33. Search in Rotated Sorted Array

There is an integer array nums sorted in ascending order (with distinct values).

Prior to being passed to your function, nums is possibly rotated at an unknown pivot index k (1 <= k < nums.length) such that the resulting array is [nums[k], nums[k+1], ..., nums[n-1], nums[0], nums[1], ..., nums[k-1]] (0-indexed). For example, [0,1,2,4,5,6,7] might be rotated at pivot index 3 and become [4,5,6,7,0,1,2].

Given the array nums after the possible rotation and an integer target, return the index of target if it is in nums, or -1 if it is not in nums.

You must write an algorithm with O(log n) runtime complexity.

Input: nums = [4,5,6,7,0,1,2], target = 0 Output: 4

```
var search = function (nums, target) {
       let N = nums.length;
       let left = 0;
       let right = N - 1;
       let res = -1;

       while (left <= right) {
           let mid = parseInt(left + (right - left) / 2);

           if (nums[mid] === target) {
               res = mid;
               break;
           }

           if (nums[left] <= nums[mid]) { // left <= right
               if (target < nums[mid] && target >= nums[left]) {
                   right = mid - 1;
               } else {
                   left = mid + 1;
               }
           } else {
               if (target > nums[mid] && target <= nums[right]) {
                   left = mid + 1;
```

```
            } else {
                right = mid - 1;
            }

        }
    }
    return res;
};
/*
Search in Rotated Sorted Array II Medium
Find Minimum in Rotated Sorted Array Medium
Pour Water Between Buckets to Make Water Levels Equal Medium
*/
```