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

Hide Tags | Stack (/tag/stack/) | Design (/tag/design/)

Hide Similar Problems | (E) Implement Stack using Queues (/problems/implement-stack-using-queues/)

Have you met this question in a real interview? Yes No

Discuss (https://discuss.leetcode.com/category/287) | Top Solutions | Pick One (/problems/random-one-question/)

C++

```
class MyQueue {

public:
    stack<int> S1;
    stack<int> S2;

    /** Initialize your data structure here. */
    MyQueue() {

    }


    /** Push element x to the back of queue. */
    void push(int x) {
        S2.push(x);
    }

    /** Removes the element from in front of queue and returns that element. */
    int pop() {
        if(S1.empty()) shift();
        int res = S1.top();
        S1.pop();
        return res;
    }

    /** Get the front element. */
    int peek() {
        if(S1.empty()) shift();
        return S1.top();
    }

    /** Returns whether the queue is empty. */
    bool empty() {
        return S1.empty()&&S2.empty();
    }

    void shift(){
        while(!S2.empty()){
```

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```
            S1.push(S2.top());
            S2.pop();
        }
    }

};

/**
 * Your MyQueue object will be instantiated and called as such:
 * MyQueue obj = new MyQueue();
 * obj.push(x);
 * int param_2 = obj.pop();
 * int param_3 = obj.peek();
 * bool param_4 = obj.empty();
 */
```

Notes

Custom Testcase

Contribute Testcase

Run Code | Submit Solution