

# BAT算法面试题(七)--反转整数

文章出自 Hello Code 开发者学习平台 CC老师

获取更新文章/视频 关注公众号:

**HelloCode开发者学习平台**



## 一.题目

给定一个 32 位有符号整数，将整数中的数字进行反转。

- 例如:

输入: 123

输出: 321

输入: 120

输出: 21

输入: -123

输出: -321

- 注意

假设我们的环境只能存储 32 位有符号整数，其数值范围是  $[-2^{31}, 2^{31} - 1]$ 。根据这个假设，如果反转后的整数溢出，则返回 0。

## 二.解决方案

- 方法：弹出和推入数字 & 溢出前进行检查

- 思路:

我们可以一次构建反转整数的一位数字。在这样做的时候，我们可以预先检查向原整数附加另一位数字是否会导致溢出。

- 算法

反转整数可以和反转字符串一起类比实现.

在没有辅助的堆栈/数组的帮助下,"弹出"和"推入"数字,可以尝试有用数学的方式.

```
//pop 操作:
pop = x % 10;
x /= 10;

//push 操作:
temp = rev * 10 + pop;
rev = temp;
```

但是,当因为当  $\text{temp} = \text{rev} * 10 + \text{pop}$  时,有可能会造成溢出.

- 解决溢出

- 假设 `rev`, 为正数
- 如果 `temp = rev*10+pop`, 导致溢出. 那么一定 `rev >= INTMAX/10`;
- 如果 `rev > INTMAX / 10`; 那么 `temp = rev*10+pop` 一定会溢出
- 如果 `rev == INTMAX / 10`, 那么只要 `pop > 7`, `temp = rev*10+pop` 就会溢出

- 代码实现

### C++ Code

```
class Solution {
public:
    int reverse(int x) {
        int rev = 0;
        while (x != 0) {
```

```
        int pop = x % 10;
        x /= 10;
        if (rev > INT_MAX/10 || (rev == INT_MAX / 10 && pop > 7)) return 0;
        if (rev < INT_MIN/10 || (rev == INT_MIN / 10 && pop < -8)) return 0;
        rev = rev * 10 + pop;
    }
    return rev;
};
```

- **复杂度**

- 时间复杂度:  $O(\log(x))$
- 空间复杂度:  $O(1)$

