

# 0001-0100

## 0001-0100

- 1. 两数之和
  - 题目
  - 解答思路
- 7. 整数反转
  - 题目
  - 解答
- 9. 回文数
  - 题目
  - 解答

## 1. 两数之和

### 题目

1

给定一个整数数组 `nums` 和一个目标值 `target`,

2

请你在该数组中找出和为目标值的那 两个 整数, 并返回他们的数组下标。

3

你可以假设每种输入只会对应一个答案。但是, 你不能重复利用这个数组中同样的元素。

4

5

示例:

6

给定 `nums = [2, 7, 11, 15]`, `target = 9`

7

8

因为 `nums[0] + nums[1] = 2 + 7 = 9`

9

所以返回 `[0, 1]`

### 解答思路

No.	思路	时间复杂度	空间复杂度
01	暴力法: 2层循环遍历	$O(n^2)$	$O(1)$
02	两遍哈希遍历	$O(n)$	$O(n)$
03(最优)	一遍哈希遍历	$O(n)$	$O(n)$

1

# 暴力法: 2层循环遍历

2

`func twoSum(nums []int, target int) []int {`

3

`for i := 0; i < len(nums); i++ {`

4

`for j := i + 1; j < len(nums); j++ {`

5

`if nums[i]+nums[j] == target {`

```

6         return []int{i, j}
7     }
8 }
9 }
10 return []int{}
11 }
12
13 # 两遍哈希遍历
14 func twoSum(nums []int, target int) []int {
15     m := make(map[int]int, len(nums))
16     for k, v := range nums {
17         m[v] = k
18     }
19
20     for i := 0; i < len(nums); i++ {
21         b := target - nums[i]
22         if num, ok := m[b]; ok && num != i {
23             return []int{i, m[b]}
24         }
25     }
26     return []int{}
27 }
28
29 # 一遍哈希遍历
30 func twoSum(nums []int, target int) []int {
31     m := make(map[int]int, len(nums))
32     for i, b := range nums {
33         if j, ok := m[target-b]; ok {
34             return []int{j, i}
35         }
36         m[b] = i
37     }
38     return nil
39 }

```

## 7. 整数反转

### 题目

- 1 给出一个 32 位的有符号整数，你需要将这个整数中每位上的数字进行反转。
- 2
- 3 示例 1:
- 4 输入: 123
- 5 输出: 321
- 6
- 7 示例 2:
- 8 输入: -123
- 9 输出: -321

10  
11 示例 3：  
12 输入：120  
13 输出：21  
14  
15 注意：假设我们的环境只能存储得下 32 位的有符号整数，则其数值范围为  $[-2^{31}, 2^{31} - 1]$ 。  
16 请根据这个假设，如果反转后整数溢出那么就返回 0。

## 解答

No.	思路	时间复杂度	空间复杂度
01	使用符号标记，转成正数，循环得到%10的余数，再加上符号	$O(\log(x))$	$O(1)$
02(最优)	对x进行逐个%10取个位，一旦溢出，直接跳出循环	$O(\log(x))$	$O(1)$

```
1 // 使用符号标记，转成正数，循环得到%10的余数，再加上符号
2 func reverse(x int) int {
3     flag := 1
4     if x < 0 {
5         flag = -1
6         x = -1 * x
7     }
8
9     result := 0
10    for x > 0 {
11        temp := x % 10
12        x = x / 10
13
14        result = result*10 + temp
15    }
16
17    result = flag * result
18    if result > math.MaxInt32 || result < math.MinInt32 {
19        result = 0
20    }
21    return result
22 }
23
24 // 对x进行逐个%10取个位，一旦溢出，直接跳出循环
25 func reverse(x int) int {
26     result := 0
27     for x != 0 {
28         temp := x % 10
```

```

29     result = result*10 + temp
30     if result > math.MaxInt32 || result < math.MinInt32 {
31         return 0
32     }
33     x = x / 10
34 }
35 return result
36 }

```

## 9.回文数

### 题目

```

1  判断一个整数是否是回文数。回文数是指正序（从左向右）和倒序（从右向左）读都是一样的整数。
2
3  示例 1：输入：121 输出：true
4
5  示例 2：输入：-121 输出：false
6  解释：从左向右读，为 -121 。 从右向左读，为 121- 。因此它不是一个回文数。
7
8  示例 3：输入：10 输出：false
9  解释：从右向左读，为 01 。因此它不是一个回文数。
10
11 进阶：
12 你能不将整数转为字符串来解决这个问题吗？

```

### 解答

No.	思路	时间复杂度	空间复杂度
01(最优)	数学解法，取出后半段数字进行翻转，然后判断是否相等	$O(\log(x))$	$O(1)$
02	转成字符串，依次判断	$O(\log(x))$	$O(\log(x))$
03	转成byte数组，依次判断，同2	$O(\log(x))$	$O(\log(x))$

```

1  // 数学解法，取出后半段数字进行翻转，然后判断是否相等
2  func isPalindrome(x int) bool {
3      if x < 0 || (x%10 == 0 && x != 0) {
4          return false
5      }
6
7      revertedNumber := 0

```

```
8     for x > revertedNumber {
9         temp := x % 10
10        revertedNumber = revertedNumber*10 + temp
11        x = x / 10
12    }
13    // for example:
14    // x = 1221 => x = 12 revertedNumber = 12
15    // x = 12321 => x = 12 revertedNumber = 123
16    return x == revertedNumber || x == revertedNumber/10
17 }
18
19 // 转成字符串, 依次判断
20 func isPalindrome(x int) bool {
21     if x < 0 {
22         return false
23     }
24
25     s := strconv.Itoa(x)
26     for i, j := 0, len(s)-1; i < j; i, j = i+1, j-1 {
27         if s[i] != s[j] {
28             return false
29         }
30     }
31     return true
32 }
33
34 // 转成byte数组, 依次判断, 同2
35 func isPalindrome(x int) bool {
36     if x < 0 {
37         return false
38     }
39     arrs := []byte(strconv.Itoa(x))
40     Len := len(arrs)
41     for i := 0; i < Len/2; i++ {
42         if arrs[i] != arrs[Len-i-1] {
43             return false
44         }
45     }
46     return true
47 }
```