Binary Search

Binary search requires sorted array.

## Flowchart:

Initialize array, key, start = 0; end = arr.length, mid = (start + end)/2

If loop doesn’t return any value means that key value not present in the array.

Return false

Check if the arr[mid] === key

Then the element is present and return true.

And break the loop

Check if arr[mid]< key

Search the element in right part of array.

Shift the start variable towards

Mid + 1 value.

start = mid+1;

if conditions

Else and else if conditions

Check if arr[mid]> key

Search the element in left part of array.

Shift the end variable towards

Mid – 1 value.

End = mid-1;

Loop Condition

while start ≤ end

## Algorithm:

Step 1: Initialize array, key, start, end mid;

Step 2: Start while loop with condition statement( while (start ≤ end))

Step 3: Check the condition if (arr[mid] === key)

Step 4: Return true if the condition is true.

Step 5: Check the second condition else if (arr[mid] > key), it means that we are on the right side of the array so we have to shift the end value to towards left.

It also means that we have to search in the left part of the array.

Else if (arr[mid] > key ) {

End = mid -1;

}

Step 6: Go to step 2.

Step 7: Check the third condition if(arr[mid] < key), it means that we are on the left side of the array so we have to shift start value toward the right.

It also means that we have to search in the right part of the array.

We ca also write the else condition because we have already checked that arr[mid] value equal to key and arr[mid] value greater than key, so now there is two conditions left if it is arr[mid] less than key or it is not present in the array.

Step 8: Go to step 2.

Step 9: If the loop completes means that key value not present in the array so it we have to return false.