

33. Search in Rotated Sorted Array

题目描述: <https://leetcode.com/problems/search-in-rotated-sorted-array/>

给定一个经过rotate的按序排列的数组，例如[7,8,9,10,5,6]

在这个数组中找到指定的数，然后返回下标，如果不存在则返回-1

解题思路:

1. 首先找到转折点，然后在进行binary search的时候加入pivot，从而实现按序的binary search
2. 找到转折点，看是在左半边还是右半边，然后binary search。

代码1:

```

class Solution {
public:
    int search(vector<int>& nums, int target) {
        int len = nums.size();
        if(len == 0) return -1;
        int pivot = 0, l = 0, r = len - 1;
        int mid = 0;
        while(l < r) {
            mid = l + (r - l) / 2;
            if(nums[l] < nums[r]) {
                break;
            }
            if(nums[mid] > nums[r]) {
                l = mid + 1;
            }
            else {
                r = mid;
            }
        }
        pivot = l;
        if(nums[0] <= target && pivot!=0) {
            l = 0; r = pivot - 1;
        }
        else {
            l = pivot; r = len - 1;
        }
        while(l <= r) {
            mid = l + (r - l) / 2 ;
            if(nums[mid] == target) {
                return mid;
            }
            else if(nums[mid] < target) {
                l = mid + 1;
            }
            else {
                r = mid - 1;
            }
        }
        return -1;
    }
};

```

代码2:

```

class Solution {
public:
    int search(vector<int>& nums, int target) {
        int len = nums.size();
        if(len == 0) return -1;
        int pivot = 0, l = 0, r = len - 1;
        int mid = 0;
        while(l < r) {
            mid = l + (r - l) / 2;
            if(nums[l] < nums[r]) {
                break;
            }
            if(nums[mid] > nums[r]) {
                l = mid + 1;
            }
            else {
                r = mid;
            }
        }
        pivot = l;
        if(nums[0] <= target && pivot!=0) {
            l = 0; r = pivot - 1;
        }
        else {
            l = pivot; r = len - 1;
        }
        while(l <= r) {
            mid = l + (r - l) / 2 ;
            if(nums[mid] == target) {
                return mid;
            }
            else if(nums[mid] < target) {
                l = mid + 1;
            }
            else {
                r = mid - 1;
            }
        }
        return -1;
    }
};

```