232. Implement Queue using Stacks

Add to List ▼

Question Editorial Solution

My Submissions (/problems/implement-queue-using-stacks/submissions/)

Total Accepted: 69292 Total Submissions: 197927 Difficulty: Easy Contributors: Admin

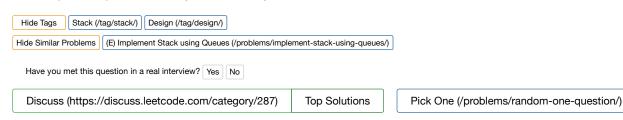
Implement the following operations of a queue using stacks.

- push(x) -- Push element x to the back of queue.
- pop() -- Removes the element from in front of queue.
- peek() -- Get the front element.
- empty() -- Return whether the queue is empty.

Notes:

- You must use only standard operations of a stack -- which means only push to top, peek/pop from top, size, and is empty operations are valid.
- Depending on your language, stack may not be supported natively. You may simulate a stack by using a list or deque (double-ended queue), as long as you use only standard operations of a stack.
- · You may assume that all operations are valid (for example, no pop or peek operations will be called on an empty queue).

Subscribe (/subscribe/) to see which companies asked this question



```
C++
                            C
                                 </>
  1
     class MyQueue {
     public:
  3
         stack<int> S1;
  5
         stack<int> S2;
  6
         /** Initialize your data structure here. */
  7
  8
         MyQueue() {
  9
 10
         }
 11
 12
 13
         /** Push element x to the back of queue. */
         void push(int x) {
 14
 15
             S2.push(x);
 16
         }
 17
         /** Removes the element from in front of queue and returns that element. */
 18
 19
         int pop() {
 20
             if(S1.empty()) shift();
             int res = S1.top();
 21
 22
             S1.pop();
 23
             return res;
 24
 25
 26
         /** Get the front element. */
 27
         int peek() {
 28
             if(S1.empty()) shift();
 29
             return S1.top();
 30
 31
         /** Returns whether the queue is empty. */
 32
 33
         bool empty() {
 34
             return S1.empty()&&S2.empty();
 35
 36
 37
         void shift(){
                                                                               Send Feedback (mailto:admin@leetcode.com?subject=Feedback)
 38
             while(!S2.empty()){
```

```
39
                 S1.push(S2.top());
40
                 S2.pop();
41
42
        }
43
44 };
45
46
47
     * Your MyQueue object will be instantiated and called as such:
    * MyQueue obj = new MyQueue();
48
49
     * obj.push(x);
     * int param_2 = obj.pop();
* int param_3 = obj.peek();
50
51
     * bool param_4 = obj.empty();
52
53
```

Custom Testcase

Contribute Testcase 2

Run Code

Submit Solution

Frequently Asked Questions (/faq/) | Terms of Service (/tos/)

Privacy

Copyright © 2017 LeetCode