

IMPLEMENTATION OF STACK USING QUEUE

```
struct queue{
    int d;
};

typedef struct {
    struct queue *q[100];
    int fr;
    int r;
}MyStack;

MyStack* myStackCreate() {
    MyStack *q1=(MyStack *)malloc(sizeof(MyStack));
    q1->fr=-1;
    q1->r=-1;
    return q1;
}

void push(MyStack* q1, struct queue* arr) {
    q1->q[++q1->r] = arr;
    if (q1->fr == -1) {
        q1->fr = 0;
    }
}

struct queueNode* pop(MyStack* q1) {
    struct queue* arr = q1->q[q1->fr];
    if (q1->fr == q1->r) {
        q1->fr = q1->r = -1;
    } else {
        q1->fr++;
    }
}
```

```

    }
    return arr;
}

void myStackPush(MyStack* obj, int x) {
    struct queue* qu= (struct queue*) malloc(sizeof(struct queue));
    qu->d = x;
    push(obj, qu);
    int size = obj->r - obj->fr + 1;
    while (size > 1) {
        struct queue* fr = pop(obj);
        push(obj, fr);
        size--;
    }
}

```

```

int myStackPop(MyStack* obj) {
    struct queue* front = pop(obj);
    int x = front->d;
    return x;
}

```

```

int myStackTop(MyStack* obj) {
    struct queue* front = obj->q[obj->fr];
    return front->d;
}

```

```

bool myStackEmpty(MyStack* obj) {
    return obj->fr == -1;
}

```

```
}
```

```
void myStackFree(MyStack* obj) {  
    free(obj);  
}
```

```
/**
```

* Your MyStack struct will be instantiated and called as such:

* MyStack* obj = myStackCreate();

* myStackPush(obj, x);

* int param_2 = myStackPop(obj);

* int param_3 = myStackTop(obj);

* bool param_4 = myStackEmpty(obj);

* myStackFree(obj);

```
*/
```

The screenshot shows a web browser window with multiple tabs. The active tab is LeetCode, displaying the submission page for the problem "Implement Stack using Queues". The submission status is "Accepted" with a runtime of 0 ms and memory usage of 6.60 MB. The code for the stack operations is visible in the editor, and the test case results show the stack operations being performed correctly.

LeetCode - The World's Leading
Github link E section - Google
Data Structures Lab/LeetCode
leetcode.com/problems/implement-stack-using-queues/submissions/1140540229/

Problem List
Dynamic Layout
Premium

Description Editorial Solutions Submissions

Accepted

Runtime: 0 ms
Beats 100.00% of users with C

Memory: 6.60 MB
Beats 75.07% of users with C

More challenges

232. Implement Queue using Stacks

Status Language Runtime Memory

Accepted a few seconds ago C 0 ms 6.6 MB

Accepted 3 hours ago C++ 3 ms 7.5 MB

Runtime Error 3 hours ago C++ N/A N/A

Accepted 3 hours ago C++ 3 ms 7.1 MB

Accepted 3 hours ago C++ 3 ms 7.2 MB

Testcase Result

Accepted Runtime: 0 ms

Case 1

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

Output: [[], [1], [2], [1], [1], [1]]

Expected: [null, null, null, 2, 2, false]

Console Run Submit

LeetCode - The World's Leading
Github link E section - Google
Data Structures-Lab/LeetCode

leetcode.com/problems/implement-stack-using-queues/submissions/1140540229/

Problem List<>Dynamic LayoutPremium

Accepted

Editorial+ Solution

Runtime

0 ms

Beats 100.00% of users with C

Memory

6.60 MB

Beats 75.07% of users with C

More challenges

232. Implement Queue using Stacks

| Status | Language | Runtime | Memory | Notes |
|---------------|----------|---------|--------|-------|
| Accepted | C | 0 ms | 6.6 MB | |
| Accepted | C++ | 3 ms | 7.5 MB | |
| Runtime Error | C++ | N/A | N/A | |
| Accepted | C++ | 3 ms | 7.1 MB | |
| Accepted | C++ | 3 ms | 7.2 MB | |

Shivraj K. Pujari

Jan 08, 2024 20:30

Details+ Solution

Runtime

0 ms

Beats 100%

Memory

6.6 MB

Beats 75.1%

Click the distribution chart to view more details

Related Tags

Select related tags0/5

```
struct queue{
    int q;
};

typedef struct {
    struct queue *q[100];
    int r;
    int f;
}MyStack;
```

ConsoleRunSubmit

21°C Cloudy

Search

ENG IN

20:52 08-01-2024