You can view this report online at : https://www.hackerrank.com/x/tests/1525161/candidates/50065012/report

Full Name:	Breeha Qasim
Email:	bq08283@st.habib.edu.pk
Test Name:	**CS 102 - Lab 5 - Spring 2023**
Taken On:	10 Feb 2023 10:31:43 PKT
Time Taken:	100 min 41 sec/ 5000 min
Work Experience:	< 1 years
Invited by:	Aisha
Skills Score:	
Tags Score:	

100%
300/300

scored in **CS 102 - Lab 5 - Spring 2023** in 100 min 41 sec on 10 Feb 2023 10:31:43 PKT

Recruiter/Team Comments:

No Comments.

Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review.

	Question Description	Time Taken	Score	Status
Q1	Priority Queue Implementation > Coding	34 min 16 sec	100/ 100	✓
Q2	HU - Help Desk > Coding	24 min 40 sec	100/ 100	!
Q3	Interleave the First Half of the queue with Second Half > Coding	16 min 24 sec	100/ 100	✓

QUESTION 1
✓ Correct Answer
Score 100

Priority Queue Implementation > Coding

QUESTION DESCRIPTION

Write an implementation of Priority Queue using List. The following methods of queue will be implemented:

1. `Enqueue(queue, item, priority):` will add the tuple `(item, priority)` to the `queue`. You should keep the queue in descending order of priority. If multiple items have same priority, their natural ordering will be retained.
2. `Dequeue(queue):` will return the item at the front of the `queue`. Since the queue is in sorted order, the highest priority item will be dequeued.

INTERVIEWER GUIDELINES

```
def is_empty(queue):
    return queue==[]

def front(queue):
    if (is_empty(queue) == True):
        return "Queue is empty"
    else:
        return queue[0]

def Dequeue(queue):
    if (is_empty(queue) == True):
        return "Queue is empty"
    else:
        return queue.pop(0)[0]

def Enqueue(queue, item, priority):
    if is_empty(queue):
        queue.append((item,priority))
    else:
        count = 0
        for _,i in queue:
            if i<priority:
                queue.insert(count,(item,priority))
                break
            elif i == priority:
                queue.insert(count+1,(item,priority))
                break
            count += 1
        if not((item,priority) in queue):
            queue.append((item,priority))
```

CANDIDATE ANSWER

Language used: **Python 3**

```
def Enqueue(queue, item, priority):
    for i in range(len(queue),0,-1):
        i=i-1
        if priority<=queue[i][1]:
            queue.insert(i+1,(item,priority))
            return
    queue.insert(0,(item,priority))


def Dequeue(queue):
    if len(queue)>0:
        item=queue[0][0]
        queue.pop(0)
        return item


```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	7	0.0399 sec	9.09 KB
Testcase 1	Easy	Sample case	Success	7	0.045 sec	9.07 KB
Testcase 2	Easy	Sample case	Success	7	0.0491 sec	9.02 KB
Testcase 3	Easy	Sample case	Success	7	0.0414 sec	9.11 KB
Testcase 4	Easy	Sample case	Success	7	0.0423 sec	8.96 KB

Testcase 5	Easy	Sample case	Success	7	0.0406 sec	9.2 KB
Testcase 6	Easy	Sample case	Success	7	0.0556 sec	9.16 KB
Testcase 7	Easy	Sample case	Success	8	0.0413 sec	8.82 KB
Testcase 8	Easy	Sample case	Success	7	0.0397 sec	9.02 KB
Testcase 9	Easy	Sample case	Success	8	0.1058 sec	9.05 KB
Testcase 10	Easy	Sample case	Success	7	0.0472 sec	8.9 KB
Testcase 11	Easy	Sample case	Success	7	0.0575 sec	8.8 KB
Testcase 12	Easy	Sample case	Success	7	0.0643 sec	8.86 KB
Testcase 13	Easy	Sample case	Success	7	0.0422 sec	9 KB

No Comments

QUESTION 2

Needs Review

Score 100

HU - Help Desk › Coding

QUESTION DESCRIPTION

You are building a system to log issues raised on HU Help Desk. The system adds all reported issues to a Priority Queue.

Insert the following issues on the queue:

Issue	Priority
AC Not working in Tariq Rafi	5
Password Change Issue	4
Need Installation on laptop	3
Need license	1
Lab PCs Setup	3
Login Issue	4

Retrieve all issues from the queue and print them in the order in which they will be addressed. The issues with highest priority will be processed first.
In the above case, the following output will be printed:

AC not working in Tariq Rafi
Password Change Issue
Login Issue
Need installation on laptop
Lab PCs Setup
Need license

CANDIDATE ANSWER

Language used: **Python 3**

```
1 def Enqueue(queue, item, priority):
2     for i in range(len(queue),0,-1):
3         i=i-1
```

```
        if priority<=queue[i][1]:
            queue.insert(i+1,(item,priority))
            return
    queue.insert(0,(item,priority))


def Dequeue(queue):
    if len(queue)>0:
        item=queue[0][0]
        queue.pop(0)
        return item
def is_empty(lst):
    if len(lst)==0:
        return True
    return False
issues=(["AC Not working in Tariq Rafi",5],["Password Change Issue",4],["Need
Installation on laptop",3],["Need license",1],["Lab PCs Setup",3],["Login
Issue",4])
empty=[]
for i in issues:
    Enqueue(empty,i[0],i[1])
    #print(empty)

while is_empty(empty)==False:
    print(Dequeue(empty))

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	100	0.038 sec	7.89 KB

No Comments

QUESTION 3

Correct Answer

Score 100

Interleave the First Half of the queue with Second Half › Coding

QUESTION DESCRIPTION

Given a queue of integers of even length, rearrange the elements by interleaving the first half of the queue with the second half of the queue.

If needed, you can additionally use a Stack for this problem. No other data structure can be used.

```
>>> InterLeaveQueue([1, 2, 3, 4],4)
[1, 3, 2, 4]

>>> InterLeaveQueue([11, 12, 13, 14, 15, 16, 17, 18, 19, 20],10)
[11, 16, 12, 17, 13, 18, 14, 19, 15, 20]
```

CANDIDATE ANSWER

Language used: **Python 3**

```python

def InterLeaveQueue(lst,length):
    lst1=[]
    lst2=[]
    for i in range(length):
        if i<length/2:
            lst1.append(lst.pop(0))
            #print(lst1)
        else:
            lst2.append(lst.pop(0))
            #print(lst2)
        output=[]
    for i in range(len(lst1)):
        output.append(lst1.pop(0))
        output.append(lst2.pop(0))
    return output


```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	25	0.0467 sec	9.25 KB
Testcase 1	Easy	Sample case	Success	25	0.1019 sec	9.02 KB
Testcase 2	Easy	Hidden case	Success	25	0.0448 sec	9.3 KB
Testcase 3	Easy	Hidden case	Success	25	0.0749 sec	9.03 KB

No Comments