FIND THE MINIMUM ELEMENT IN A STACK IN CONSTANT TIME

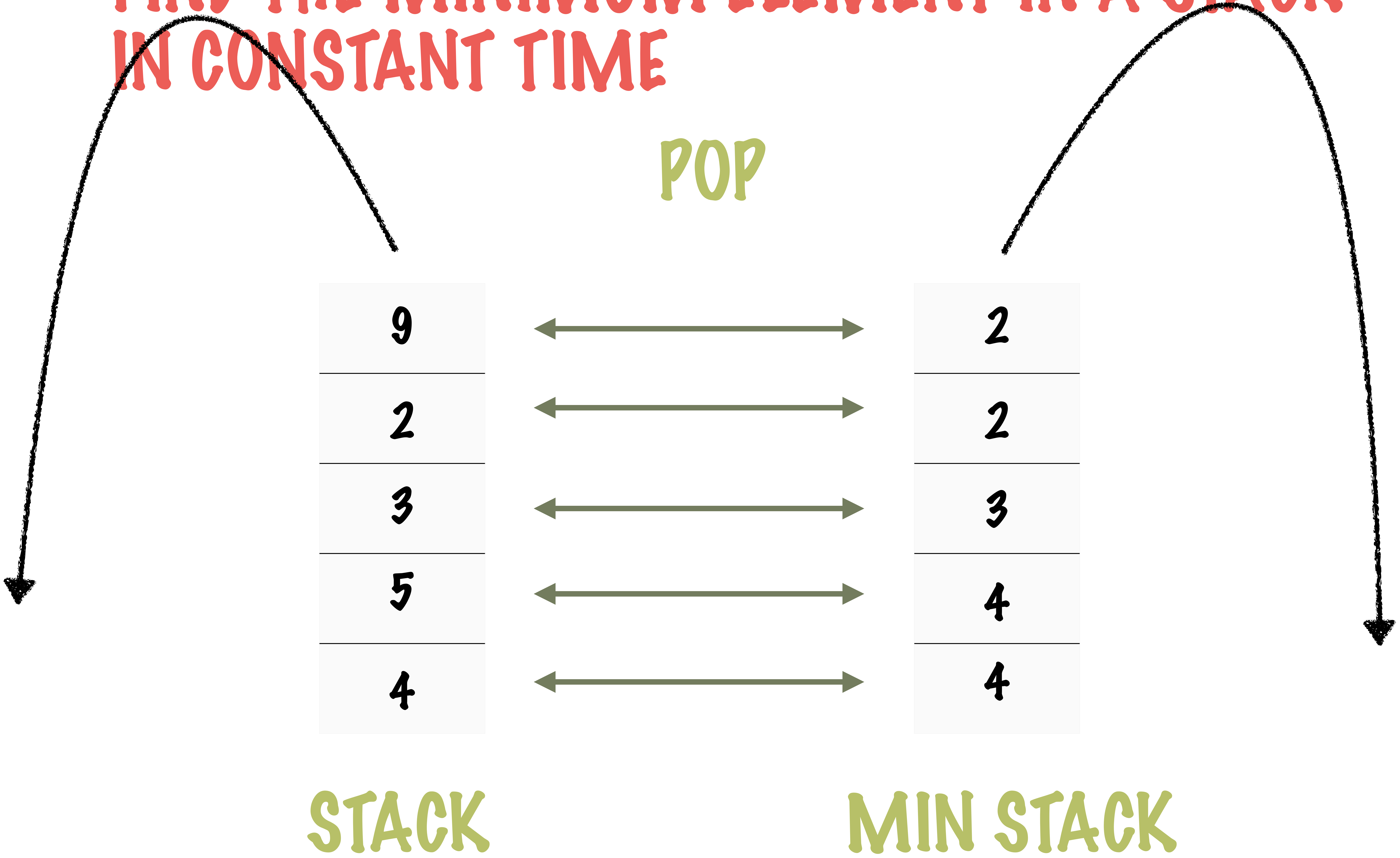
KEEP TRACK OF THE MINIMUM ELEMENT FOR EACH ELEMENT PUSHED ON TO THE STACK

HAVE 2 STACKS, ONE TO HOLD THE ACTUAL ELEMENTS, AND ANOTHER TO HOLD THE MINIMUM ELEMENT CORRESPONDING TO EVERY ELEMENT ADDED TO THE STACK

WHEN EACH ELEMENT IS PUSHED ONTO THE STACK COMPARE IT WITH THE LAST MINIMUM ELEMENT AND PUSH BOTH THE MAIN STACK AND THE MINIMUM ELEMENT STACK TOGETHER

DO THE SAME WHEN POPPING AN ELEMENT FROM THE STACK

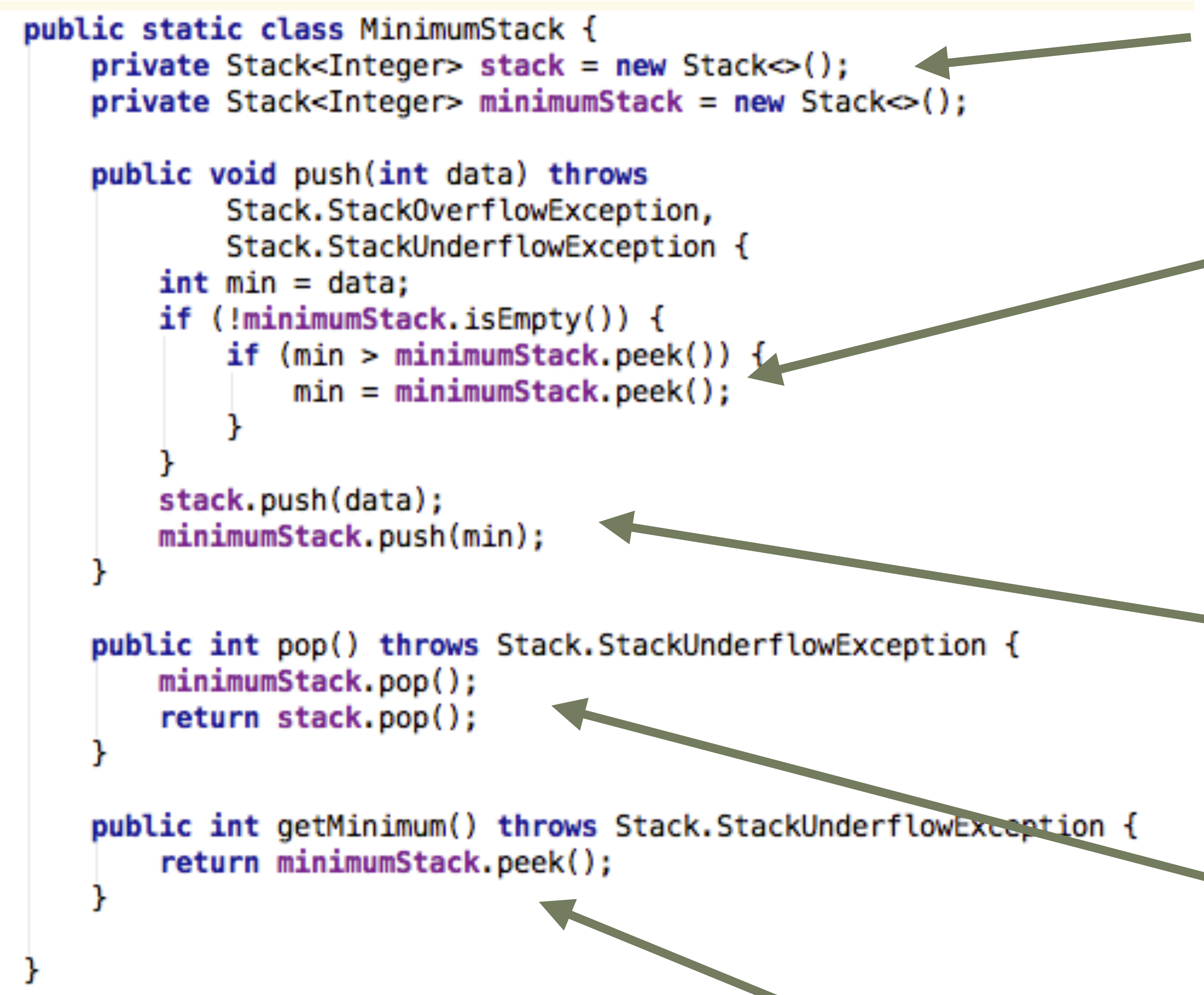
FIND THE MINIMUM ELEMENT IN A STACK IN CONSTANT TIME



POP

STACK	MIN STACK
9	2
2	2
3	3
5	4
4	4

USE STACKS TO FIND THE MINIMUM ELEMENT



```java
public static class MinimumStack {
    private Stack<Integer> stack = new Stack<>();
    private Stack<Integer> minimumStack = new Stack<>();

    public void push(int data) throws
            Stack.StackOverflowException,
            Stack.StackUnderflowException {
        int min = data;
        if (!minimumStack.isEmpty()) {
            if (min > minimumStack.peek()) {
                min = minimumStack.peek();
            }
        }
        stack.push(data);
        minimumStack.push(min);
    }

    public int pop() throws Stack.StackUnderflowException {
        minimumStack.pop();
        return stack.pop();
    }

    public int getMinimum() throws Stack.StackUnderflowException {
        return minimumStack.peek();
    }

}
```

SET UP ONE STACK TO HOLD THE INFORMATION AND ANOTHER TO HOLD THE MINIMUM ELEMENT

FIND THE MINIMUM BETWEEN THE ELEMENT JUST ADDED AND THE TOP OF THE MINIMUM STACK

PUSH THE ELEMENT ON TO THE STACK AND THE MINIMUM ELEMENT ON TO THE MINIMUM STACK

POP SHOULD POP ELEMENTS FROM BOTH STACKS

GETMINIMUM IS NOW AN O(1) OPERATION