

在排序数组中查找数字 I

统计一个数字在排序数组中出现的次数。

示例 1:

输入: nums = [5,7,7,8,8,10], target = 8

输出: 2

示例 2:

输入: nums = [5,7,7,8,8,10], target = 6

输出: 0

```
class Solution {
public:
    int search(vector<int>& nums, int target) {
        if(nums.empty())
            return 0;
        if(nums.size()==1&&nums[0]==target)
            return 1;
        int low=0;
        int high=nums.size()-1;
        int cout=0;
        int mid;
        while(low<=high)//仔细推理以及测试发现 此处得为 low<=high 比如输入数组
        大小为 2 且 target 为第二个数 需要考虑 low=high 否则无法进入 while 循环
        {
            mid=(low+high)/2;
            if(target<nums[mid])
            {
                high=mid-1;
            }
            else
            {
                if(target==nums[mid])
                {
                    cout++;
                    break;//需要跳出循环考虑从中间向两边查找
                }
                else
                {
                    low=mid+1;
                }
            }
        }
    }
};
```

```
    }  
  }  
  if(nums[mid]==target)  
  {  
    low=high=mid;  
    while(--low>=0&&nums[low]==target)  
    {  
      cout++;  
    }  
    while(++high<nums.size()&&nums[high]==target)  
    {  
      cout++;  
    }  
  }  
  return cout;  
}  
};
```