# 题目描述

**题目：4.两个有序数组的中间数**

There are two sorted arrays **nums1** and **nums2** of size m and n respectively.

Find the median of the two sorted arrays. The overall run time complexity should be O(log (m+n)).

You may assume **nums1** and **nums2** cannot be both empty.

**Example 1:**

```
nums1 = [1, 3]
nums2 = [2]

The median is 2.0
```

**Example 2:**

```
nums1 = [1, 2]
nums2 = [3, 4]

The median is (2 + 3)/2 = 2.5
```

给你两个int数组，数组里面的元素都是有序的，找出两个数组之间的中间数。

# 解题思路

参考归并排序里面的归并

# 代码实现

```
public double findMedianSortedArrays(int[] nums1, int[] nums2) {
  int len = nums1.length + nums2.length;
  int m = 0, n = 0;

  int before = 0;
  while (m + n <= (len / 2) - 1) {
    if (m >= nums1.length) before = nums2[n++];
    else if (n >= nums2.length) before = nums1[m++];
    else if (nums1[m] < nums2[n]) before = nums1[m++];
    else before = nums2[n++];
  }

  int mid;
  if (m >= nums1.length) mid = nums2[n++];
  else if (n >= nums2.length) mid = nums1[m++];
```

```
    else if (nums1[m] < nums2[n]) mid = nums1[m++];
    else mid = nums2[n++];

    if ((len & 1) == 0) { //
      return (before + mid) / 2.0;
    } else {
      return mid;
    }
  }
```

# 题目描述

**题目：7.整数反转**

Given a 32-bit signed integer, reverse digits of an integer.

Example 1:

```
Input: 123
Output: 321
```

Example 2:

```
Input: -123
Output: -321
```

Example 3:

```
Input: 120
Output: 21
```

# 解题思路

# 代码实现

```
public int reverse(int x) {
  int result = 0;
  while (x != 0) {
    int tail = x % 10;
    int newResult = result * 10 + tail;
    // 防止翻转后数字超过Interge.MAX_VALUE;
    if((newResult-tail)/10!=result) return 0;
    result = newResult;
    x = x / 10;
  }
  return result;
}
```

# 题目描述

**题目：9.对称数**

Determine whether an integer is a palindrome. An integer is a palindrome when it reads the same backward as forward.

Example 1:

```
Input: 121
Output: true
```

Example 2:

```
Input: -121
Output: false
Explanation: From left to right, it reads -121. From right to left, it becomes 121-.
Therefore it is not a palindrome.
```

Example 3:

```
Input: 10
Output: false
Explanation: Reads 01 from right to left. Therefore it is not a palindrome.
```

给你一个整数，判断它是否是一个对称数。

# 解题思路

其实就是上一道题目，整数反转的简化版本（因为不用考虑溢出问题）。

# 代码实现

```java
public boolean isPalindrome(int x) {
    if (x < 0) return false;
    int result = 0;
    int orig = x;
    while (x != 0) {
        result = result * 10 + x % 10;
        x = x / 10;
    }
    return result == orig;
}
```