4. 两数之和

地址：两数之和

题目：

- English：

1. Two Sum

难度 简单 11511

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to* `target`.

You may assume that each input would have ***exactly* one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

Example 1:

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Output: Because nums[0] + nums[1] == 9, we return [0, 1].
```

Example 2:

```
Input: nums = [3,2,4], target = 6
Output: [1,2]
```

Example 3:

```
Input: nums = [3,3], target = 6
Output: [0,1]
```

Constraints:

- $2 <= nums.length <= 10^4$
- $-10^9 <= nums[i] <= 10^9$
- $-10^9 <= target <= 10^9$
- **Only one valid answer exists.**

- 中文：

1. 两数之和

难度 简单 11511

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** *`target`* 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。

示例 1：

```
输入：nums = [2,7,11,15], target = 9
输出：[0,1]
解释：因为 nums[0] + nums[1] == 9 ，返回 [0, 1] 。
```

示例 2：

```
输入：nums = [3,2,4], target = 6
输出：[1,2]
```

示例 3：

```
输入：nums = [3,3], target = 6
输出：[0,1]
```

提示：

- $2 <= nums.length <= 10^4$
- $-10^9 <= nums[i] <= 10^9$
- $-10^9 <= target <= 10^9$
- **只会存在一个有效答案**

思路1：暴力枚举，两层循环

分析：

> ⭐ 两层循环，枚举不同下标,将前一个数与后一个数相加，和等于目标数，返回这两个数值。

代码

```java
// Java
class Solution {


  public int[] twoSum(int[] nums, int target) {
    int[] a = new int[2];
    int numsSize = nums.length;
    for (int i = 0; i < numsSize - 1; i++) {
      for (int j = i + 1; j < numsSize; j++) {
        if (nums[i] + nums[j] == target) {
          a[0] = i;
          a[1] = j;
          return a;
        }
      }
    }
    return new int[0];
  }
}
```

复杂度分析：

- 时间复杂度：O(n^2)
- 空间复杂度 : O(1)