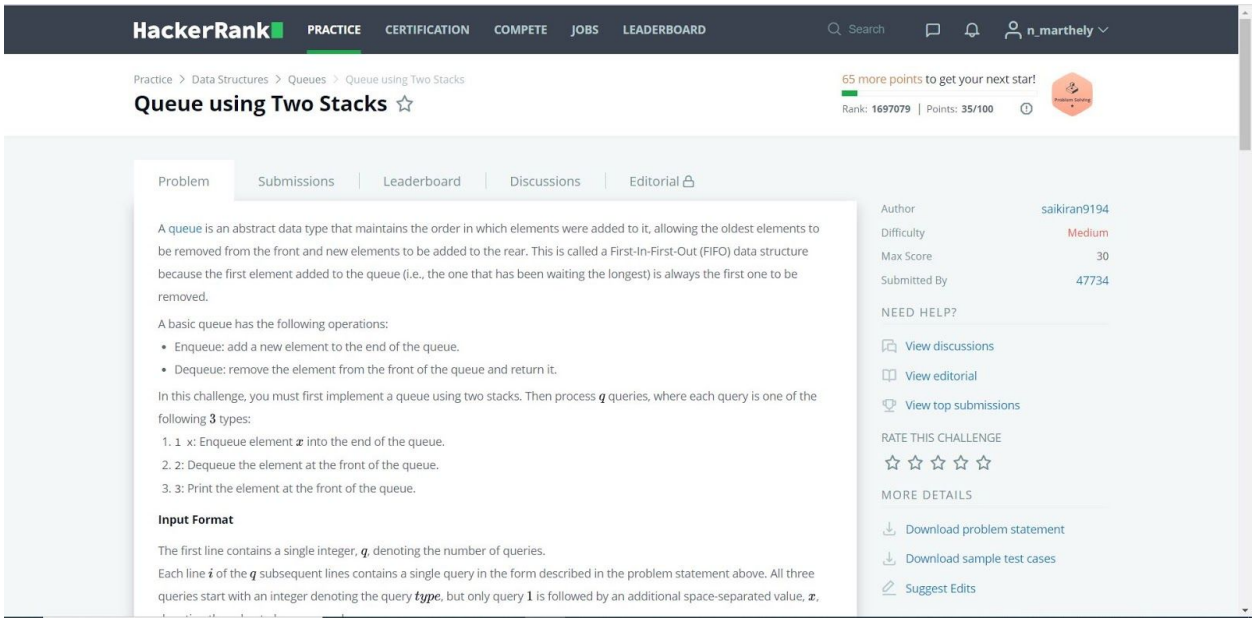
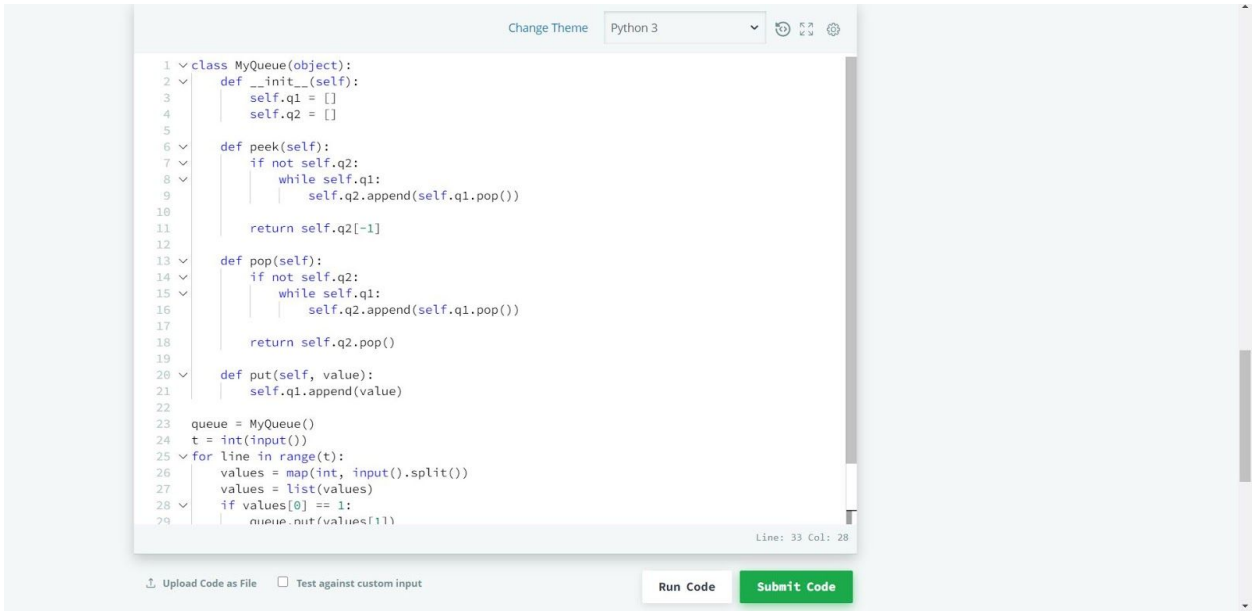
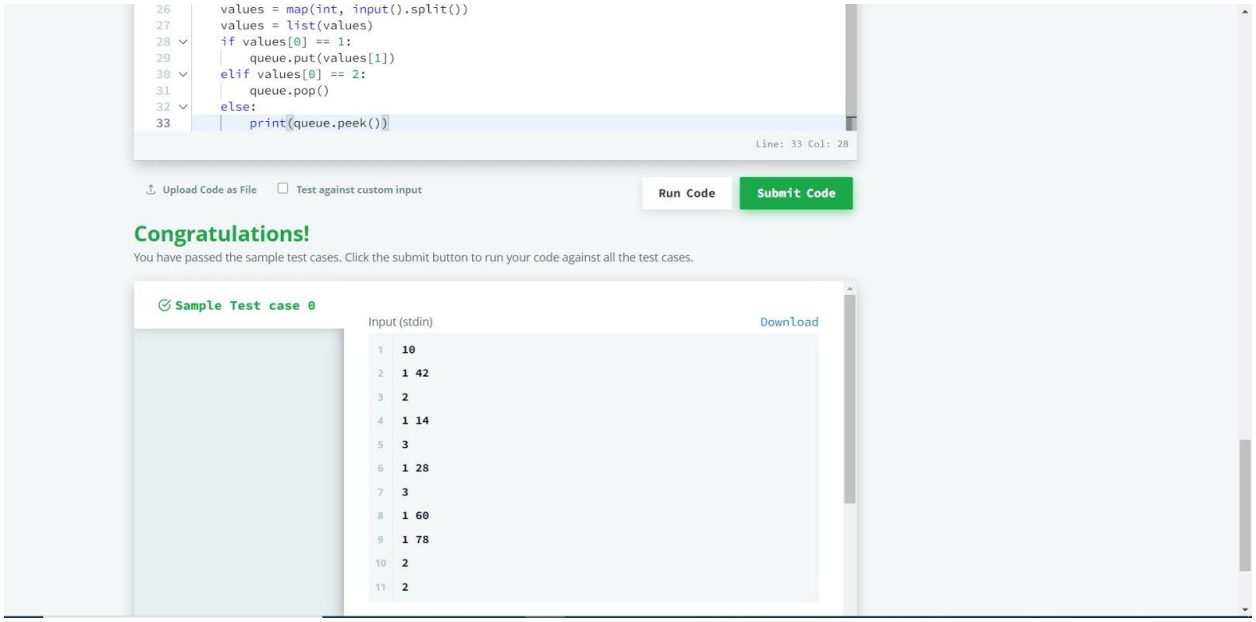
My HackerRank username: @n_marthely

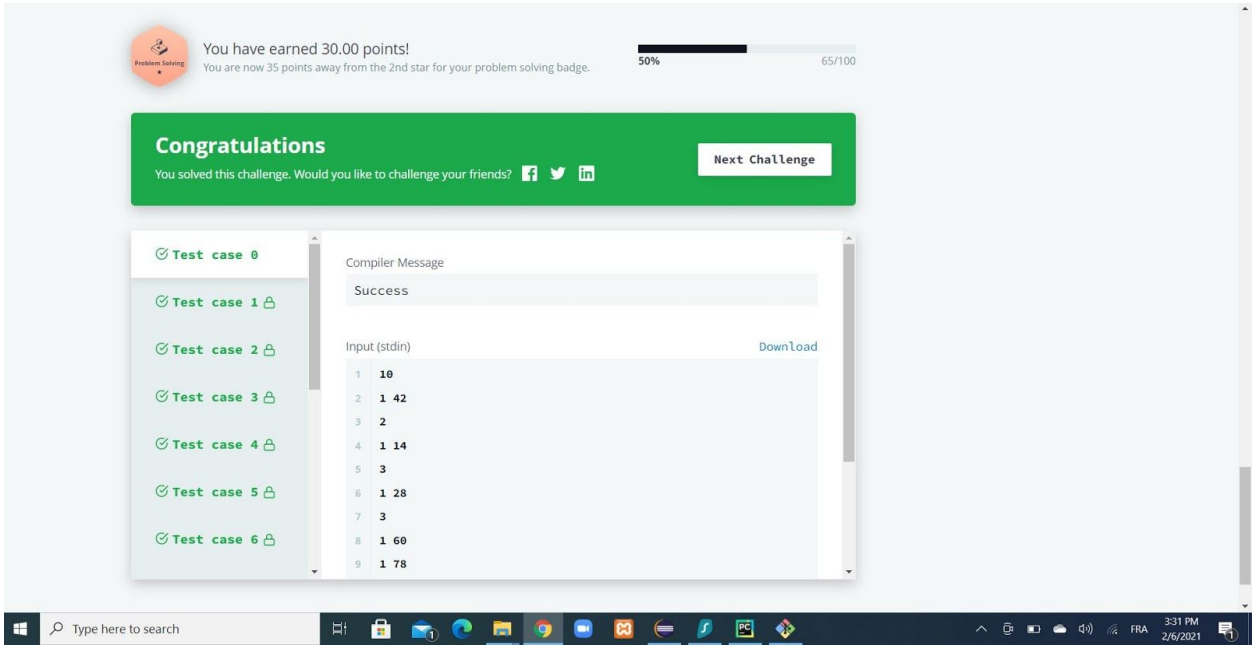
Before submitting my code:

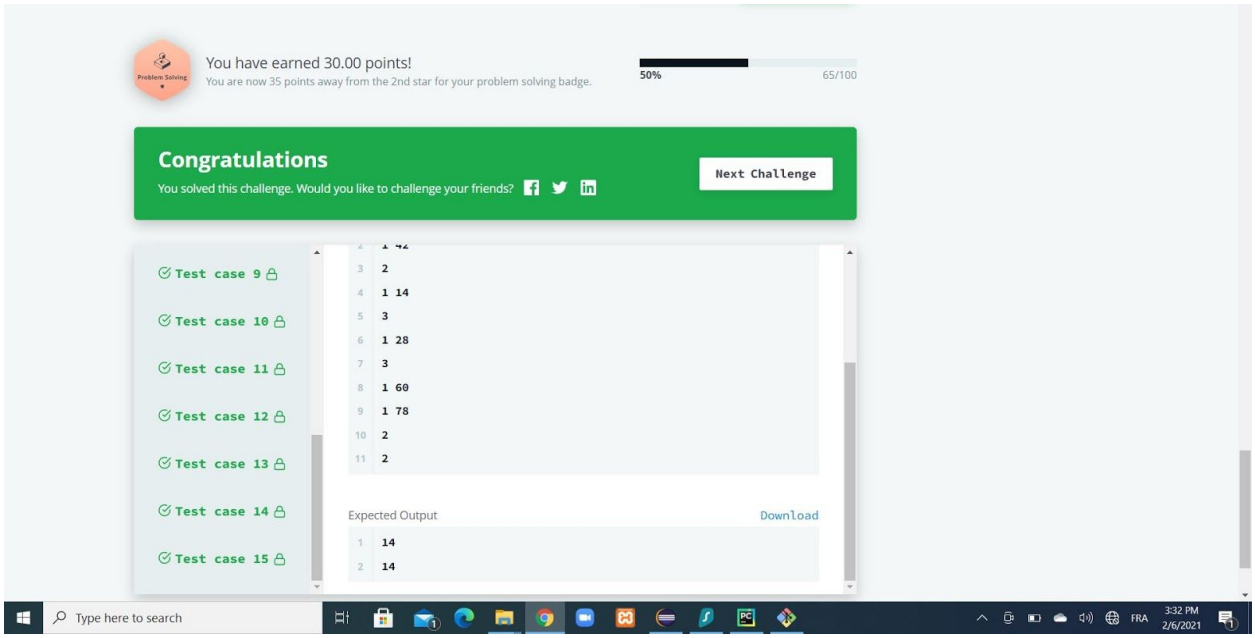


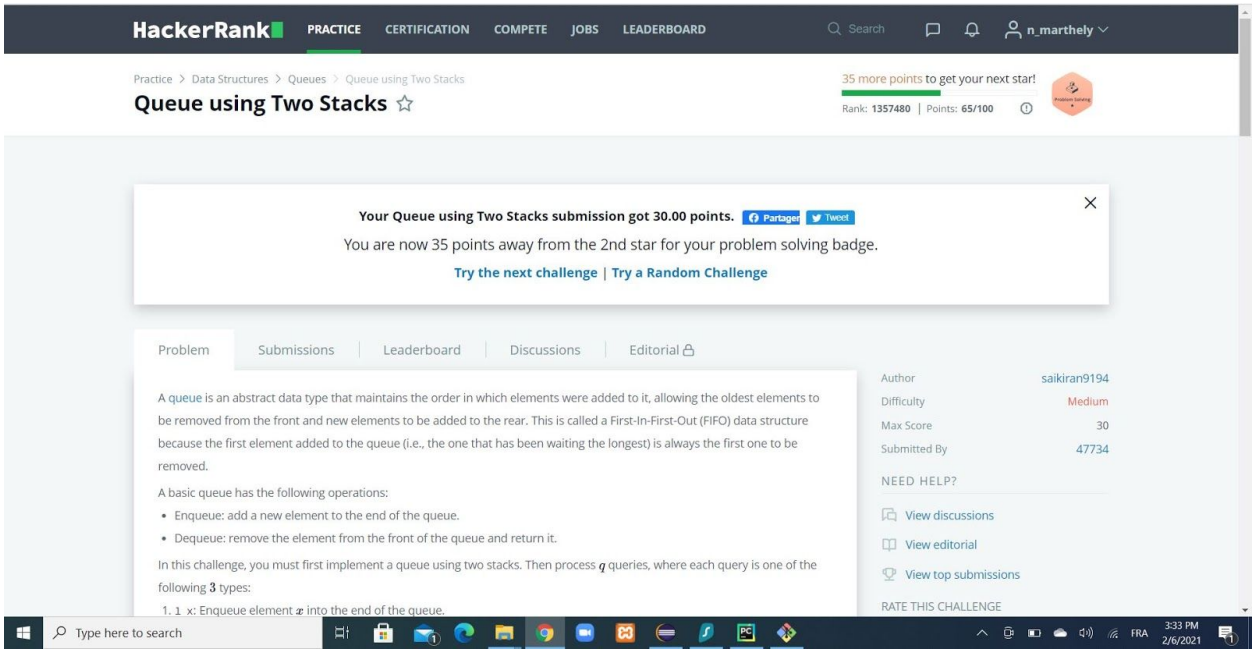


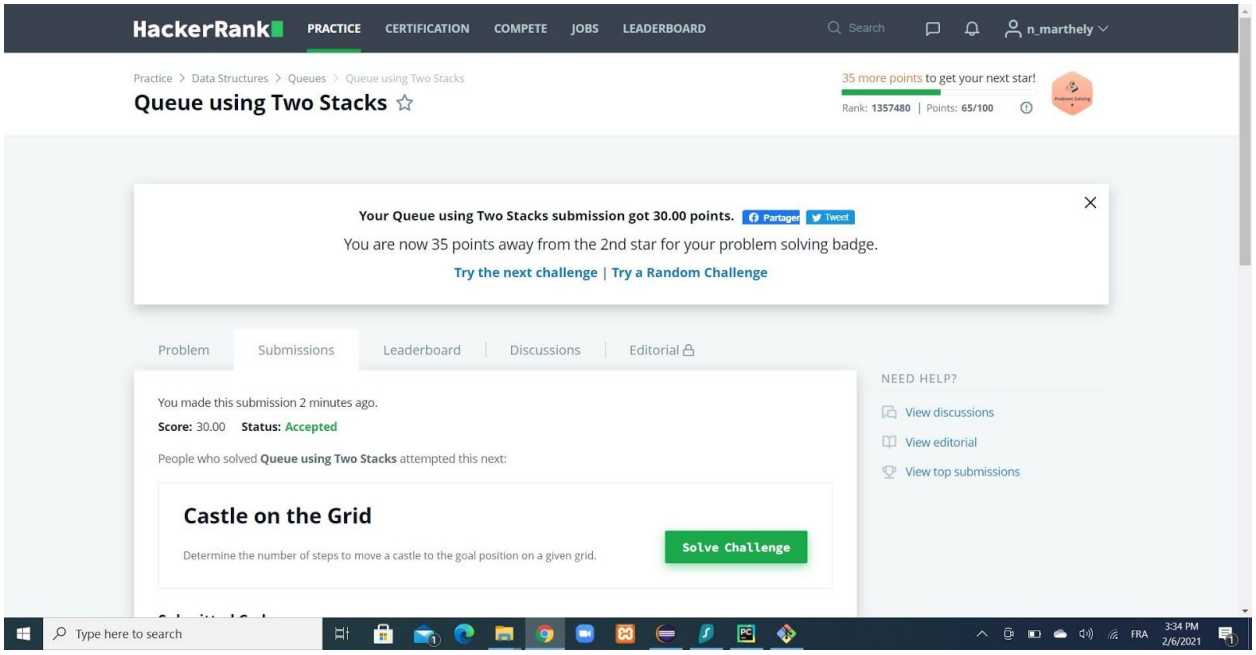


After submitting my code :




You have earned 30.00 points!
You are now 35 points away from the 2nd star for your problem solving badge.
50%
65/100
Congratulations
You solved this challenge. Would you like to challenge your friends?
Next Challenge
Test case 9
Test case 10
Test case 11
Test case 12
Test case 13
Test case 14
Test case 15
2
1 14
3
1 28
3
1 60
1 78
2
2
Expected Output
Download
14
14
Type here to search
FRA
3:32 PM
2/6/2021


HackerRank
PRACTICE
CERTIFICATION
COMPETE
JOBS
LEADERBOARD
Search
n_marthely
Practice > Data Structures > Queues > Queue using Two Stacks
Queue using Two Stacks
35 more points to get your next star!
Rank: 1357480 | Points: 65/100
Your Queue using Two Stacks submission got 30.00 points.
Partager
Tweet
You are now 35 points away from the 2nd star for your problem solving badge.
Try the next challenge | Try a Random Challenge
Problem
Submissions
Leaderboard
Discussions
Editorial
A queue is an abstract data type that maintains the order in which elements were added to it, allowing the oldest elements to be removed from the front and new elements to be added to the rear. This is called a First-In-First-Out (FIFO) data structure because the first element added to the queue (i.e., the one that has been waiting the longest) is always the first one to be removed.
A basic queue has the following operations:
Enqueue: add a new element to the end of the queue.
Dequeue: remove the element from the front of the queue and return it.
In this challenge, you must first implement a queue using two stacks. Then process q queries, where each query is one of the following 3 types:
1. 1 x: Enqueue element x into the end of the queue.
Author
saikiran9194
Difficulty
Medium
Max Score
30
Submitted By
47734
NEED HELP?
View discussions
View editorial
View top submissions
RATE THIS CHALLENGE
Type here to search
FRA
3:33 PM
2/6/2021



My code:

```
class MyQueue(object):
    def __init__(self):
        self.q1 = []
        self.q2 = []

    def peek(self):
        if not self.q2:
            while self.q1:
                self.q2.append(self.q1.pop())

        return self.q2[-1]

    def pop(self):
        if not self.q2:
            while self.q1:
                self.q2.append(self.q1.pop())

        return self.q2.pop()
```

```
    def put(self, value):
        self.q1.append(value)

queue = MyQueue()
t = int(input())
for line in range(t):
    values = map(int, input().split())
    values = list(values)
    if values[0] == 1:
        queue.put(values[1])
    elif values[0] == 2:
        queue.pop()
    else:
        print(queue.peek())
```