

题目是这样的

## 4. 寻找两个正序数组的中位数

已解答 

困难 相关标签 相关企业 Ax

给定两个大小分别为  $m$  和  $n$  的正序（从小到大）数组 `nums1` 和 `nums2`。请你找出并返回这两个正序数组的 **中位数**。

算法的时间复杂度应该为  $O(\log(m+n))$ 。

### 示例 1:

输入: `nums1 = [1,3]`, `nums2 = [2]`

输出: `2.00000`

解释: 合并数组 = `[1,2,3]` , 中位数 `2`

### 示例 2:

输入: `nums1 = [1,2]`, `nums2 = [3,4]`

输出: `2.50000`

解释: 合并数组 = `[1,2,3,4]` , 中位数  $(2 + 3) / 2 = 2.5$

- 1, 要求写一个能寻找两个正序数组的中位数的函数
- 2, 两个数组都是从小到大的顺序, 是要找到中位数, 我们只要先把两个数组合并起来, 需要合并后依然是从小到大的顺序
- 3, 然后求出中间的下标, 由于合并后数组的元素个数偶数和奇数都有可能, 所以后面要判断数组元素个数是偶数还是奇数
- 4, 如果是奇数, 那元素个数除以 2 再取整直接就是中位数的下标了, 因为数组索引是从 0 开始的
- 5, 如果是偶数, 元素个数除以 2 是最中间两个数的第二个, 只需要再减一个 1 得到最中间第二个数, 两数相加除以 2 得到中位数 (不能取整), 下面为示例  
用 4 个元素的数组代表元素个数为偶数的数组  
用 3 个元素的数组代表元素个数为奇数的数组  
可以推出中位数在所有偶数个元素和奇数个元素数组中的位置

```
数组索引: 0  1  2  3
数组元素: 11 12 13 14
元素个数: 4
           4/2==2 , 得到第二中位数 13 下标
           2-1==1 , 得到第一中位数 12 下标
数组索引: 0  1  2
数组元素: 11 12 13
元素个数: 3
           3/2==1 , 得到中位数 12 下标
```

5. 接下来写函数代码

```

double findMedianSortedArrays(int* nums1, int nums1Size, int* nums2, int nums2Size) {
    int i,x,y;
    x=y=0;
    int nums3[nums1Size+nums2Size];//创建一个可以容纳 nums1 和 nums2 所以元素的数组
    for(i=0;i<nums1Size+nums2Size;)//实现合并后依然保持正序
    {
        if(x<nums1Size&& y<nums2Size)
        if(nums1[x]<nums2[y])
        {
            nums3[i++]=nums1[x++];
        }
        else
        {
            nums3[i++]=nums2[y++];
        }
        if(x>=nums1Size&& y<nums2Size)//num1 的元素合并完， num2 没有， 直接赋值
        {
            while(y<nums2Size)
            {
                nums3[i++]=nums2[y++];
            }
            break;
        }
        if(y>=nums2Size&& x<nums1Size) //num2 的元素合并完， num1 没有， 直接赋值
        {
            while(x<nums1Size)
            {
                nums3[i++]=nums1[x++];
            }
            break;
        }
    }
    double result;
    if(i%2==1)//判断元素个数是偶数还是奇数， 以两种方式得到 result 结果
    {
        result=nums3[i/2];
    }
    else
    {
        result=(nums3[i/2]+nums3[i/2-1])/2.0;
    }
    return result;
}

```