

寻找数组的中心索引

2020年12月23日 21:02

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。

解析:

分析提供得知，如果是一个非空数组，那么可以将数组分为三个部分 左侧元素之和(`LeftSum`)、中间值 (`MidValue`)、右侧元素之和(`RightSum`)，那么由中间索引的概念可以得出下面公式：

$$\text{ArraySum} = \text{LeftSum} + \text{MidValue} + \text{RightSum} \quad (\text{ArraySum为整个数组的元素总和})$$

又因为 `LeftSum == RightSum`，因此可以变化为下面公式：

$$\begin{aligned}\text{ArraySum} &= \text{LeftSum} * 2 + \text{MidValue} \\ \text{LeftSum} * 2 &= \text{ArraySum} - \text{MidValue} \\ \text{LeftSum} &= (\text{ArraySum} - \text{MidValue} - \text{LeftSum})\end{aligned}$$

得出的这个关系便可以进行程序的编写

C语言:

```
int pivotIndex(int* nums, int numsSize){
    if (numsSize <= 0)
        return -1;
    int i = 0;
    int sum = 0;
    int temp_sum = 0;
    int temp_sum_2 = 0;
    for (i = 0; i < numsSize; i++)
        sum += nums[i];
    temp_sum = 0;
```

```
for (i = 0; i < numsSize; i++)
{
    // if (temp_sum * 2 == (sum - nums[i]))
    if (temp_sum == (sum - nums[i] - temp_sum))
    {
        return i;
    }
    temp_sum += nums[i];
}
return -1;
}
```