search for contacts. Use an appropriate data structure (e.g., hash table or balanced search tree) for efficient operations.