Our Solution(s) Run Code

Solution 1

```java
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

import java.util.*;

class Program {
  static class MinMaxStack {
    List<Map<String, Integer>> minMaxStack = new ArrayList<Map<S
    List<Integer> stack = new ArrayList<Integer>();

    // O(1) time | O(1) space
    public int peek() {
      return stack.get(stack.size() - 1);
    }

    // O(1) time | O(1) space
    public int pop() {
      minMaxStack.remove(minMaxStack.size() - 1);
      return stack.remove(stack.size() - 1);
    }

    // O(1) time | O(1) space
    public void push(int number) {
      Map<String, Integer> newMinMax = new HashMap<String, Integ
      newMinMax.put("min", number);
      newMinMax.put("max", number);
      if (minMaxStack.size() > 0) {
        Map<String, Integer> lastMinMax =
            new HashMap<String, Integer>(minMaxStack.get(minMaxS
        newMinMax.replace("min", Math.min(lastMinMax.get("min"),
        newMinMax.replace("max", Math.max(lastMinMax.get("max"),
      }
      minMaxStack.add(newMinMax);
      stack.add(number);
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```java
class Program {
  // Feel free to add new properties and methods to the class.
  static class MinMaxStack {
    public int peek() {
      // Write your code here.
      return -1;
    }

    public int pop() {
      // Write your code here.
      return -1;
    }

    public void push(Integer number) {
      // Write your code here.
    }

    public int getMin() {
      // Write your code here.
      return -1;
    }

    public int getMax() {
      // Write your code here.
      return -1;
    }
  }
}

```

Our Tests Custom Output Submit Code

```
}

public void testMinMaxStack(int min, int max, int peek, Progra
  while (assertTrue(stack.getMin() == min);
  while (assertTrue(stack.getMax() == max);
  while (assertTrue(stack.peek() == peek);
}

@Test
public void testCases() {
  Program.MinMaxStack stack = new Program.MinMaxStack();
  stack.push(5);
  testMinMaxPeek(5, 5, 5, stack);
  stack.push(7);
  testMinMaxPeek(5, 7, 7, stack);
  stack.push(3);
  testMinMaxPeek(3, 7, 3, stack);
  stack.push(8);
  testMinMaxPeek(3, 8, 8, stack);
  stack.push(1);
  testMinMaxPeek(1, 8, 1, stack);
  stack.push(9);
```

Run or submit code when you're ready.