BINARY SEARCH AND LOGARITHMIC SEARCH

(BOTH SHARE A TIME COMPLEXITY OF O(logn) – WHY?

is a [search algorithm](https://en.wikipedia.org/wiki/Search_algorithm) that finds the position of a target value within a [sorted array](https://en.wikipedia.org/wiki/Sorted_array). Binary search compares the target value to the middle element of the array. If they are not equal, the half in which the target cannot lie is eliminated and the search continues on the remaining half, again taking the middle element to compare to the target value, and repeating this until the target value is found. If the search ends with the remaining half being empty, the target is not in the array.

Binary search runs in logarithmic time in the worst case, making O(log n) comparisons, where n is the number of elements in the array. Binary search is faster than linear search except for small arrays. However, the array must be sorted first to be able to apply binary search.

Binary search works on sorted arrays. Binary search begins by comparing an element in the middle of the array with the target value. If the target value matches the element, its position in the array is returned. If the target value is less than the element, the search continues in the lower half of the array. If the target value is greater than the element, the search continues in the upper half of the array. By doing this, the algorithm eliminates the half in which the target value cannot lie in each iteration

**Step-by-step Binary Search Algorithm:** We basically ignore half of the elements just after one comparison.

1. Compare x with the middle element.
2. If x matches with the middle element, we return the mid index.
3. Else If x is greater than the mid element, then x can only lie in the right half subarray after the mid element. So we recur for the right half.
4. Else (x is smaller) recur for the left half.

Binary search works on sorted arrays. Binary search **begins by comparing an element in the middle of the array with the target value**. If the target value matches the element, its position in the array is returned. If the target value is less than the element, the search continues in the lower half of the array.

Binary search is a very efficient and fast algorithm **to find an element inside a sorted list of elements**, this algorithm works based on the principle of divide and conquer.

Binary search is a fast search algorithm with run-time complexity of Ο(log n). This search algorithm works on the principle of divide and conquer. For this algorithm to work properly, the data collection should be in the sorted form.

Binary search looks for a particular item by comparing the middle most item of the collection. If a match occurs, then the index of item is returned. If the middle item is greater than the item, then the item is searched in the sub-array to the left of the middle item. Otherwise, the item is searched for in the sub-array to the right of the middle item.

Binary Search is also logarithmic so they share the same time complexity.