

81. Search in Rotated Sorted Array II

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

给定一个按序排列的数组，在中间某位置处两边颠倒，给定某个元素，判断在这个数组中是否存在。
例如：

```
[4,5,6,7,1,2,3]
```

给定target = 3

```
return true;
```

解题思路：

判断是在前半边还是后半边，然后再查找。

代码：

```
class Solution {
public:
    bool search(vector<int>& nums, int target) {
        if(nums.size() == 0)
            return false;
        if(target < nums[0]) {
            for(int i = nums.size() - 1; i >= 0; i--) {
                if(target == nums[i])
                    return true;
                else if(target > nums[i])
                    return false;
            }
        }
        else if(target > nums[0]) {
            for(int i = 0; i < nums.size(); i++) {
                if(target == nums[i])
                    return true;
                else if(target < nums[i])
                    return false;
            }
        }
        else
            return true;
        return false;
    }
};
```