

给定一个整数类型的数组 `nums`，请编写一个能够返回数组“**中心索引**”的方法。

我们是这样定义数组**中心索引**的：数组中心索引的左侧所有元素相加的和等于右侧所有元素相加的和。

如果数组不存在中心索引，那么我们应该返回 -1。如果数组有多个中心索引，那么我们应该返回最靠近左边的那一个。

示例 1:

输入:

```
nums = [1, 7, 3, 6, 5, 6]
```

输出: 3

解释:

索引3 (`nums[3] = 6`) 的左侧数之和 ($1 + 7 + 3 = 11$)，与右侧数之和 ($5 + 6 = 11$) 相等。

同时，3 也是第一个符合要求的中心索引。

示例 2:

输入:

```
nums = [1, 2, 3]
```

输出: -1

解释:

数组中不存在满足此条件的中心索引。

说明:

- `nums` 的长度范围为 `[0, 10000]`。
- 任何一个 `nums[i]` 将会是一个范围在 `[-1000, 1000]` 的整数。

思路，用数组`nums`记录动态记录前 $1 \sim i$ 的和与数组的总和，用相减的形式判断两边值，需注意找最靠近左边的索引，并且0位和 $n-1$ 位左右是为0，需要提前和延后判断。

```
class Solution {
public:
    int pivotIndex(vector<int>& nums) {
        int sum;
        int n=nums.size();
        if(n==0){
            return -1;
        }
        sum=nums[0];
        for(int i=1;i<n;i++){
```

```
        sum+=nums[i];
        nums[i]+=nums[i-1];
    }
    if(sum-nums[0]==0){
        return 0;
    }
    for(int i=1;i<n-1;i++){
        if(sum-nums[i]==nums[i-1]){
            return i;
        }
    }
    if(nums[n-2]==0){
        return n-1;
    }
    return -1;
}

};
```