题目：

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

给的是int类型的数字，Integer有个函数是reverse。我试了试，他并不是一个简单的倒置数字，所以不能应用与本题。

首先我想到的是给int转化成string来进行倒置。

转换方法String s = Integer.toString(int x);

但是直接转换很麻烦，需要判断第一位的正负，还要判断尾数的0.所以这种方法先放下。

正负数字差别就是前边的正负号，我们抛开正负号，直接先倒置数字，倒置出来了数字然后再添加正负号。

怎么倒置数字呢？开始我还是想在字符串里边直接倒置，就是找个轴心点左右互换呗。但是字符串建立了就不能改变的，所以这种方法并不行。

然后我就想到了用stringbuffer。

就是先把数字转化成为String，然后倒着来一个一个用string。append加入Stringbuffer。在加入过程中需要判断是否为0.不能仅仅判断是否为0，如果这样数字中间的0就不能加入了。在判断为0的前提需要判断stringbuffer是否为空，因为只有倒置后开头的0不能加入，如果stringbuffer里面有东西就是说明这个0在中间，直接加入就行了。倒置好的数字保存在Stringbuffer中，我们需要想办法把Stringbuffer转化成为int。两种方法：

（1）int res=Integer.valueOf(sb.toString)

（2）int res =Integer. parseInt(sb.toString)

至于怎么判断stringbuffer是否为空，用stringbuffer。Length（）是否等于0来判断。其他的方法就是将stringbuffer。Tostring成string后用string.equal（“”）或者string=“”来判断，因为stringbuffer没有重写equal方法，所以不能用直接equal方法来判断，需转化成String后在用equal方法判断。

做好了数字倒置，我们来解决符号问题。

正数就是简单的数字倒置，那么负数呢？直接0减去负数对应的正数不就好了吗.本题目完成。

算法的时间复杂度O（n），官网没给答案。