Min Stack Implementation

Give DSA notes for the below problem Min STACK. Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.

Implement the MinStack class:

MinStack() initializes the stack object.
void push(int val) pushes the element val onto the stack.
void pop() removes the element on the top of the stack.
int top() gets the top element of the stack.
int getMin() retrieves the minimum element in the stack.
implement a solution with O(1) time complexity for each function.

Min Stack Notes

The notes for designing and implementing a Min Stack are ready. Let me know if you need further clarifications or edits!

can you give notes based on the below program.

```
class MinStack {
    private Node head;

    class Node {
        int data;
        int min;
        Node next;

        Node(int data) {
            this.data = data;
        }
    }

    public void push(int val) {
        if(head==null) {
            head = new Node(val);
            head.min = val;
            return;
        }
        Node temp = new Node(val);
        temp.next = head;
        head = temp;
        head.min = (val < head.next.min) ? val : head.next.min;

    }

    public void pop() {
        head = head.next;
    }

    public int top() {
        return head.data;
    }

    public int getMin() {
        return head.min;
    }

}
```

2/2

The notes have been updated to reflect the given `MinStack` implementation using a node-based approach. Let me know if there's anything else you'd like to refine or expand on!