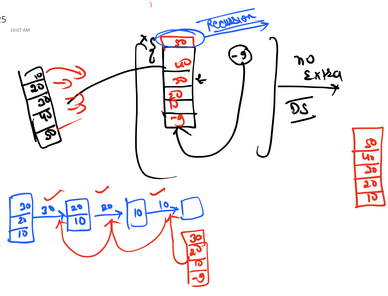
Lec-25

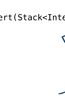
30/09/24 10:07 AM

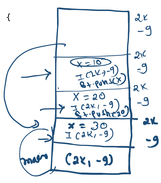


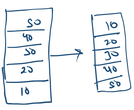


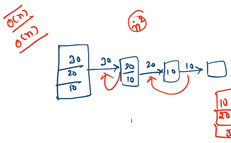



```
public static void Insert(Stack<Integer> st, intitem) {
    if(st.isEmpty()) {
        st.push(item);
        return;
    }
    intx= st.pop();
    Insert(st, item);
    st.push(x);
}
```

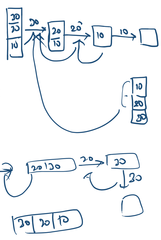




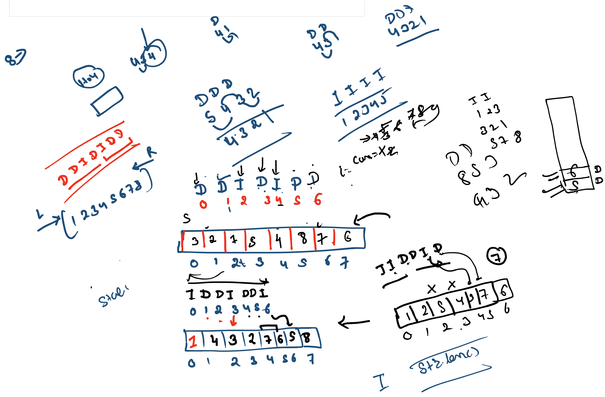




```
Public static voidRevser(Stack<Integer> st) {
    if(st.isEmpty()) {
        return;
    }
    intx= st.pop();
    Revser(st);
    Insert(st, x);
}
public static voidInsert(Stack<Integer> st, intitem) {
    if(st.isEmpty()) {
        st.push(item);
        return;
    }
    intx= st.pop();
    Insert(st, item);
    st.push(x);
}
```







```
public static String Construct_Smallest(String s) {
    int[] arr= newint[s.length() + 1];
    Stack<Integer> st= newStack<>();
    intc= 1;
    for(inti= 0; i<=s.length(); i++) {
        if(i==s.length() || s.charAt(i) == 'I') {
            arr[i] = c;
            c++;
            while(!st.isEmpty()) {
                arr[st.pop()] =c;
                c++;
            }
        }else{
            st.push(i);
        }
    }
    String ans= "";
    for(inti= 0; i< arr.length; i++) {
        ans= ans+ arr[i];
    }
    return ans;
}
```