

9 回文数

Label: 数学

给你一个整数 x ，如果 x 是一个回文整数，返回 `true`；否则，返回 `false`。

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。例如，121 是回文，而 123 不是。

输入: $x = 121$

输出: `true`

输入: $x = -121$

输出: `false`

- 转为字符数组，遍历

```
class Solution {
    public boolean isPalindrome(int x) {
        if (x < 0) return false;

        String a = Integer.valueOf(x).toString();
        char[] array = a.toCharArray();
        int i = 0;
        int j = array.length-1;
        while (i < j) {
            if (array[i] != array[j]) return false;
            i++;
            j--;
        }
        return true;
    }
}
```

也可以转为字符数组用栈

- 数学运算

```
class Solution {
    public boolean isPalindrome(int x) {
        //边界判断
        if (x < 0) return false;
        int div = 1;

        // 先得到初始量级
        while (x / div >= 10)
            div *= 10;

        int left = 0;
        int right = 0;
        while (x > 0) {
            left = x / div;
            right = x % 10;

            if (left != right) return false;

            x = (x % div) / 10; // x % div 去掉最高位  /10 去掉最低位
            div /= 100;
        }
        return true;
    }
}
```

- 计算倒序数值

```
class Solution {
    public boolean isPalindrome(int x) {
        if(x < 0) return false;

        int cur = 0;
        int num = x;
        while(num != 0) {
            cur = cur * 10 + num % 10;
            num /= 10;
        }
        return cur == x;
    }
}
```

- 计算倒序数值 (优化)

```
class Solution {
    public boolean isPalindrome(int x) {
        if (x < 0 || (x % 10 == 0 && x != 0))
            return false;

        int revertedNumber = 0;
        while (x > revertedNumber) {
            revertedNumber = revertedNumber * 10 + x % 10;
            x /= 10;
        }
        return x == revertedNumber || x == revertedNumber / 10;
    }
}
```