# 0007. 整数反转

ITCharge 大约 1 分钟

- 标签：数学
- 难度：中等

## 题目链接

- 0007. 整数反转 - 力扣

## 题目大意

给定一个 32 位有符号整数 x，将 x 进行反转。

## 解题思路

x 的范围为 $[-2^{31}, 2^{31} - 1]$，即 $[-214\quad\quad 3648, 2147483647]$。

反转的步骤就是让 x 不断对 10 取余，再除以 10，得到每一位的数字，同时累积结果。

注意累积结果的时候需要判断是否溢出。

当 ans * 10 + pop > INT_MAX ，或者 ans * 10 + pop < INT_MIN 时就会溢出。

按题设要求，无法在溢出之后对其判断。那么如何在进行累积操作之前判断溢出呢？

ans * 10 + pop > INT_MAX 有两种情况：

1. ans > INT_MAX / 10，这种情况下，无论是否考虑 pop 进位都会溢出；
2. ans == INT_MAX / 10，这种情况下，考虑进位，如果 pop 大于 IN_MAX 的个位数，就会导致溢出。

同理 ans * 10 + pop < INT_MIN 也有两种情况：

1. ans < INT_MIN / 10
2. ans == INT_MIN / 10 且 pop < INT_MIN 的个位数，就会导致溢出

# 代码

```py
class Solution:
    def reverse(self, x: int) -> int:
        INT_MAX_10 = (1<<31)//10
        INT_MIN_10 = int((-1<<31)/10)
        INT_MAX_LAST = (1<<31) % 10
        INT_MIN_LAST = (-1<<31) % -10
        ans = 0
        while x:
            pop = x % 10 if x > 0 else x % -10
            x = x // 10 if x > 0 else int(x / 10)
            if ans > INT_MAX_10 or (ans == INT_MAX_10 and pop > INT_MAX_LAST):
                return 0
            if ans < INT_MIN_10 or (ans == INT_MIN_10 and pop < INT_MIN_LAST):
                return 0
            ans = ans*10+pop
        return ans
```