1881. Maximum Value after Insertion

Medium 114 29 Add to List Share

You are given a very large integer `n`, represented as a string, and an integer digit `x`. The digits in `n` and the digit `x` are in the **inclusive** range `[1, 9]`, and `n` may represent a **negative** number.

You want to **maximize** `n` **'s numerical value** by inserting `x` anywhere in the decimal representation of `n`. You **cannot** insert `x` to the left of the negative sign.

- For example, if `n = 73` and `x = 6`, it would be best to insert it between `7` and `3`, making `n = 763`.
- If `n = -55` and `x = 2`, it would be best to insert it before the first `5`, making `n = -255`.

Return *a string representing the* ***maximum*** *value of* `n` *after the insertion*.

Example 1:

```
Input: n = "99", x = 9
Output: "999"
Explanation: The result is the same regardless of where
you insert 9.
```

Example 2:

```
class Solution:
    def maxValue(self, n: str, x: int) ->
str:

        string = list(n)
        if string[0] !='-':
            for i in range(0,len(string)):
                if x >= int(string[i]):
                    string.insert(i,str(x))
                    break
        elif string[0] == '-':
            for i in range(1,len(string)):
                if x <= int(string[i]):
                    string.insert(i,str(x))
                    break

        p = int("".join(string))
        return(str(p))
```