

1. Implement Queue using Stack

The screenshot displays a coding platform interface for the problem "Implement Queue using Stack". The solution is accepted, with 22/22 testcases passed. The runtime is 0 ms, beating 100.00% of other solutions, and the memory usage is 9.74 MB, beating 26.95%.

Code:

```
class MyQueue {
private:
    stack<int> input;
    stack<int> output;
public:
    MyQueue() {}
    void push(int x) {
        input.push(x);
    }
    int pop() {
        peek();
        int val = output.top();
        output.pop();
        return val;
    }
};
```

Testcase 1:

Input: ["MyQueue", "push", "push", "peek", "pop", "empty"]

Output: [(), [1], [2], [], [1, 1], [null, null, null, 1, false]]

2. Implement Min Stack using Two Stacks

The screenshot displays a coding platform interface for the problem "Implement Min Stack using Two Stacks". The solution is accepted, with 31/31 testcases passed. The runtime is 7 ms, beating 19.90% of other solutions, and the memory usage is 23.96 MB, beating 18.36%.

Code:

```
class MinStack {
private:
    vector<vector<int>> st;
public:
    MinStack() {}
    void push(int val) {
        int min_val = getMin();
        if (st.empty() || min_val > val) {
            min_val = val;
        }
        st.push_back({val, min_val});
    }
    void pop() {}
};
```

Testcase 1:

Input: ["MinStack", "push", "push", "push", "getMin", "pop", "top", "getMin"]

Output: [(), [-2], [0], [-3], [1, 1], [1], [null, null, null, null, -3, null, 0, -2]]

3. Implement BST (Inorder Traversal) using Stack (Iterative DFS)

Problem List

Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 71 / 71 testcases passed

Prakhar Sharma submitted at Mar 19, 2023 15:11

Runtime 0 ms Beats 100.00% Memory 10.81 MB Beats 66.32%

Analyze Complexity

0.88% of solutions used 4 ms of runtime

Code C++

```
class Solution {
public:
    vector<int> inorderTraversal(TreeNode* root) {
        vector<int> ans;
        stack<TreeNode*> sta;
        while(true)
        {
            while(root)

```

More challenges

Code C++ Auto

```

10 {
11     sta.push(root);
12     root=root->left;
13 }
14 if(sta.empty())
15 {
16     break;
17 }
18 root=sta.top();
19 ans.push_back(root->val);
20 sta.pop();
21 root=root->right;
22 }
23 return ans;
24 }
25

```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3 Case 4

Input

root =

[1,null,2,3]

Output

[1,3,2]

Expected

4.Implement Stack using Queue

Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 18 / 18 testcases passed

Prakhar Sharma submitted at Mar 19, 2023 15:16

Runtime 0 ms Beats 100.00% Memory 9.31 MB Beats 85.26%

Analyze Complexity

class MyStack {

```

private:
    std::queue<int> q;

public:
    MyStack() {}

    void push(int x) {

```

More challenges

Code C++ Auto

```

13 }
14 }
15 }
16 int pop() {
17     int top = q.front();
18     q.pop();
19     return top;
20 }
21
22 int top() {
23     return q.front();
24 }
25
26 bool empty() {
27     return q.empty();
28 }
29

```

Testcase Test Result

Accepted Runtime: 0 ms

Case 1

Input

["MyStack","push","push","top","pop","empty"]

[[], [1], [2], [], [], []]

Output

[null,null,null,2,2,false]

5.Implement LRU Cache using Hash Table

DescriptionEditorialSolutionsAcceptedSubmissions

All Submissions

Accepted23 / 23 testcases passed

Prakhar Sharma submitted at Mar 19, 2025 15:22

EditorialSolution

Runtime

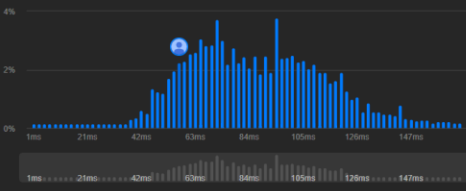
58 ms | Beat: 90.09%

Analyze Complexity

Memory

173.22 MB | Beat: 62.31%

Analyze Complexity



Code | C++

```
class LRUCache
{
public:
    list<pair<int, int>> l;
    unordered_map<int, list<pair<int, int>>::iterator> m;
    int size;
    LRUCache(int capacity)
    {
```

View more

More challenges

Code

C++Auto

```
30 {
31     if(m.find(key)!=m.end())
32     {
33         l.splice(l.begin(),l,m[key]);
34         m[key]->second=0;
35         return;
36     }
37     if(l.size()==size)
38     {
39         auto d_key=l.back().first;
40         l.pop_back();
41         m.erase(d_key);
42     }
43     l.push_front({key,value});
44     m[key]=l.begin();
45 }
```

SavedLn 35,C

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1

Input

["LRUCache","put","put","get","put","get","put","get","get","get"]

[[2],[1,1],[2,2],[1],[3,3],[2],[4,4],[1],[3],[4]]

Output

[null,null,null,1,null,-1,3,4]