402. Remove K Digits

Given string num representing a non-negative integer `num`, and an integer `k`, return *the smallest possible integer after removing* `k` *digits from* `num`.

Example 1:

```
Input: num = "1432219", k = 3
Output: "1219"
Explanation: Remove the three digits 4, 3, and 2 to form the new number 1219 which is the smallest.
```

Example 2:

```
Input: num = "10200", k = 1
Output: "200"
Explanation: Remove the leading 1 and the number is 200. Note that the output must not contain leading zeroes.
```

Example 3:

```
Input: num = "10", k = 2
Output: "0"
Explanation: Remove all the digits from the number and it is left with nothing which is 0.
```

Constraints:

- `1 <= k <= num.length <= 10<sup>5</sup>`
- `num` consists of only digits.
- `num` does not have any leading zeros except for the zero itself.

```python
class Solution:
    def removeKdigits(self, s: str, k: int) -> str:
        stack = []

        for i in range(len(s)):
            ch = s[i]

            while len(stack) and k>0 and stack[-1]>int(ch):
                stack.pop()
                k-=1
```

```
        stack.append(int(ch))

    while k>0:
        stack.pop()
        k-=1

    ans = ''.join([str(x) for x in stack])
    if len(ans)==0:
        return '0'
    if len(ans)==1:
        return ans
    idx = 0
    while idx<len(ans):
        if ans[idx]=='0':
            idx+=1
        else:
            break
    ans  = ans[idx:]
    if len(ans)==0:
        return '0'
    else:
        return ans
```