1

```
public class Solution {
    public int search(int[] nums, int target) {
        if (nums.length == 0) return -1;
        int l = 0, r = nums.length-1;

        while (l < r) {
            int mid = (l + r) / 2;
            if (nums[mid] == target) return mid;
            else if (nums[mid] < target) {
               if (target <= nums[r] || nums[mid] > nums[r]) {
                   l = mid+1;
               } else {
                   r = mid-1;
               }
            } else {
                if (target >= nums[l] || nums[mid] < nums[r]) {
                   r = mid-1;
               } else {
                   l = mid+1;
               }
            }
        }
        return nums[l] == target ? l : -1;
    }
}
```

1:

```
class Solution {
    public int search(int[] nums, int target) {
        if (nums == null || nums.length == 0)
            return -1;
        int l = 0, r = nums.length - 1;
        while (l + 1 < r) {
            int mid = (r - l) / 2 + l;
            if (nums[mid] == target) {
                return mid;
            } else if (nums[mid] > nums[r]) {
                if (target >= nums[l] && target < nums[mid]) {
                    r = mid;
                } else {
                    l = mid;
                }
            } else if (nums[mid] < nums[r]) {
                if (target > nums[mid] && nums[r] >= target) {
                    l = mid;
                } else {
                    r = mid;
                }
            }
        }
        return nums[l] == target ? l : nums[r] == target ? r : -1;
    }
}
```

2: duplicates allowed

```
class Solution {
    public boolean search(int[] nums, int target) {
        if (nums.length == 0)
            return false;
        int l = 0, r = nums.length - 1;

        while (l + 1 < r) {
            int mid = (r - l) / 2 + l;
            if (nums[mid] == target) {
                return true;
            } else if (nums[mid] == nums[l]) {
                l++;
            } else if (nums[mid] == nums[r]) {
                r--;
            } else if (nums[mid] > nums[r]) {
                if (target >= nums[l] && target < nums[mid]) {
                    r = mid;
                } else {
                    l = mid;
                }
            } else if (nums[mid] < nums[r]) {
                if (target > nums[mid] && nums[r] >= target) {
                    l = mid;
                } else {
                    r = mid;
                }
            }
        }
        return nums[l] == target || nums[r] == target;
    }
}
```