Problem Challenge 3

We'll cover the following

- Frequency Stack (hard)
- Try it yourself

Frequency Stack (hard)

Design a class that simulates a Stack data structure, implementing the following two operations:

1. `push(int num)` : Pushes the number 'num' on the stack.
2. `pop()` : Returns the most frequent number in the stack. If there is a tie, return the number which was pushed later.

Example:

```
After following push operations: push(1), push(2), push(3), push(2), push(1), push(2), push(5)

1. pop() should return 2, as it is the most frequent number
2. Next pop() should return 1
3. Next pop() should return 2
```

Try it yourself

Try solving this question here:

Java | Python3 | JS | C++

```java
import java.util.*;

class FrequencyStack {

  public void push(int num) {
    // TODO: Write your code here
  }

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

  public static void main(String[] args) {
    FrequencyStack frequencyStack = new FrequencyStack();
    frequencyStack.push(1);
    frequencyStack.push(2);
    frequencyStack.push(3);
    frequencyStack.push(2);
    frequencyStack.push(1);
    frequencyStack.push(2);
    frequencyStack.push(5);
    System.out.println(frequencyStack.pop());
    System.out.println(frequencyStack.pop());
    System.out.println(frequencyStack.pop());
  }
}

```

Run | Save | Reset

Back

Solution Review: Problem Challenge 2

Next

Solution Review: Problem Challenge 3

Mark as Completed