Binary Search

Binary search is a searching technique that is based upon the Divide-and-Conquer Rule.

In this searching technique, a sorted array is divided into two equal halves and then the same technique is applied onto the two halves searching for the element by comparing the high and the low.

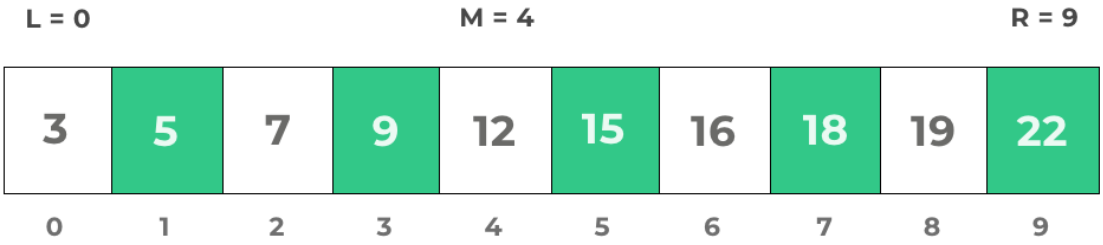
Time Complexity (Best)	Ω(1)
Time Complexity (Avg)	Θ(log n)
Time Complexity (Worst)	O(log n)
Space Complexity	O(1)

Working of binary search in JAVA

1. For Binary Search to be performed on any array, the array must be already sorted in any format, that is, either ascending or descending.
2. Find the middle index of the array/list.
3. If the middle element is equal to the search element, Stop Searching.
4. If the element that is to be searched is less then the middle element then consider the first half as a separate list.
5. Else-If the element that is to be searched is larger then the middle element then consider the second half as a separate list.
6. Repeat Step 2-3-4-5 Until desired result is found.

Binary Search in Java

$$M = \frac{(L + R)}{2} \quad \text{or} \quad M = L + \frac{(R - L)}{2}$$

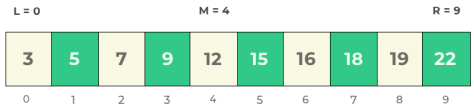


Algorithm of Binary Search in Java

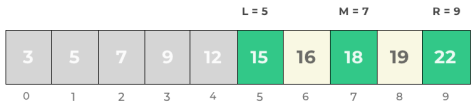
- while(left<=right) mid=left + (right – left)/2;
- if(a[mid]<search_element) left=mid+1;
- else if(a[mid]>search_element) right=mid-1;
- If found return index+1
- Else return -1

```
int binarySearch(int array[], int left, int right, int item){

  if (right >= left){

    // calculation of new mid
    int mid = left + (right - left)/2;

    // returns position where found
    if (array[mid] == item)
      return mid+1;

    // goes to recursive calls in left half
    if (array[mid] > item)
      return binarySearch(array, left, mid-1, item);

    // goes to recursive calls in right half
    else
      return binarySearch(array, mid+1, right, item);
  }
  // if element is not found we return -1
  else
    return -1;
}
```

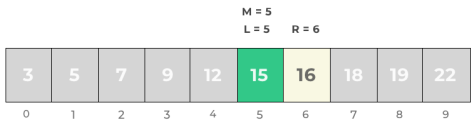
Search: 16



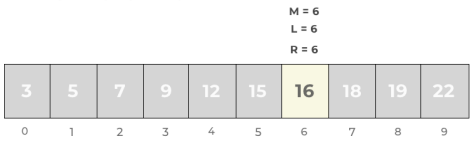
- array[mid] < item i.e. 12 < 16
- binarySearch(array, mid+1, right, item)
- binarySearch(array, 5, 9, 16)



- array[mid] > item i.e. 18 > 16
- binarySearch(array, left, mid-1, item)
- binarySearch(array, 5, 6, 16)



- array[mid] < item i.e. 15 > 16
- binarySearch(array, mid+1, right, item)
- binarySearch(array, 6, 6, 16)



- array[mid] == item
- Return mid + 1