Binary Search In JavaScript

Binary Search is searching tec' uer approach. It used to search any element in a sorted array.

As compared to linear, binary search is much faster with Time Complexity of O(logN) whereas linear search algorithm works in O(N) time complexity.

In this article, implement of Binary Search in Javascript using both iterative and recursive ways are discussed.

Given a sorted array of numbers. The task is to search a given element x in the array using Binary search.

Examples:

```
Input : arr[] = {1, 3, 5, 7, 8, 9}
        x = 5
Output : Element found!

Input : arr[] = {1, 3, 5, 7, 8, 9}
        x = 6
Output : Element not found!
```

Note: Assuming the array is sorted.

Recommended: Please try your approach on *{IDE}* first, before moving on to the solution.

Recursive Approach :

4. If greater, call the same function with ending index = middle-1 and repeat step 1.
5. If smaller, call the same function with starting index = middle+1 and repeat step 1.

Below is the implementation of Binary Search(Recursive Approach) in JavaScript:

```
<script>
let recursiveFunction = function (arr, x, start, end) {

    // Base Condition
    if (start > end) return false;

    // Find the middle index
    let mid=Math.floor((start + end)/2);

    // Compare mid with given key x
    if (arr[mid]===x) return true;

    // If element at mid is greater than x,
    // search in the left half of mid
    if(arr[mid] > x)
        return recursiveFunction(arr, x, start, mid-1);
    else

        // If element at mid is smaller than x,
        // search in the right half of mid
        return recursiveFunction(arr, x, mid+1, end);
}

// Driver code
let arr = [1, 3, 5, 7, 8, 9];
let x = 5;

if (recursiveFunction(arr, x, 0, arr.length-1))
    document.write("Element found!<br>");
else document.write("Element not found!<br>");

x = 6;

if (recursiveFunction(arr, x, 0, arr.length-1))
    document.write("Element found!<br>");
else document.write("Element not found!<br>");
</script>
```

Output:

```
Element found!
Element not found!
```

Time Complexity: O(logN).

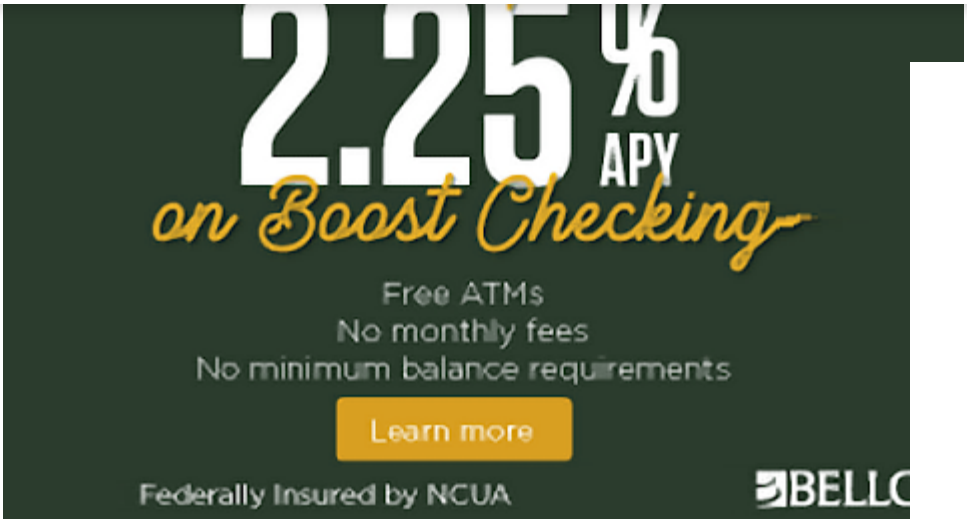
Below is the implementation of Binary Search(Iterative Approach) in JavaScript:

```
<script>
// Iterative function to implement Binary Search
let iterativeFunction = function (arr, x) {

    let start=0, end=arr.length-1;

    // Iterate while start not meets end
    while (start<=end){

        // Find the mid index
        let mid=Math.floor((start + end)/2);

        // If element is present at mid, return True
        if (arr[mid]===x) return true;

        // Else look in left or right half accordingly
        else if (arr[mid] < x)
             start = mid + 1;
        else
             end = mid - 1;
    }

    return false;
}

// Driver code
let arr = [1, 3, 5, 7, 8, 9];
let x = 5;

if (iterativeFunction(arr, x, 0, arr.length-1))
    document.write("Element found!<br>");
else document.write("Element not found!<br>");

x = 6;

if (iterativeFunction(arr, x, 0, arr.length-1))
    document.write("Element found!<br>");
else document.write("Element not found!<br>");
</script>
```

Output:

```
Element found!
Element not found!
```

Time Complexity: O(logN).



Recommended Posts:

Implementation of Binary Search Tree in Javascript

Meta Binary Search | One-Sided Binary Search

Why is Binary Search preferred over Ternary Search?

Linear Search vs Binary Search

Interpolation search vs Binary search

Search Bar using HTML, CSS and JavaScript

Case insensitive search in JavaScript

JavaScript | string.search() Method

Binary Search

Binary Search in PHP

Binary Search using pthread

Binary Search a String

Variants of Binary Search

The Ubiquitous Binary Search | Set 1

Binary Search in Java

imdhruvgupta
Check out this Author's contributed articles.

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article

Improved By : nidhi_biet

Article Tags : Algorithms Divide and Conquer JavaScript Searching Technical Scripter Binary Search Technical Scripter 2018

Practice Tags : Searching Divide and Conquer Binary Search Algorithms

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org