

Introduction to Data Structure and Algorithm in PYTHON (QUEUES)

Activities for this lab:

- ▶ **Explain the concepts of Queue.**
- ▶ ***Queue creation and implementation***
- ▶ ***Lab exercise***

QUEUES

Queue is an ordered collection of items from which items may be deleted at one end (called front or head of the queue) and into which items may be inserted at the other end (called the rear end or tail of the queue). It is First-in-First-out (FIFO) type of data structure. Operations on queue are: Create Queue, insert items, remove items, display etc.

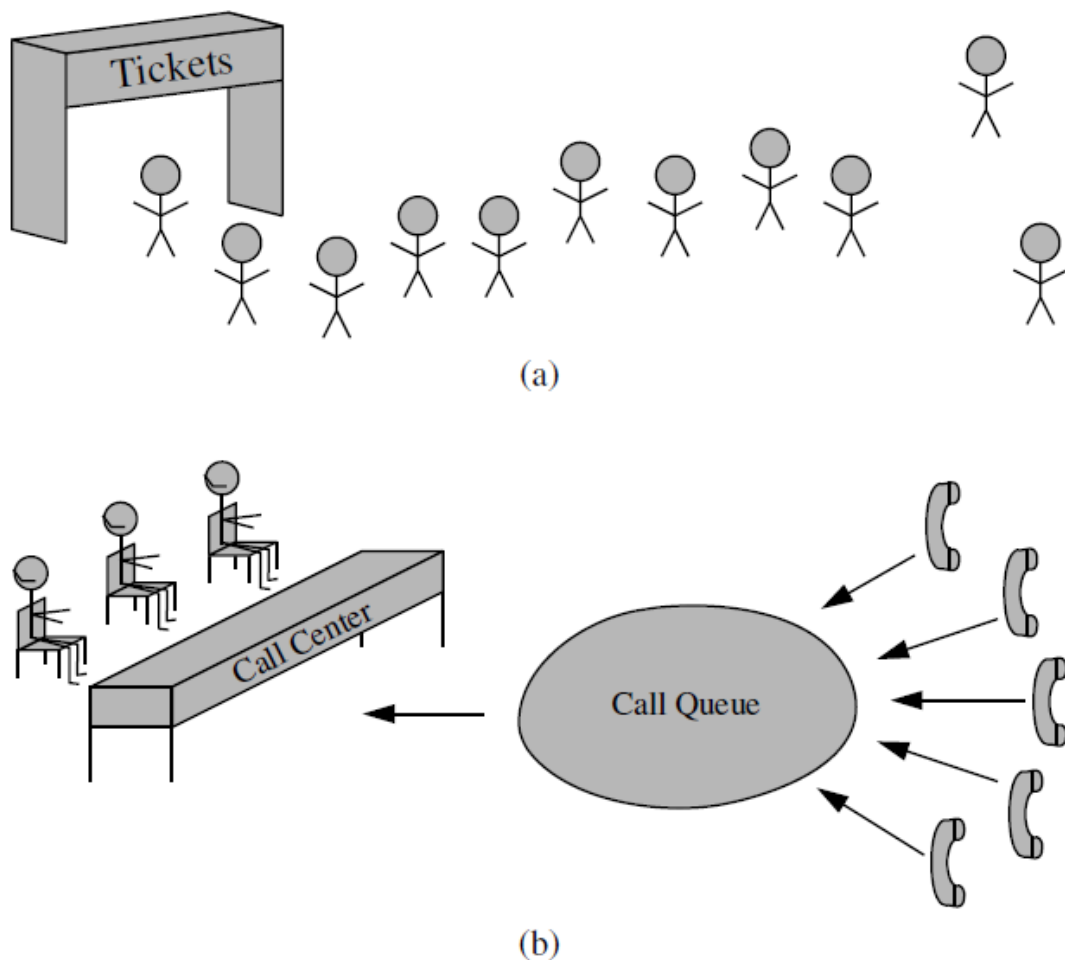


Figure 1:

Effects on an initially empty queue Q of integers

Operation	Return Value	first \leftarrow Q \leftarrow last
Q.enqueue(5)	–	[5]
Q.enqueue(3)	–	[5, 3]
len(Q)	2	[5, 3]
Q.dequeue()	5	[3]
Q.is_empty()	False	[3]
Q.dequeue()	3	[]
Q.is_empty()	True	[]
Q.dequeue()	“error”	[]
Q.enqueue(7)	–	[7]
Q.enqueue(9)	–	[7, 9]
Q.first()	7	[7, 9]
Q.enqueue(4)	–	[7, 9, 4]
len(Q)	3	[7, 9, 4]
Q.dequeue()	7	[9, 4]

AIM:

A PYTHON program to implement the Queue ADT using arrays and LL to perform all the queue operations

Algorithm for Implementation of Array Based Queue

1. Declare and initialize necessary variables, front = 0, rear = -1
2. For enqueue operation,
 - If rear \geq MAXSIZE - 1
print "Queue is full"
 - Else
 - Increment rear by 1 i.e. rear = rear + 1;
 - queue[rear] = item;
3. For next enqueue operation, go to step 2.
4. For dequeue operation

```

    If front > rear
        print "Queue is Empty"
    Else
        - item = queue[front]
        - increment front by 1 i.e. front = front + 1
5. For dequeue next data items, go to step 4.
6. Stop

```

Sample

INPUT / OUTPUT:

```

Enter the operation to be performed: 1)Enqueue 2)Dequeue 3)Display 4)Exit
1
Enter the number to be added 12
Enter the operation to be performed: 1)Enqueue 2)Dequeue 3)Display 4)Exit
1
Enter the number to be added 23
Enter the operation to be performed: 1)Enqueue 2)Dequeue 3)Display 4)Exit
1
Enter the number to be added 33
Enter the operation to be performed: 1)Enqueue 2)Dequeue 3)Display 4)Exit
1
Enter the number to be added 44
Enter the operation to be performed: 1)Enqueue 2)Dequeue 3)Display 4)Exit
2
The number to be deleted is 12
Enter the operation to be performed: 1)Enqueue 2)Dequeue 3)Display 4)Exit
3
The queue is 23 33 44

```

Operation	Running Time
Q.enqueue(e)	$O(1)^*$
Q.dequeue()	$O(1)^*$
Q.first()	$O(1)$
Q.is.empty()	$O(1)$
len(Q)	$O(1)$

Figure 2: Performance of an array-based implementation of a queue.
The space usage is $O(n)$, where n is the current number of elements in the queue.

Group Assignment

Use the linked list to implement Queue that will contain the following functions:

1. Add to Queue using input ()
2. Remove item from the Queue
3. Count Element in Queue
4. Show List of Elements in the Queue
5. Check for empty Queue
6. Exit

Submission on Sunday 15th October, 2017 before or by 11.59pm