895. Maximum Frequency Stack

19 March 2022 08:26 AM

Design a stack-like data structure to push elements to the stack and pop the most frequent element from the stack.

Implement the `FreqStack` class:

- `FreqStack()` constructs an empty frequency stack.
- `void push(int val)` pushes an integer `val` onto the top of the stack.
- `int pop()` removes and returns the most frequent element in the stack.
    - If there is a tie for the most frequent element, the element closest to the stack's top is removed and returned.

Example 1:

```
Input
["FreqStack", "push", "push", "push", "push",
"push", "push", "pop", "pop", "pop", "pop"]
[[], [5], [7], [5], [7], [4], [5], [], [], [],
[]]
Output
[null, null, null, null, null, null, null, 5, 7,
5, 4]
```

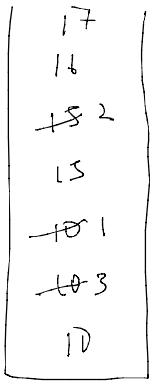
```
Explanation
FreqStack freqStack = new FreqStack();
freqStack.push(5); // The stack is [5]
freqStack.push(7); // The stack is [5,7]
freqStack.push(5); // The stack is [5,7,5]
freqStack.push(7); // The stack is [5,7,5,7]
freqStack.push(4); // The stack is [5,7,5,7,4]
freqStack.push(5); // The stack is [5,7,5,7,4,5]
freqStack.pop();   // return 5, as 5 is the most
frequent. The stack becomes [5,7,5,7,4].
freqStack.pop();   // return 7, as 5 and 7 is
the most frequent, but 7 is closest to the top.
The stack becomes [5,7,5,4].
freqStack.pop();   // return 5, as 5 is the most
frequent. The stack becomes [5,7,4].
freqStack.pop();   // return 4, as 4, 5 and 7 is
the most frequent, but 4 is closest to the top.
The stack becomes [5,7].
```

Normal Stack ⇒ LIFO

Here, max freq stack

push(val) ⇒ add

here we pop max freq element



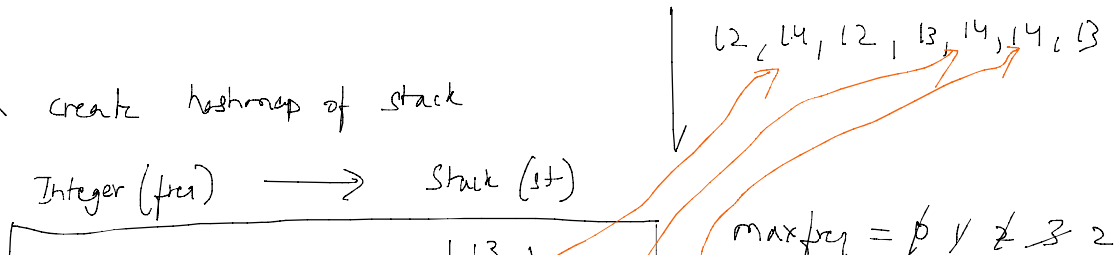
⇒ Here we pop 10

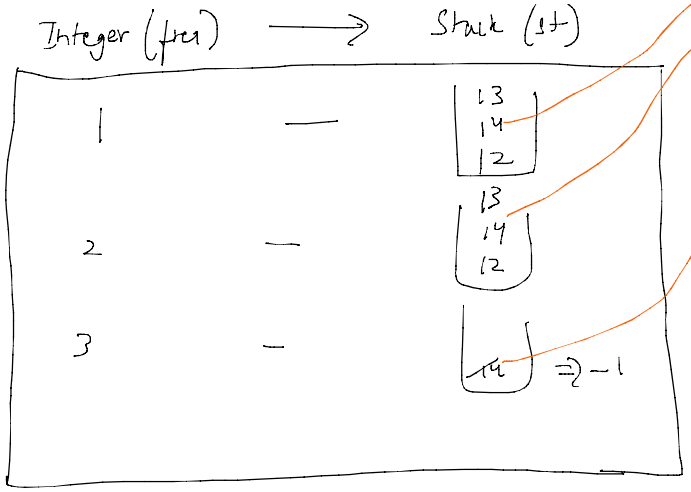
10, 15, so on ...

if in case 15 / 10 > has 3 freq element, the one in top is need to remove

You can create hashmap of stack

Integer (freq) ⟶ Stack (st)





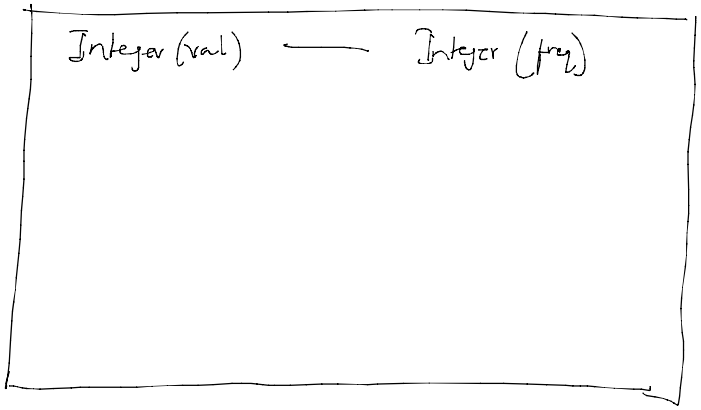
maxfreq = 0 ✗ 3 ✗ 2

Ans →

pop() ⇒ 14, 13,

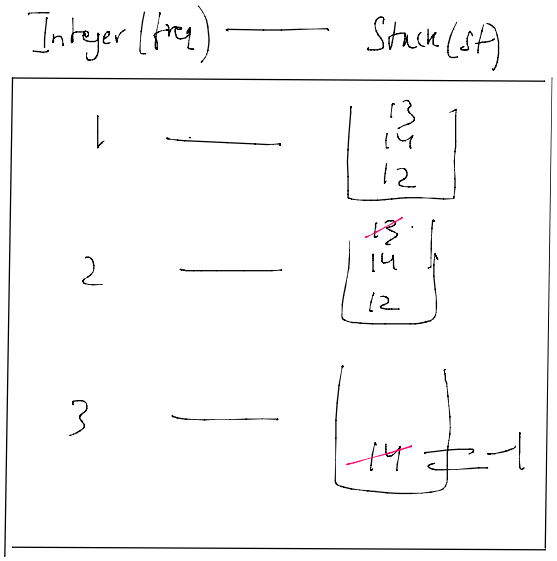
⇒ Once stack becomes empty mark it as -1, so max freq 3-1 = 2

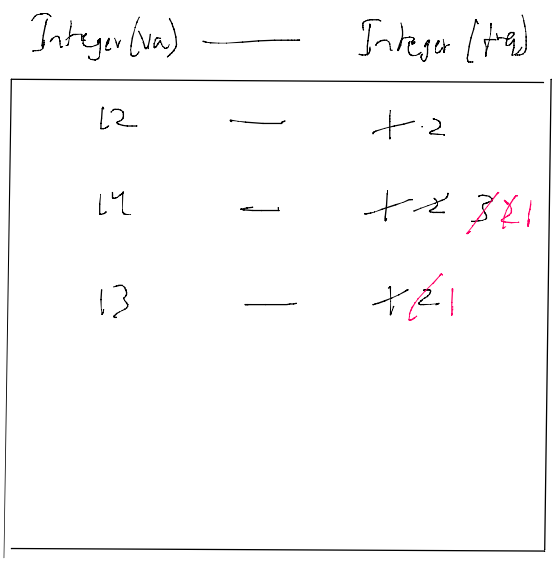
one more hashmap (to store in which stack)



Again dry run

12 14 12 13 14 13 14





maxfreq = 0 ✗ 3, 2

pop ⇒ 14 13 14

```java
class FreqStack {

    HashMap<Integer, LinkedList<Integer>> st;
    HashMap<Integer, Integer> fmap;
    int maxFreq;

    public FreqStack(){
        st = new HashMap<>();
        fmap = new HashMap<>();
        maxFreq = 0;
    }

    public void push(int val){
        int cFreq = fmap.getOrDefault(val,0);
        cFreq++;
        fmap.put(val,cFreq);

        if(st.containsKey(cFreq)==false){
            st.put(cFreq, new LinkedList<Integer>());
        }

        st.get(cFreq).addFirst(val);
        maxFreq = Math.max(maxFreq, cFreq);
    }

    public int pop(){
        int ans = st.get(maxFreq).removeFirst();
        int cFreq = fmap.get(ans);
        cFreq--;
        fmap.put(ans, cFreq);

        if(st.get(maxFreq).size()==0){
            maxFreq--;
        }
        return ans;
    }
}
```