Binary search algorithm

The Binary Search Algorithm is a searching algorithm used to find a specific element in a sorted array or list. It follows a divide-and-conquer approach by repeatedly dividing the search space in half

how the Binary Search Algorithm works:

1. Start with defining the lower and upper bounds of the search space. Initially, the lower bound is the first element of the array, and the upper bound is the last element.
2. Calculate the middle element of the current search space by taking the average of the lower and upper bounds (middle = (lower + upper) / 2).
3. Compare the middle element with the target element that you are searching for.

- If the middle element is equal to the target element, the search is successful, and the index of the target element is found.

- If the middle element is greater than the target element, the target element, if present, must be in the lower half of the search space. Update the upper bound to be the middle element minus one and go to step 2.

- If the middle element is less than the target element, the target element, if present, must be in the upper half of the search space. Update the lower bound to be the middle element plus one and go to step 2.

4. Repeat steps 2 and 3 until the target element is found or the lower bound becomes greater than the upper bound. If the

lower bound becomes greater than the upper bound, the target element is not present in the array.

Binary search is an efficient algorithm with a time complexity of O(log n), where n is the number of elements in the array. It significantly reduces the number of comparisons needed compared to linear search algorithms, especially for large arrays.

Note that the Binary Search Algorithm requires the array to be sorted beforehand for accurate results**.**

Done By: **Shaima Riyadh**

IS