Name: Vankar Divyesh Kumar

Roll Number : 210101108

Binary Search Visualization Software - Report

Problem Description:

The objective of this software project is to implement a binary search algorithm on a sorted list of elements. The elements can be of any similar data type (integer, float, string), and the software will use Visual Basic and C/C++ to provide a visual representation of the binary search process.

Solution Overview:

1. User Interface:

- I will design a user-friendly interface for inputting the sorted list and the target element.
- I will implement a visual display of the sorted list to enhance user interaction.
- I will make visualise the binary search algorithm steps using graphics to make the search process transparent.

2. Binary Search Algorithm:

- I will implement the binary search algorithm to efficiently locate the target element.
- I track and display the step-by-step progress of the binary search, emphasizing elements being compared.

3. Visualization:

- I will utilized Visual Basic to create a visually appealing representation of the sorted list and binary search process.
- I incorporat graphical elements such as arrows and color changes to illustrate algorithmic steps.
- I will provid a dynamic view of the search process by updating the display in real-time.

4. Input Validation:

- I will Implement robust input validation to handle invalid user inputs and unsorted lists.
- I will Display appropriate error messages to guide users towards correct inputs.
- I have to handle input validation like if user input the string or integer but as input everything is in text so I will handle this to convert them

in ASCII code or completely in Integer.(here for case of float I have to convert completely to float)

5. Testing:

- I will developed comprehensive test cases to validate the correctness and efficiency of the binary search implementation.
- Ensured the software handles various scenarios, including edge cases.

Visualization Example:

I will display the sorted list horizontally, visually highlighting elements during the binary search. This approach provides a clear representation of the algorithm's steps.

Binary Search Visualization

Array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Target: 7

Initial state:

Low: 0

High: 9

Mid: 4

Array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Is 7 == 6 (mid element)?

No, 7 > 6, so update low to mid + 1.

New state:

Low: 5

High: 9

Mid: 7

Array: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Is 7 == 7 (mid element)?

Yes! We found the target at index 7.