

CU-Assessments/assignment6... Implement Queue using Stacks Queue Implementation with Sta... Convert JPG to PDF, Images JP... +

leetcode.com/problems/implement-queue-using-stacks/submissions/1579216861/

Gmail YouTube Maps Stake: Online Casin... Internshala Trainings Dashboard | Hacker... eDoc DSA-Bootcamp-Jav... Amazon Mechanical...

All Bookmarks

Problem List < > ✎ Run Submit Ctrl Enter

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 22 / 22 testcases passed Pankaj13860 submitted at Mar 19, 2025 21:11

Runtime 0 ms | Beats 100.00% Memory 41.32 MB | Beats 67.65%

Analyze Complexity

Java Auto

```
--  
13     public int pop() {  
14         peek();  
15         return output.pop();  
16     }  
17     public int peek() {  
18         if (output.isEmpty()) {  
19             while (!input.isEmpty()) {  
20                 output.push(input.pop());  
21             }  
22         }  
23         return output.peek();  
24     }  
25     public boolean isEmpty() {  
26         return input.isEmpty() && output.isEmpty();  
27     }  
28 }
```

Ln 5, Col 35

Code Java

```
import java.util.Stack;  
  
class MyQueue {  
    private Stack<Integer> input;  
    private Stack<Integer> output;  
    public MyQueue() {
```

Saved

Testcase Test Result

Accepted Runtime: 0 ms

Case 1

Input

CU-Assessments/assignment6... Min Stack - LeetCode Queue Implementation with Sta... Convert JPG to PDF, Images JP... (2) WhatsApp

leetcode.com/problems/min-stack/submissions/1579222469/

All Submissions Accepted 31 / 31 testcases passed Pankaj13860 submitted at Mar 19, 2025 21:17

Runtime: 5 ms | Beats 37.85% | Analyze Complexity

Memory: 44.37 MB | Beats 99.35%

Java

```
16     minStack.push(val);
17 }
18 }
19
20 public void pop() {
21     if (mainStack.isEmpty()) return;
22     int removed = mainStack.pop();
23     if (removed == minStack.peek()) {
24         minStack.pop();
25     }
26 }
27
28 public int top() {
29     return mainStack.peek();
30 }
31
32 public int getMin() {
33     return minStack.peek();
34 }
35 }
```

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1

Input

```
import java.util.Stack;

class MinStack {
    private Stack<Integer> mainStack;
    private Stack<Integer> minStack;
```

The screenshot shows a browser window with multiple tabs open, including one for "Inorder Traversal - Naukri.com". The main content is the code360 platform interface for a Java problem.

Current submission:

- Status: Accepted
- Test cases: 50/50
- EXP: 40/40
- Language: Java
- Penalty: 0%

A few seconds ago

Previous submissions:

Status	Test cases	Score	Penalty	Language
Accepted	50/50	40/40	0%	Java

Java (SE 1.8)

```
1 import java.util.List;
2 import java.util.ArrayList;
3 import java.util.Stack;
4
5 public class Solution {
6     public static List<Integer> getInOrderTraversal(TreeNode root) {
7         List<Integer> result = new ArrayList<>();
8         Stack<TreeNode> stack = new Stack<>();
9         TreeNode current = root;
10
11         while (current != null || !stack.isEmpty()) {
12             while (current != null) {
13                 stack.push(current);
14                 current = current.left;
15             }
16             current = stack.pop();
17             result.add(current.data);
18             current = current.right;
19         }
20     }
21     return result;
22 }
23
```

Last saved at 9:21 PM

Run Submit code View hints

CU-Assessments/assignment

Implement Stack using Queues

Queue Implementation with

Inorder Traversal - Naukri

Convert JPG to PDF, Image

(2) WhatsApp

leetcode.com/problems/implement-stack-using-queues/submissions/1579234461/

Gmail YouTube Maps Stake: Online Casino Internshala Trainings Dashboard | Hacker... eDoc DSA-Bootcamp-Java Amazon Mechanical...

All Bookmarks

Problem List

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 18 / 18 testcases passed

Pankaj13860 submitted at Mar 19, 2025 21:28

Runtime: 0 ms | Beats 100.00% Memory: 40.59 MB | Beats 99.96%

Analyze Complexity

Java Auto

```
import java.util.LinkedList;
import java.util.Queue;

class MyStack {
    private Queue<Integer> q1;
    private Queue<Integer> q2;

    public MyStack() {
        q1 = new LinkedList<>();
        q2 = new LinkedList<>();
    }

    public void push(int x) {
        q2.add(x);
        while (!q1.isEmpty()) {
            q2.add(q1.poll());
        }
        Queue<Integer> temp = q1;
        q1 = q2;
        q2 = temp;
    }
}
```

Saved Ln 14, Col 20

Testcase Test Result

Accepted Runtime: 0 ms

Case 1

Input

```
import java.util.LinkedList;
import java.util.Queue;

class MyStack {
    private Queue<Integer> q1;
    private Queue<Integer> q2;
```

CU-Assessments/assignment6... Implement Deque - Naukri Cod... Queue Implementation with Sta... Convert JPG to PDF, Images JP... (2) WhatsApp

naukri.com/code360/problems/deque_1170059?leftPanelTabValue=SUBMISSION

Gmail YouTube Maps Stake: Online Casin... Internshala Trainings Dashboard | Hacker... eDoc DSA-Bootcamp-Jav... Amazon Mechanical...

All Bookmarks

code 360 by codingninjas Interview problem list View all problems Your EXP 130

Problem Submissions Hints & solutions Discuss Java (SE 1.8)

Current submission Accepted A few seconds ago

Test cases EXP 40/40 Language Java Penalty 0%

Runtime graph Memory graph

You are better than 69.8% Runtime 1037 ms

968 1,047.8 1,127.5 1,207.3 1,287 Runtime (ms)

Did you find these test cases useful?

Showing recommended problems

LCA of three Nodes EXP 0/40 Easy

1 import java.util.*;
2
3 public class Deque {
4 private int[] arr;
5 private int front, rear, size, capacity;
6
7 // Constructor: Initializes deque with a given size.
8 public Deque(int n) {
9 this.capacity = n;
10 this.arr = new int[n];
11 this.front = -1;
12 this.rear = -1;
13 this.size = 0;
14 }
15
16 // Pushes 'X' to the front of the deque.
17 public boolean pushFront(int x) {
18 if (isFull()) return false;
19
20 if (isEmpty()) {
21 front = rear = 0;
22 } else {
23 front = (front - 1 + capacity) % capacity;
24 }
25
26 arr[front] = x;
27 size++;
28 return true;
29 }
30}

Last saved at 9:30 PM

Prev Run Submit code Next View hints

Screenshot of a coding challenge submission page on code360 by Coding Ninjas.

Current submission (Accepted, A few seconds ago)

Test cases: 20/20 | EXP: 40/40 | Language: Java | Penalty: 0%

Runtime graph (You are better than 83.39% | Runtime: 1459 ms)

Code Snippet:

```
public class Solution {
    static class Stack {
        private int[] arr;
        private int top;
        private int capacity;

        // Constructor: Initializes stack with given capacity.
        Stack(int capacity) {
            this.capacity = capacity;
            this.arr = new int[capacity];
            this.top = -1; // Empty stack
        }

        // Push an element onto the stack
        public void push(int num) {
            if (isFull() == 1) return; // Stack overflow check
            arr[++top] = num;
        }

        // Pop and return the top element
        public int pop() {
            return isEmpty() == 1 ? -1 : arr[top--]; // Stack underflow check
        }

        // Return the top element without popping
        public int top() {
            return isEmpty() == 1 ? -1 : arr[top];
        }

        // Check if the stack is empty (returns 1 if empty, 0 otherwise)
        public boolean isEmpty() {
            return top == -1;
        }
    }
}
```

Sample test case (Custom test case)

- Test case 6
- Test case 7
- Test case 8
- Test case 9
- Test case 10

Buttons: Run, View hints

CU-Assessments/assignment6... Queue Using Array | Practice | Queue Implementation with Sta... Convert JPG to PDF, Images JP... (2) WhatsApp

geeksforgeeks.org/problems/implement-queue-using-array/1

Gmail YouTube Maps Stake: Online Casin... Internshala Trainings Dashboard | Hacker... eDoc DSA-Bootcamp-Jav... Amazon Mechanical...

Courses Tutorials Jobs Practice Contests

Output Window

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully ✓

Test Cases Passed 170 / 170

Attempts : Correct / Total 1 / 1 Accuracy : 100%

Points Scored 1 / 1 Time Taken 2.73

Your Total Score: 2 ↑

Solve Next

C++ STL | Set 5 (queue) Queue Reversal Professor and Parties

Kick start your career with GfG 160! >

Java (1.8) Start Timer

```
1 // ( ) Driver Code Ends
2
3
4
5
6 class MyQueue {
7     int front, rear;
8     int arr[] = new int[100005];
9
10    MyQueue()
11    {
12        front=0;
13        rear=0;
14    }
15
16    //Function to push an element x in a queue.
17    void push(int x)
18    {
19        arr[rear] = x;
20        rear++;
21    }
22
23    //Function to pop an element from queue and return that element.
24    int pop()
25    {
26        if(front == rear) return -1;
27        int val = arr[front];
28        front++;
29        return val;
30    }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
```

Custom Input Compile & Run Submit

CU-Assessments/assignment6... Design Circular Queue - LeetCode Queue Implementation with Sta... Convert JPG to PDF, Images JP... (2) WhatsApp

leetcode.com/problems/design-circular-queue/submissions/1579250870/

Gmail YouTube Maps Stake: Online Casin... Internshala Trainings Dashboard | Hacker... eDoc DSA-Bootcamp-Jav... Amazon Mechanical...

All Bookmarks

Problem List Run Submit

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 59 / 59 testcases passed Pankaj13860 submitted at Mar 19, 2025 21:43

Runtime 4 ms | Beats 100.00% Analyze Complexity

Memory 44.93 MB | Beats 35.94% Analyze Complexity

Java Auto

```
// Returns the front element
public int Front() {
    return isEmpty() ? -1 : queue[front];
}

// Returns the rear element
public int Rear() {
    return isEmpty() ? -1 : queue[rear];
}

// Checks if the queue is empty
public boolean isEmpty() {
    return size == 0;
}

// Checks if the queue is full
public boolean isFull() {
    return size == capacity;
}
```

Saved Ln 53, Col 1

Testcase Test Result

Accepted Runtime: 0 ms Case 1 Input

```
class MyCircularQueue {
    private int[] queue;
    private int front, rear, size, capacity;

    // Constructor: Initializes queue with given capacity
    public MyCircularQueue(int k) {
```

The screenshot shows a browser window with multiple tabs open. The active tab is for a problem titled "Implement Stack using Linked List". The page has a dark theme with green highlights for successful submissions.

Compilation Results:

- Test Cases Passed: **1115 / 1115**
- Attempts: Correct / Total: **1 / 1**
- Accuracy: **100%**
- Points Scored: **2 / 2**
- Your Total Score: **4**

Solve Next:

- [Implement stack using array](#)
- [Queue Reversal](#)
- [Pairwise Consecutive Elements](#)

Java (1.8) Code:

```
1 // (//) Driver Code Ends
49 class MyStack {
50     // Node structure for Stack
51     class StackNode {
52         int data;
53         StackNode next;
54     }
55     StackNode(int a) {
56         data = a;
57         next = null;
58     }
59 }
60
61 StackNode top; // Points to the top of the stack
62
63 // Function to push an integer into the stack
64 void push(int a) {
65     StackNode newNode = new StackNode(a);
66     newNode.next = top; // Link the new node to the current top
67     top = newNode; // Update top
68 }
69
70 // Function to remove and return the top item from the stack
71 int pop() {
72     if (top == null) return -1; // Stack is empty
73
74     int poppedValue = top.data; // Get top element
75     top = top.next; // Move top to the next node
76     return poppedValue;
77 }
78
79
80
81 }
```

At the bottom right, there are buttons for "Custom Input", "Compile & Run", and "Submit".

Screenshot of a browser window showing a coding challenge on code360 by codingninjas.

The URL in the address bar is naukri.com/code360/problems/implement-queue-using-linked-list_8161235?leftPanelTabValue=SUBMISSION

The page title is "Implement Queue Using Link...

The main content area shows a Java code editor with the following code:

```
1 //***** Following is the class structure of the Node class:
2
3 class Node {
4     public int data;
5     public Node next;
6
7     Node()
8     {
9         this.data = 0;
10        this.next = null;
11    }
12
13    Node(int data)
14    {
15        this.data = data;
16        this.next = null;
17    }
18
19    Node(int data, Node next)
20    {
21        this.data = data;
22        this.next = next;
23    }
24
25    }
26
27
28 class Queue{
29     Node front;
30     Node rear;
31 }
```

The code editor interface includes tabs for "Java (SE 1.8)", "Submissions", "Hints & solutions", and "Discuss".

Current submission status: Accepted, submitted A few seconds ago.

Test cases: 50/50, EXP: 40/40, Language: Java, Penalty: 0%.

Previous submissions table:

Status	Test cases	Score	Penalty	Language
Accepted	50/50	40/40	0%	Java

Buttons at the bottom include "Prev", "Run", "Submit code", "Next", and "View hints".