

Binary Search

Binary Search is a searching algorithm for finding an element's position in a **sorted array**.

In this approach, the element is always searched in the middle of a portion of an array.

Binary search can be implemented only on a sorted list of items. If the elements are not sorted already, we need to sort them first.

Binary Search Working

Binary Search Algorithm can be implemented in two ways which are discussed below.

1. Iterative Method
2. Recursive Method

The recursive method follows [the divide and conquer](#) approach.

The general steps for both methods are discussed below.

1. The array in which searching is to be performed is:

3	4	5	6	7	8	9
---	---	---	---	---	---	---

Initial array

Let $x = 4$ be the element to be searched.

2. Set two pointers **low** and **high** at the lowest and the highest positions respectively.



Setting pointers

3. Find the middle element **mid** of the array
ie. $\text{arr}[(\text{low} + \text{high})/2] = 6$.



Mid element

4. If $x == \text{mid}$, then return mid. Else, compare the element to be searched with m.
5. If $x > \text{mid}$, compare x with the middle element of the elements on the right side of **mid**. This is done by setting **low** to $\text{low} = \text{mid} + 1$.

6. Else, compare x with the middle element of the elements on the left side of mid . This is done by setting $high$ to $high = mid - 1$.

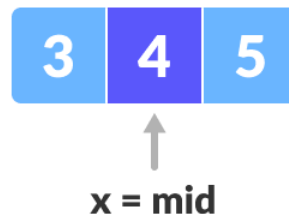


Finding mid element

7. Repeat steps 3 to 6 until low meets high.



Mid element



8. $x = 4$ is found.

Binary Search Algorithm

Iteration Method

do until the pointers low and high meet each other.

```
mid = (low + high)/2
```

```
if (x == arr[mid])
```

```
    return mid
```

```
else if (x > arr[mid]) // x is on the right side
```

```
    low = mid + 1
```

```
else // x is on the left side
```

```
    high = mid - 1
```

Recursive Method

```
binarySearch(arr, x, low, high)
```

```
    if low > high
```

```
        return False
```

```
    else
```

```
        mid = (low + high) / 2
```

```
        if x == arr[mid]
```

```
            return mid
```

```
        else if x > arr[mid] // x is on the right side
```

```
            return binarySearch(arr, x, mid + 1, high)
```

```
        else // x is on the left side
```

```
            return binarySearch(arr, x, low, mid)
```