

一、题目说明

这个题目是33. Search in Rotated Sorted Array，说的是在一个“扭转”的有序列表中，查找一个元素，时间复杂度 $O(\log n)$ 。

二、我的解答

这是一个查找，根据复杂度，我们知道只能用二分查找。但由于这个不是一个完全的有序列表，故需要改造。先写出二分查找（来源于数据结构）：

二分查找：

```
int search(vector<int>& nums, int target){
    if(nums.size()<0) return -1;
    int start = 0, end=nums.size()-1;
    int mid = -1;
    while(start <= end){
        //mid = (start + end) /2;
        mid = start + (end - start) /2;
        if(target == nums[mid]){
            return mid;
        }else if(target < nums[mid]){
            end = mid -1;
        }else{
            start = mid + 1;
        }
    }
    return -1;
}
```

然后在二分查找基础上进行改进：

```
class Solution{
public:
    int search(vector<int>& nums, int target){
        if(nums.size()<0) return -1;
        int start = 0, end=nums.size()-1;
        int mid = -1;
        while(start<=end){
            mid = start + (end - start) / 2;
            if(nums[mid] == target) return mid;

            //前半部分有序
            if(nums[start]<=nums[mid]){
                //target在前半部分
                if(target >=nums[start] && target<nums[mid]){
                    end = mid -1;
                }else{
                    start = mid + 1;
                }
            }else{
                //后半部分有序
                if(target<=nums[end] && target>nums[mid]){
                    start = mid + 1;
                }
            }
        }
    }
}
```

```
        }else{
            end = mid -1;
        }
    }
    return -1;
}
};
```

三、优化措施

无