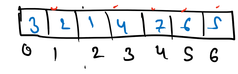
Lec-37

04/04/24 10:45 AM



You are given a **0-indexed** string `pattern` of length `n` consisting of the characters `'I'` meaning **increasing** and `'D'` meaning **decreasing**.

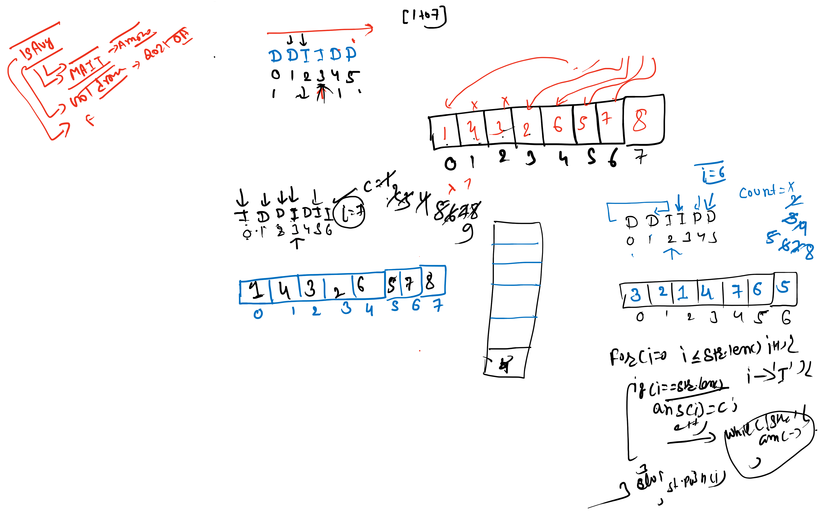
A **0-indexed** string `num` of length `n + 1` is created using the following conditions:

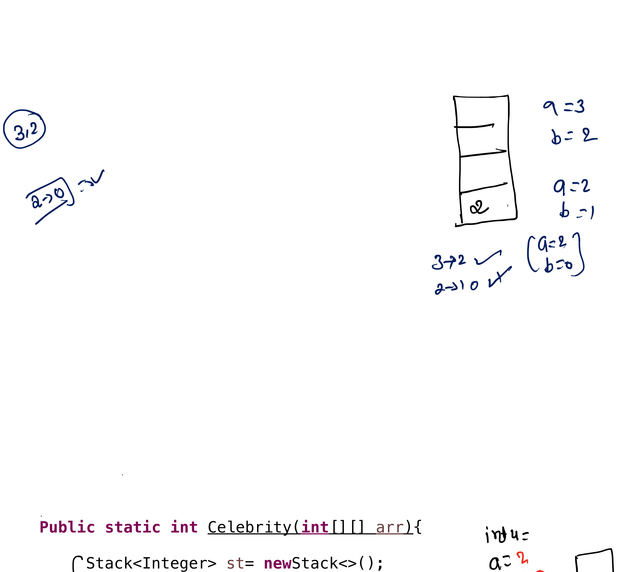
- `num` consists of the digits `'1'` to `'9'`, where each digit is used **at most** once.
- If `pattern[i] == 'I'`, then `num[i] < num[i + 1]`.
- If `pattern[i] == 'D'`, then `num[i] > num[i + 1]`.

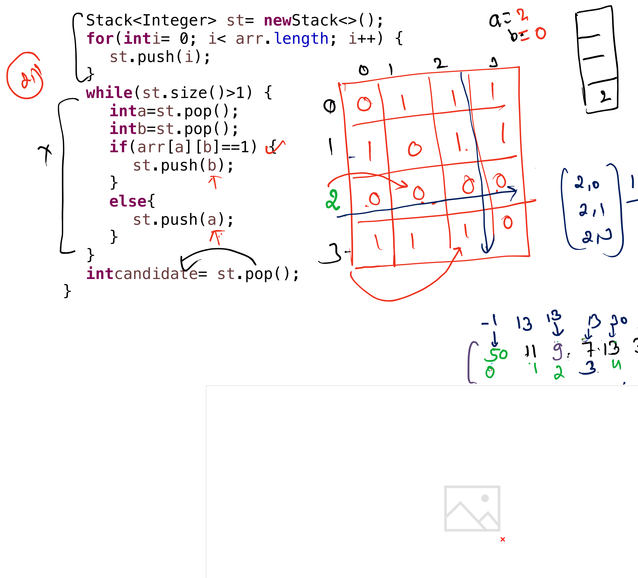
Return *the lexicographically **smallest** possible string* `num` *that meets the conditions.*

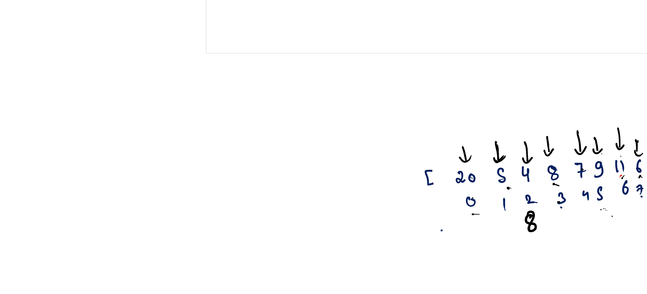
Example 1:

```
Input: pattern = "IIIDIDDD"
Output: "123549876"
```









```
Public static int Celebrity(int[][] arr){
    Stack<Integer> st= newStack<>();
    for(inti= 0; i< arr.length; i++) {
        st.push(i);
    }
    while(st.size()>1) {
        inta=st.pop();
        intb=st.pop();
        if(arr[a][b]==1) {
            st.push(b);
        }
        else{
            st.push(a);
        }
    }
    intcandidate= st.pop();
}
```