

Lecture - 4

CSE-604
Data Structure Lab (Part-1)

Experiment 3: Binary Search Algorithm

What is Binary Search?

A Binary Search is a sorting algorithm, that is used to search an element in a **sorted array**. A binary search technique works only on a sorted array, so an array must be sorted to apply binary search on the array. It is a searching technique that is better than the linear search technique as the number of iterations decreases in the binary search.

Binary Search Algorithm can be implemented in the following two ways:

- Iterative Method
- Recursive Method

Working of Binary search

1. Binary search follows the **divide and conquer approach** in which the list is divided into **two halves**, and the item is compared with the **middle element** of the list. If the match is found then, the location of the middle element is returned. Otherwise, we search into either of the halves depending upon the result produced through the match.

The basic steps to perform Binary Search are:

- Begin with the mid element of the whole array as a search key.
- If the value of the search key is equal to the item then return an index of the search key.
- Or if the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half.
- Otherwise, narrow it to the upper half.
- Repeatedly check from the second point until the value is found or the interval is empty.

Example

The general steps for both methods are discussed below:

1. Let's assume the array in which the searching is to be performed:

Initial array

0	1	2	3	4	5	6	7	8
10	12	24	29	39	40	51	56	69

**Applicable Cases of
Search in each iteration:**

1. Data = Mid Value
2. Data < Mid Value
3. Data > Mid Value

Let the element to search is, K = 56

We have to use the below formula to calculate the mid of the array -

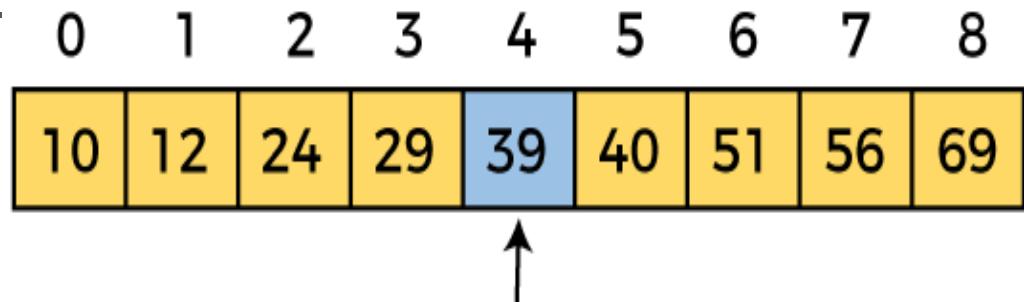
$$\text{mid} = (\text{beg} + \text{end})/2$$

So, in the given array -

$$\text{beg} = 0$$

$$\text{end} = 8$$

$$\text{mid} = (0 + 8)/2 = 4. \text{ So, } 4 \text{ is the mid of the array.}$$



$$A[\text{mid}] = 39$$

$$A[\text{mid}] < K \text{ (or, } 39 < 56)$$

$$\text{So, beg} = \text{mid} + 1 = 5, \text{end} = 8$$

$$\text{Now, mid} = (\text{beg} + \text{end})/2 = 13/2 = 6$$

0 1 2 3 4 5 6 7 8

10	12	24	29	39	40	51	56	69
----	----	----	----	----	----	----	----	----



$$A[\text{mid}] = 51$$

$$A[\text{mid}] < K \text{ (or, } 51 < 56\text{)}$$

$$\text{So, beg} = \text{mid} + 1 = 7, \text{end} = 8$$

$$\text{Now, mid} = (\text{beg} + \text{end})/2 = 15/2 = 7$$

0 1 2 3 4 5 6 7 8

10	12	24	29	39	40	51	56	69
----	----	----	----	----	----	----	----	----



$A[\text{mid}] = 56$

$A[\text{mid}] = K$ (or, $56 = 56$)

So, location = mid

Element found at 7th location of the array

The element is found. Now the algorithm will return the index of the value.

Time Complexity

Best Case: $O(1)$

Average Case: $O(\log n)$

Worst Case: $O(\log n)$

Space Complexity: $O(1)$

Advantages of Binary Search Algorithm

- Since it follows the technique to eliminate half of the array elements, it is more efficient as compared to linear search for large data.
- Better time complexity and thus takes less compilation time.
- An improvement over linear search as it breaks the array down in half rather than sequentially traversing through the array elements.

Limitations of Binary Search Algorithm

- Binary Search algorithm could only be implemented over a sorted array.
- Small unsorted arrays would take considerable time in sorting and then searching the desired element. So, binary search is not preferred in such cases.
- It has poor locality of reference compared to linear search algorithm when comes to in-memory searching for short intervals.

Applications of Binary Search

- This algorithm is used to search element in a given sorted array with more efficiency.
- It could also be used for few other additional operations like- to find the smallest element in the array or to find the largest element in the array.

What is Iteration Algorithm?

In the case of **Iterative algorithms**, a certain set of statements are repeated a certain number of time. An Iterative algorithm will use looping statements such as for loop, while loop or do-while loop to repeat the same steps number of time.

What is Recursive Algorithm?

Recursive algorithm, **a function calls itself again and again till the base condition(stopping condition) is satisfied.**