HW_Remove K Digits 2

$$\frac{1432219}{num} \quad \frac{3}{k}$$

	0	1	2	3	4	5	6
	'1'	'4'	'3'	'2'	'2'	'1'	'9'

$Sb = 9121$

reverse()

$1219 \rightarrow o/p$

$St = [\ ]$

$i = 0,\ '1',\ k = 3 > 0,\ St = [1]$

$i = 1,\ '4',\ k = 3 > 0,\ 1 > 4 f,\ St = [1,4]$

$i = 2,\ '3',\ k = 3 > 0,\ 4 > 3 T,\ St = [1,3],\ k--$

$i = 3,\ '2',\ k = 2 > 0,\ 3 > 2 T,\ St = [1,2],\ k--$

$i = 4,\ '2',\ k = 1 > 0,\ 2 > 2 f,\ St = [1,2,2]$

$\qquad k = 1 > 0,\ 2 > 1 T,\ St = [1,2,1],\ k--$

$i = 5,\ '1',\ k = 1 > 0,$

$i = 6,\ '9',\ k = 0 > 0\ \ 1 > 9 f,\ St = [1,2,1,9]$

```java
public static String remove(String num, int k){

    if(k == num.length()){
        return "0";
    }

    Stack<Character> st = new Stack<>();

    for(char c : num.toCharArray()){
        while(!st.isEmpty() && k > 0 && st.peek() > c){
            st.pop();
            k--;
        }
        st.push(c);
    }

    // k > 0 even after all the iterations
    while(k > 0){
        st.pop();
        k--;
    }
}
```

O(n)

```java
    // k > 0 even after all the iterations
    while(k > 0){
        st.pop();
        k--;
    }

    StringBuilder sb  = new StringBuilder();

    while(!st.isEmpty()){
        sb.append(st.pop());
    }
    sb.reverse();

    // remove the leading zeros
    while(sb.length() > 1 && sb.charAt(0) == '0'){
        sb.deleteCharAt(0);
    }

    return sb.toString();
}
```

HW_Remove Outermost Parentheses 4

$S = \big(() \; () \big) \big| \big(() \big)$

$\boxed{(\;)\;(\;)} + \boxed{(\;)}$

$((A+B))$

$(\;)\;(\;) \qquad + \qquad (\;) \quad = \quad \underline{(\;)(\;)(\;)}$

$S = \big(() \; () \big) \quad = \quad () \; ()$

$S = $ (( ) ( ))

indices: 0 1 2 3 4 5

Step	index	char	Action	Stack	Sb
1	0	'('	→ push('(')	[( ]	". "
2	1	'('	→ push('(') → Append('(' if str not empty)	[(, ( ]	" ("
3	2	')'	→ pop() → append	[( ]	" () "
4	3	'('	→ push → append	[(, ( ]	" ()("
5	4	')'	→ pop → append	[( ]	" ()() "
6	5	')'	→ pop	[ ]	" ()() "

```java
public static String removeOutermost(String str){

    Stack<Character> st = new Stack<>();
    StringBuilder sb = new StringBuilder();

    for(int i = 0; i < str.length(); i++){
        char c = str.charAt(i);

        if(c == '('){
            if(!st.isEmpty()){
                sb.append(c);
            }
            st.push(c);
        }
        else{
            st.pop();
            if(!st.isEmpty()){
                sb.append(c);
            }
        }
    }
    return sb.toString();
}
```

$O(n)$