Our Solution(s) Run Code

Solution 1

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class MinMaxStack:
    def __init__(self):
        self.minMaxStack = []
        self.stack = []

    # O(1) time | O(1) space
    def peek(self):
        return self.stack[len(self.stack) - 1]

    # O(1) time | O(1) space
    def pop(self):
        self.minMaxStack.pop()
        return self.stack.pop()

    # O(1) time | O(1) space
    def push(self, number):
        newMinMax = {"min": number, "max": number}
        if len(self.minMaxStack):
            lastMinMax = self.minMaxStack[len(self.minMaxStack) - 1]
            newMinMax["min"] = min(lastMinMax["min"], number)
            newMinMax["max"] = max(lastMinMax["max"], number)
        self.minMaxStack.append(newMinMax)
        self.stack.append(number)

    # O(1) time | O(1) space
    def getMin(self):
        return self.minMaxStack[len(self.minMaxStack) - 1]["min"]

    # O(1) time | O(1) space
    def getMax(self):
        return self.minMaxStack[len(self.minMaxStack) - 1]["max"]

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
# Feel free to add new properties and methods to the class.
class MinMaxStack:
    def peek(self):
        # Write your code here.
        pass

    def pop(self):
        # Write your code here.
        pass

    def push(self, number):
        # Write your code here.
        pass

    def getMin(self):
        # Write your code here.
        pass

    def getMax(self):
        # Write your code here.
        pass

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.