Maximum Frequency Stack

Try to solve the Maximum Frequency Stack problem.

We'll cover the following

Statement

Design a stack-like data structure. You should be able to push elements to this data structure and pop elements with maximum frequency.

You'll need to implement the **FreqStack** class that should consist of the following:

- **FreqStack**: This is a class used to declare a frequency stack.
- **Push(value)**: This is used to push an integer data onto the top of the stack.
- **Pop()**: This is used to remove and return the most frequent element in the stack.

> **Note:** If there is a tie for the most frequent element, then the most recently pushed element is removed and returned.

Constraints:

- $0 \leq$ `value` $\leq 10^9$
- At most, 2×10^3 calls will be made to **Push()** and **Pop()**.
- It is guaranteed that there will be at least one element in the stack before calling **Pop()**.

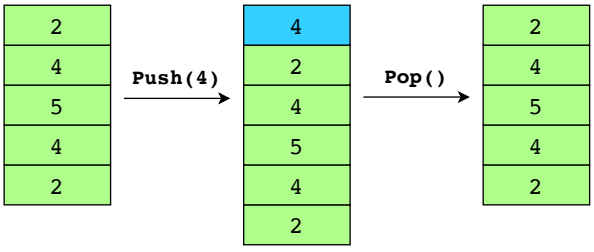
Examples

Sample example 1

Stack Elements: [2, 4, 5, 4, 2, 4]

Push 2, 4, 5, 4, 2 onto the stack.

Stack:



After pushing the last "4" onto the stack, it becomes

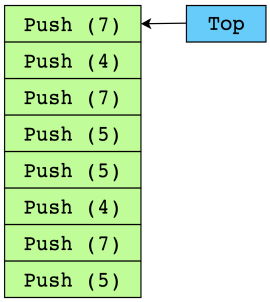
the most frequent element, we'll **pop** it.

Understand the problem

Let's take a moment to make sure we have correctly understood the problem. The quiz below helps us to check that we are solving precisely the right problem:

Maximum Frequency Stack

1 What is the output if four **Pop()** calls are made onto this stack?



A) [5, 4, 7, 5]

B) [5, 7, 4, 5]

C) [7, 5, 4, 7]

Submit Answer

Question 1 of 3
0 attempted

Reset Quiz

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.

Drag and drop the cards to rearrange them in the correct sequence.

Create a hash map or a dictionary to store frequencies of all elements.

Iterate over the input array and store its frequency in a hash map or dictionary. The corresponding frequency value is a stack containing all the elements with that frequency.

After all elements are pushed onto the stack, start removing the most frequent elements.

While removing elements, also decrease their frequency count from the hash map or dictionary.

When there's a case where two or more elements have the same frequency, pop the latest element that was pushed onto the stack.

Reset | Show Solution | Submit

Try it yourself

Implement your solution in the following coding playground:

Java

usercode > FreqStack.java

```java
import java.util.*;
class FreqStack {
    // Declare a FreqStack class containing frequency and group hashmaps
    // and maxFrequency integer
    // Use constructor to initialize the FreqStack object
    public FreqStack() {
        // Write your code here
    }

    // Write your code here
    }

    // Use the pop function to pop the showName from the FreqStack
    public int pop() {
        // Write your code here
        return -1;
    }
}

```

Powered by AI

Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

["FreqStack","push()","push()","push()","push()","pop()"]

Input #2

[null,5,7,7,5,null]

Maximum Frequency Stack

Back

Solution: Group Anagr...

Next

Solution: Maximum Fr...

Mark as Completed