SE(ECE)	Data Structures and Algorithms	
Experiment No: 8	Implement Queue using a) arrays and b) linked list.	Page: 1/6

Aim

To implement Queue operations (Enqueue, Dequeue, Peek, Display) using Array and Linked List.

Objectives

1. To understand the concept of Queue data structure (FIFO – First In First Out).
2. To implement queue operations using array-based implementation.
3. To implement queue operations using linked list-based implementation.
4. To compare array and linked list approaches for implementing queue.

Theory

Queue Overview

A **queue** is a linear data structure that follows the FIFO (First In First Out) principle.

- The element inserted first is the first one to be removed.
- Main operations:
  - **Enqueue:** Insert an element into the queue (at rear).
  - **Dequeue:** Remove the element from the queue (from front).
  - **Peek (Front):** Retrieve the element at the front without removing it.
  - **Display:** Show all elements in the queue.

Queue using Array

- Implemented with a fixed-size array**.**
- Two indices are used:
  - front → points to the first element.
  - rear → points to the last element.
- Advantages: Easy to implement.
- Limitations: Fixed size and wastage of space if not implemented circularly.

Algorithm (Array Implementation):

Enqueue(x):

1. Increment rear.
2. Insert element at arr[rear].

Dequeue():

1. Retrieve element from arr[front].
2. Increment front.

Peek():

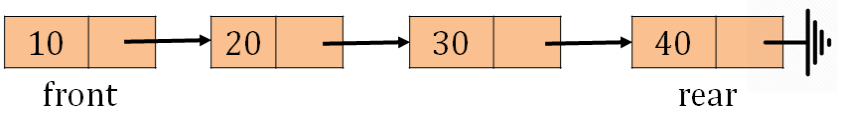
Return element at arr[front].

Display():

Traverse from front to rear.

Queue using Linked List

- Implemented dynamically using nodes**.**
- Two pointers are maintained:
  - front → points to the first node.
  - rear → points to the last node.



Algorithm (Linked List Implementation):

Enqueue(x):

1. Create a new node with value x.
2. If queue is empty, set front = rear = newNode.
3. Else, link rear->next to newNode and update rear.

Dequeue():

1. If front == NULL, queue underflow.
2. Store front->data.
3. Move front to front->next.
4. Delete old front node.
5. Return stored data.

Peek():

Return front->data.

Display():

Traverse from front to rear.

Applications of Queue

1. CPU Scheduling (Round-robin, FCFS).
2. Printer Queue (tasks handled one by one).
3. Breadth First Search (BFS) in graphs.
4. IO Buffers in operating systems.
5. Order processing systems in real life.

References

1. Yedidyah Langsam, Moshe J Augenstein, Aaron M Tenenbaum – Data structures using C and C++ - PHI Publications (2nd Edition).
2. Ellis Horowitz, Sataraj Sahni- Fundamentals of Data Structures – Galgotia Books source.

Questions

1. **A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as a ?**

   a) Queue
   b) Stack
   c) Tree
   d) Linked list

2. **The data structure required for Breadth First Traversal on a graph is?**

   a) Stack
   b) Array
   c) Queue
   d) Tree

3. **Let the following circular queue can accommodate maximum six elements with the following data**

   front = 2 rear = 4
   queue = _____; L, M, N, ___, ___
   What will happen after ADD O operation takes place?
   a) front = 2 rear = 5
   queue = _____; L, M, N, O, ___
   b) front = 3 rear = 5
   queue = L, M, N, O, ___
   c) front = 3 rear = 4
   queue = _____; L, M, N, O, ___
   d) front = 2 rear = 4
   queue = L, M, N, O, ___

4. **A queue is a?**

   a) FIFO (First In First Out) list
   b) LIFO (Last In First Out) list.
   c) Ordered array
   d) Linear tree

5. **In Breadth First Search of Graph, which of the following data structure is used?**

   a) Stack
   b) Queue
   c) Linked list
   d) None

6. **If the elements "A", "B", "C" and "D" are placed in a queue and are deleted one at a time, in what order will they be removed?**

   a) ABCD
   b) DCBA
   c) DCAB
   d) ABCD

7. **In linked list implementation of a queue, where does a new element be inserted?**

   a) At the head of link list
   b) At the tail of the link list
   c) At the centre position in the link list
   d) None

8. **In the array implementation of circular queue, which of the following operation take worst case linear time?**

   a) Insertion
   b) Deletion
   c) To empty a queue
   d) None

9. **In linked list implementation of queue, if only front pointer is maintained, which of the following operation take worst case linear time?**

   a) Insertion
   b) Deletion
   c) To empty a queue
   d) Both a) and c)

10. If the MAX_SIZE is the size of the array used in the implementation of circular queue. How is rear manipulated while inserting an element in the queue?

a) rear=(rear%1)+MAX_SIZE
b) rear=rear%(MAX_SIZE+1)
c) rear=(rear+1)%MAX_SIZE
d) rear=rear+(1%MAX_SIZE)

11. If the MAX_SIZE is the size of the array used in the implementation of circular queue, array index start with 0, front point to the first element in the queue, and rear point to the last element in the queue. Which of the following condition specify that circular queue is FULL?

a) Front=rear= -1
b) Front=(rear+1)%MAX_SIZE
c) Rear=front+1
d) Rear=(front+1)%MAX_SIZE

12. A circular queue is implemented using an array of size 10. The array index starts with 0, front is 6, and rear is 9. The insertion of next element takes place at the array index.

a) 0
b) 7
c) 9
d) 10

13. If the MAX_SIZE is the size of the array used in the implementation of circular queue, array index start with 0, front point to the first element in the queue, and rear point to the last element in the queue. Which of the following condition specify that circular queue is EMPTY?

a) Front=rear=0
b) Front= rear=-1
c) Front=rear+1
d) Front=(rear+1)%MAX_SIZE

14. A data structure in which elements can be inserted or deleted at/from both the ends but not in the middle is?

a) Queue
b) Circular queue
c) Dequeue
d) Priority queue

15. **In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into a NONEMPTY queue?**

    a) Only front pointer
    b) Only rear pointer
    c) Both front and rear pointer
    d) None of the front and rear pointer

16. **A normal queue, if implemented using an array of size MAX_SIZE, gets full when**

    a) Rear=MAX_SIZE-1
    b) Front=(rear+1)mod MAX_SIZE
    c) Front=rear+1
    d) Rear=front

17. **In linked list implementation of a queue, front and rear pointers are tracked. Which of these pointers will change during an insertion into EMPTY queue?**

    a) Only front pointer
    b) Only rear pointer
    c) Both front and rear pointer
    d) None

18. **An array of size MAX_SIZE is used to implement a circular queue. Front, Rear, and count are tracked. Suppose front is 0 and rear is MAX_SIZE -1. How many elements are present in the queue?**

    a) Zero
    b) One
    c) MAX_SIZE-1
    d) MAX_SIZE

Conclusion

A	C	O	T	Sign.