

## Binary Search Algorithm

Binary Search Algorithm is a searching algorithm used in a sorted array by repeatedly dividing the search interval in half. The idea of binary search is to use the information that the array is sorted and reduce the time complexity to  $O(\log N)$ .

### What is Binary Search Algorithm?

**Binary search** is a search algorithm used to find the position of a target value within a **sorted** array. It works by repeatedly dividing the search interval in half until the target value is found or the interval is empty. The search interval is halved by comparing the target element with the middle value of the search space.

### Conditions to apply Binary Search Algorithm in a Data Structure:

To apply Binary Search algorithm:

- The data structure must be sorted.
- Access to any element of the data structure should take constant time.

### Binary Search Algorithm:

Below is the step-by-step algorithm for Binary Search:

- Divide the search space into two halves by **finding the middle index “mid”**.
- Compare the middle element of the search space with the **key**.
- If the **key** is found at middle element, the process is terminated.
- If the **key** is not found at middle element, choose which half will be used as the next search space.
  - If the **key** is smaller than the middle element, then the **left** side is used for next search.
  - If the **key** is larger than the middle element, then the **right** side is used for next search.

- This process is continued until the **key** is found or the total search space is exhausted.

### How to Implement Binary Search Algorithm?

The **Binary Search Algorithm** can be implemented in the following two ways

- Iterative Binary Search Algorithm
- Recursive Binary Search Algorithm

