# 算法

#### 1.两数之和

给定一个整数数组 nums 和一个整数目标值 target，请你在该数组中找出 和为目标值 的那 两个 整数，并返回它们的数组下标。

你可以假设每种输入只会对应一个答案。但是，数组中同一个元素在答案里不能重复出现。

你可以按任意顺序返回答案。

var twoSum = function(nums, target) {

    const \_length = nums.length;

    const \_mayMap = new Map();

    for (let i = 0; i < \_length; i++) {

        if (\_mayMap.has(target - nums[i])) {

        return [\_mayMap.get(target - nums[i]), i];

        }

    \_mayMap.set(nums[i], i);

    }

};

#### 2.无重复字符串的最长子串

给定一个字符串，请你找出其中不含有重复字符的最长子串的长度

输入 s=’abcc’

输出3 因为无重复字符的最长子串是abc，长度为3

function lengthOfLongestSubstring(s){

let resLen = 0;

let begin = 0, end = 1;

while(end <= s.length){

let index = s.slice(begin, end-1).indexOf(s[end - 1]);

if(index == -1){

end - begin > resLen ? resLen = end - begin : null;

end ++;

}else{

begin ++;

}

}

return resLen;

}

#### 3.寻找两个正序数组的中位数

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

var findMedianSortedArrays = function(nums1, nums2) {

    let arr = [...nums1, ...nums2].sort((x, y)=>x-y)

    let Isodd = arr.length % 2 == 0 ? false : true;

    if(Isodd){

        return arr[parseInt(arr.length/2)]

    }else{

        return (arr[arr.length/2] +arr[arr.length/2-1]) /2

    }

};