7.整数反转

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。

示例 1:

输入: 123

输出: 321

 示例 2:

输入: -123

输出: -321

示例 3:

输入: 120

输出: 21

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 [−231,  231 − 1]。请根据这个假设，如果反转后整数溢出那么就返回 0。

比较简单的一道题，通过判断整数的位数，考虑输入输出各种情况时的处理即可解出

下面是我自己提交的解法：

class Solution(object):

    def reverse(self, x):

        """

        :type x: int

        :rtype: int

        """

        res=0

        digit=len(str(x))-1

        flag=0

        if x<0:

            x\*=-1;

            digit-=1

            flag=1

        while x!=0:

            res+=x%10\*(10\*\*digit)

            x/=10

            digit-=1

        if flag==1:

            res\*=-1

        if -2\*\*31<res<2\*\*31-1:

            return res

        return 0

在别人的题解中也发现了通过将整数转化为列表的算法，但总体思路是差不多的：

class Solution:

def reverse(self, x: int) -> int:

if x>0:

lis=list(str(x))

a=len(lis)

y=0

for i in range(0,a):

y+=int(lis[i])\*(10\*\*i)

if x==0:

y=0

if x<0:

x=abs(x)

lis=list(str(x))

a=len(lis)

y=0

for i in range(a):

y+=int(lis[i])\*(10\*\*i)

y=0-y

if -2\*\*31<res<2\*\*31-1:

             return y

        return 0

官方给出的解法是这样的（c++）：

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;

}

};

个人感觉比较巧妙的是递归解法：

class Solution {

public:

int reverse(int x) {

string s, m = x > 0 ? "2147483647" : "-2147483648";

for(s = x >= 0 ? "" : "-"; x; x/=10) s += to\_string(abs(x % 10));

return s.size() == m.size() and s > m or s == "" ? 0 : stoi(s);

}

};