# [搜索旋转排序数组](https://leetcode-cn.com/explore/interview/card/bytedance/243/array-and-sorting/1017/)

**头条重点**

## 题目

假设按照升序排序的数组在预先未知的某个点上进行了旋转。( 例如，数组 [0,1,2,4,5,6,7] 可能变为 [4,5,6,7,0,1,2] )。

搜索一个给定的目标值，如果数组中存在这个目标值，则返回它的索引，否则返回 -1 。

你可以假设数组中不存在重复的元素。

你的算法时间复杂度必须是 O(log n) 级别。

示例 1:

输入: nums = [4,5,6,7,0,1,2], target = 0
输出: 4
示例 2:

输入: nums = [4,5,6,7,0,1,2], target = 3
输出: -1

## 解题思路

1. 旋转数组是分为两段有序，主要得注意 mid 落在哪个段上

public static int search(int[] nums, int target) {
 int start = 0, end = nums.length - 1;

 while (start <= end) {
 int mid = (start + end) / 2;
 if (nums[mid]==target) return mid;

 if (nums[mid] >= nums[start]) {
 if (target < nums[mid] && target >= nums[start]) {
 end = mid - 1;
 } else {
 start = mid + 1;
 }

 }

 if (nums[mid] <= nums[end]) {
 if (target > nums[mid] && target <= nums[end]) {
 start = mid + 1;
 } else {
 end = mid - 1;
 }
 }
 }

 return -1;
}