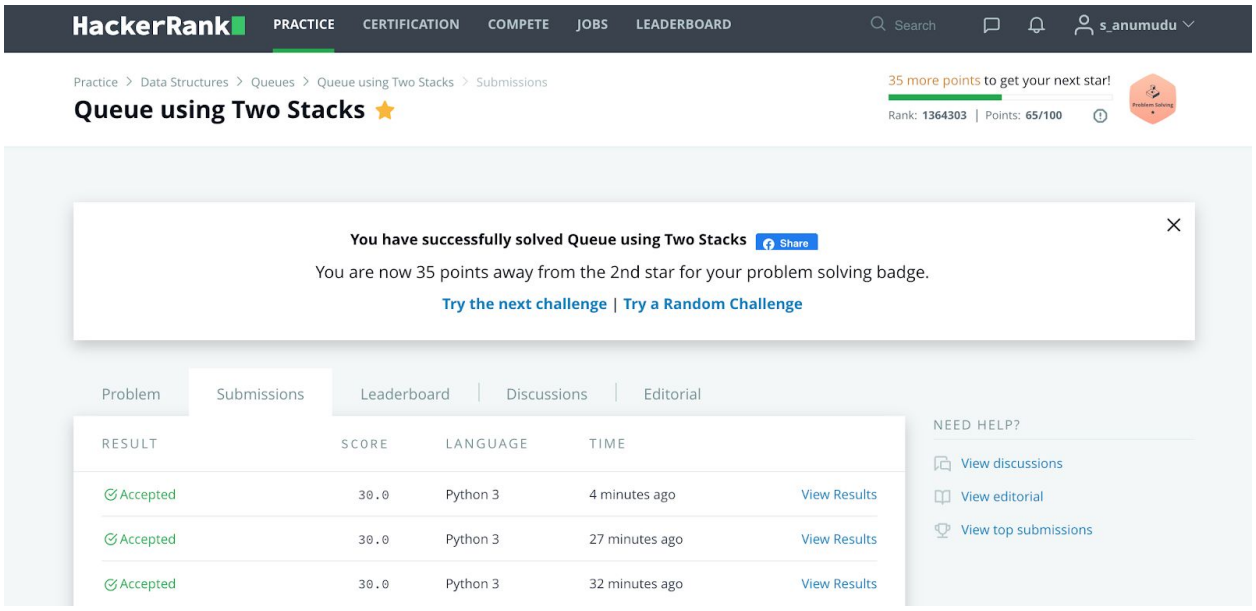
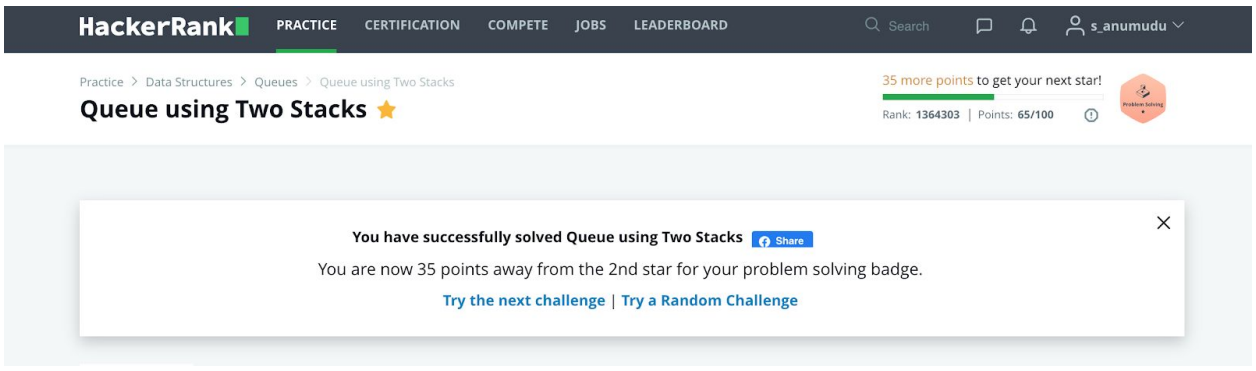
HackerRank Username: Samuel Anumudu / @s_anumudu

Screenshots





ick to go back, hold to see history

Change Theme | Python 3

```python
# Enter your code here. Read input from STDIN. Print output to STDOUT

# Get query input
query_input = int(input())

# initialize two stacks
instack = []
outstack = []

# iterate using a for loop
for i in range(query_input):
    # use a list to get a query type, split and store in a list
    t = list(input().split())
    # process each queries then enqueue
    if t[0] == '1':
        instack.append(t[1])

    # dequeue
    elif t[0] == '2':
        if not outstack:
            while instack:
                outstack.append(instack.pop())
        outstack.pop()

    #print front element
    else:
        if not outstack:
            while instack:
```

Line: 31 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Line: 31 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Earn a certificate in Problem Solving

Kudos on your progress! Take the HackerRank Skills Certification test and enrich your profile

Get Certified

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Input (stdin) Download

```
10
1 42
2
1 14
3
1 28
3
1 60
1 78
2
2
```

Expected Output Download

Code Snippet:

```
# Enter your code here. Read input from STDIN. Print output to STDOUT

# Get query input
query_input = int(input())

# initialize two stacks
instack = []
outstack = []

# iterate using a for loop
for i in range(query_input):
    # use a list to get a query type, split and store in a list
    t = list(input().split())
    # process each queries then enqueue
    if t[0] == '1':
        instack.append(t[1])

    # dequeue
    elif t[0] == '2':
        if not outstack:
            while instack:
                outstack.append(instack.pop())
        outstack.pop()

    #print front element
    else:
        if not outstack:
            while instack:
                outstack.append(instack.pop())
        print(outstack[-1])
```