Min Stack

Try to solve the Min Stack problem.

We'll cover the following

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

Statement

Design a custom stack class, **Min Stack**, allowing us to push, pop, and retrieve the minimum value in constant time. Implement the following methods for **Min Stack**:

- Constructor: This initializes the Min Stack object.
- Pop(): This removes and returns from the stack the value that was most recently pushed onto it.
- **Push():** This pushes the provided value onto the stack.
- **Min Number():** This returns the minimum value in the stack in O(1) time.

Note: The time complexity of all the methods above should be O(1).

Constraints:

- $-2^{31} \le \text{value} \le 2^{31} 1$
- The Pop() and Min Number() methods will always be called on non-empty stacks.
- At most, 3×10^3 calls will be made to **Push()**, **Pop()**, and **Min Number()**.

Examples

?

Ττ

6



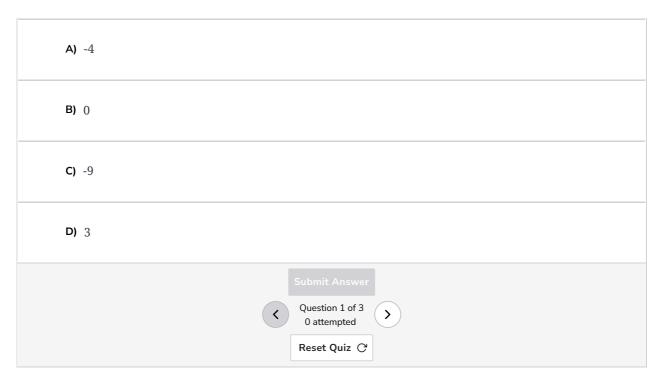
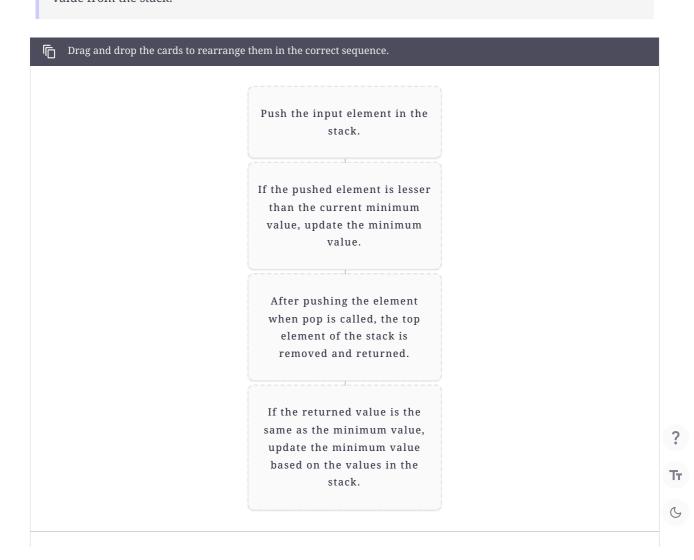


Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

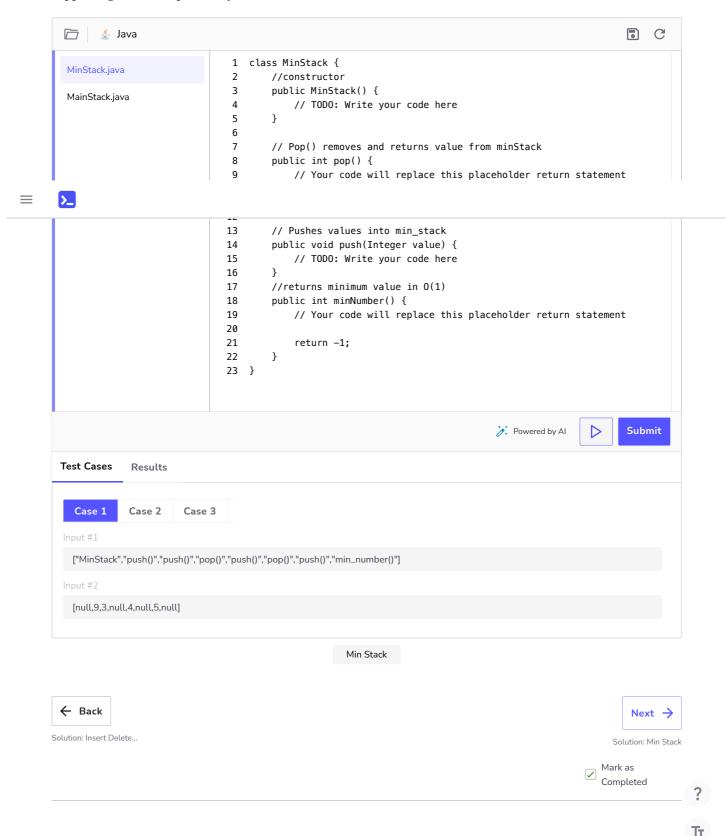
Note: These building blocks relate to the scenario of pushing, popping, and retrieving the minimum value from the stack.



Reset Show Solution Submit

Try it yourself

Implement your solution in MinStack.java in the following coding playground. You will need the provided supporting code to implement your solution.



6