

MCQs

Q.1 Which of the following datastructure can be used to implement queue?

- a. Array
- b. Stack
- c. Linked List
- d. All of the above

Answer. All of the above

Q.2 Following code is an example of

```
if(this.isEmpty() === false) {  
    Let rv = this.data[this.front];  
    this.data[this.front] = 0  
    this.front = this.front + 1;  
    return rv;  
}  
}
```

- a. Enqueue
- b. Dequeue
- c. Insertion
- d. None of these

Answer. Dequeue

Q.3 How many pointers are needed while implementing queue using array?

- a. 1
- b. 2
- c. 3
- d. 4

Answer. 2

Q.4 Time complexity to perform dequeue operation in queue is

- a. $O(1)$
- b. $O(n)$
- c. $O(n^2)$
- d. $O(\log n)$

Answer. $O(1)$

Q.5 Insertion of an element inside the queue can happen from

- a. front
- b. end
- c. middle
- d. anywhere

Answer. end

Q.6 Time complexity to perform enqueue operation in queue is

- a. $O(1)$
- b. $O(n)$
- c. $O(n^2)$
- d. $O(\log n)$

Answer. $O(1)$

Q.7 Operation of adding the element from end is termed as

- a. Dequeue
- b. Enqueue
- c. Insertion
- d. Deletion

Answer. Enqueue

Q.8 Dequeue operation is not possible when the queue is already empty. Hence, leading to a condition of

- a. Overflow
- b. Underflow
- c. Error
- d. None of these

Answer. Underflow

Q.9 Insertion of an element inside the dequeue can happen from

- a. front
- b. end
- c. both front and end
- d. anywhere

Answer. both front and end

Q.10 A queue follows _____

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

Answer. FIFO (First In First Out) principle

Q.11 Which of the following statement is invalid

- a. RemoveBack method of the Deque is similar to the pop method of stack.
- b. Dequeue removeFront is similar to Queue's Deque method.
- c. Dequeue addBack is similar to Queue's Enqueue method.
- d. None of these

Answer. None of these

Q.12 Which of the following is a valid statement

- a. Dequeue removeFront is similar to Queue's Deque method.
- b. Dequeue removeEnd is similar to Queue's Deque method.
- c. Dequeue addFront is similar to Queue's Deque method.
- d. Dequeue addEnd is similar to Queue's Deque method.

Answer. Dequeue removeFront is similar to Queue's Deque method.

Q.13 Which of the following is a valid operation of deque

- a. addFront
- b. addBack
- c. removeFront
- d. All of the above

Answer. All of the above

Q.14 Stack + Queue can result in an implementation of

- a. array
- b. linkedlist
- c. deque
- d. None of these

Answer. Deque

Q.15 Which of the following is not the type of queue?

- a. Ordinary queue
- b. Single ended queue
- c. Circular queue
- d. Priority queue

Answer. Single ended queue

Q.16 How many queues are needed to implement a stack. Consider the situation where no other data structure like arrays, linked list is available to you.

- a. 1
- b. 2
- c. 3
- d. 4

Answer. 2

Q.17 Which of the following is true about linked list implementation of queue?

- a. In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.
- b. In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.
- c. Both of the above
- d. None of the above

Answer. Both of the above

Q.18 A queue is implemented using an array such that ENQUEUE and DEQUEUE operations are performed efficiently. Which one of the following statements is **CORRECT** (n refers to the number of items in the queue)?

- | | |
|---|--|
| a. Both operations can be performed in $O(1)$ time | b. At most one operation can be performed in $O(1)$ time but the worst case time for the other operation will be $\Omega(n)$ |
| c. The worst case time complexity for both operations will be $\Omega(n)$ | d. Worst case time complexity for both operations will be $\Omega(\log n)$ |

Answer. Both operations can be performed in $O(1)$ time

Q.19 How many stacks are needed to implement a queue. Consider the situation where no other data structure like arrays, linked list is available to you.

- | | |
|------|------|
| a. 1 | b. 2 |
| c. 3 | d. 4 |

Answer. 2

Q.20 Following is javascript like pseudo code of a function that takes a Queue as an argument, and uses a stack S to do processing.

```
function(queue){  
  
  let S; //Stack  
  
  while (!isEmpty(Q))  
  {  
  
    // deQueue an item from Q and push the dequeued item to S
```

- a. Removes the last from Q
- b. Keeps the Q same as it was before the call
- c. Makes Q empty
- d. Reverses the Q

Answer. Reverses the Q

Q.21 Consider the following pseudo code. Assume that `IntQueue` is an integer queue. What does the function `fun` do?

```
function fn(n)
{
    let q ; //Queue will contain only integer

    q.enqueue(0);

    q.enqueue(1);

    for (let i = 0; i < n; i++)
    {
```

- a. Prints numbers from 0 to n-1
- b. Prints numbers from n-1 to 0
- c. Prints first n Fibonacci numbers
- d. Prints first n Fibonacci numbers in reverse order.

Answer. Prints first n Fibonacci numbers

Q.22 In a linked list implementation following code is used for

```
if(this.head==null){  
    this.head=node;  
    return;  
}  
  
let temp = this.head;  
while(temp.next!=null)  
    temp=temp.
```

- a. searching
- b. deletion
- c. insertion
- d. None of these

Answer. insertion

Q.23 Which of the following is the correct syntax for the constructor of a linked list class?

- | | |
|--|---|
| a. "class ListNode {

constructor(data) {

this.data = data

this.next = data

}

}" | b. class ListNode {

constructor(data) {

data=data

this.next = null

}

} |
|--|---|

```
c. "class ListNode {  
    constructor(data) {  
        this.data = data  
        this.next = null  
    }  
}"
```

```
d. "class ListNode {  
    constructor(data) {  
        this.data = null  
        this.next = data  
    }  
}"
```

Answer. "class ListNode {
 constructor(data) {
 this.data = data
 this.next = null
 }
}"

Q.24 A normal queue, if implemented using an array of size MAX_SIZE, gets full when?

a. $\text{Front} = (\text{rear} + 1) \bmod \text{MAX_SIZE}$

b. $\text{Front} = \text{rear} + 1$

c. $\text{Rear} = \text{MAX_SIZE} - 1$

d. $\text{Rear} = \text{front}$

Answer. $\text{Rear} = \text{MAX_SIZE} - 1$

Q.25 What does the following code snippet of a linked list class do?

```
if(this.head == null || this.head.next == null)
```

```
    return this.head;
```

```
    let temp = this.head;
```

```
    while(temp.next != null){
```

```
        if(temp.data == temp.next.data)
```

```
            temp
```

- a. check duplicate element
- b. remove duplicate elements
- c. insert element
- d. delete node

Answer. remove duplicate elements

Q.26 Insertion of an element at the middle of a linked list requires the modification of how many pointers?

- a. 2
- b. 1
- c. 3
- d. 4

Answer. 2

Q.27

Insertion of an element at the ends of a linked list requires the modification of how many pointers?

- a. 2
- b. 1
- c. 3
- d. 4

Answer. 1

Q.28

In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element will be?

- a. $O(1)$
- b. $O(n)$
- c. $O(n^2)$
- d. $O(\log n)$

Answer. $O(\log n)$

Q.29 .

What is the time complexity to insert an element to the rear of a LinkedList(head pointer given)?

- a. $O(1)$
- b. $O(n)$
- c. $O(n^2)$
- d. $O(\log n)$

Answer. $O(n)$

Q.30 Polynomial addition can be implemented using which of the following datastructure?

- a. Linked List
- b. Queue
- c. Dequeue
- d. Stack

Answer. Linked List

Q.31 What is the time complexity of a program to reverse a linked list?

- a. $O(1)$
- b. $O(n)$
- c. $O(n^2)$
- d. $O(\log n)$

Answer. $O(n)$

Q.32

In which type of linked lists traversals can be performed in both directions?

- a. Singly linked list
- b. Doubly linked list
- c. Circular Linked List
- d. None of the above

Answer. Circular linked list

Q.33 .

Which of the following algorithm is the optimal way to find the middle element of the linked list?

- a. Fast and slow pointer method
- b. find length and then traverse two length/2
- c. get the middle node by finding the distance between all the nodes
- d. None of the above

Answer. Fast and slow pointer method

Q.34 What will be the output of the following code snippet for 1->2->3->4->5?

```
function solve () {  
    while(this.head != NULL) {  
        console.log(this.head.data);  
        this.head = this.head.next;  
    }  
}
```

- a. 1 2 3 4 5
- b. 5 4 3 2 1
- c. 1 3 5 2 4
- d. 2 4 1 3 5

Answer. 1 2 3 4 5

Q.35 What does the following code snippet do?

```
void solve(node) {  
    node = node -> next;  
    return node;  
}
```

- a. Delete the given node from linked list
- b. Delete the head of the list
- c. Delete the last node of the list
- d. None of the above

Answer. Delete the given node from linked list

Q.36

Which of the following problems can be solved using 2 pointers on linked list?

- a. Detection cycle in linked list
- b. Finding intersection of two linked list
- c. Finding middle element of the linked list
- d. All of the above

Answer. All of the above

Q.37 Which of the following linked list operation takes $O(1)$ time?

- a. Insert element at start of linked list
- b. Insert element at end of linked list
- c. Finding length of linked list
- d. None of the above

Answer. Insert element at start of linked list

Q.38 What does the following code does?

```
let data= []  
  
    let temp = this.head;  
  
while(temp!=null){  
  
    data.push(temp.data);  
  
    temp = temp.next;  
  
}  
  
console.log(...data);
```

- a. traverse the linked list
- b. prints the linked list
- c. add element to the linked list
- d. None of the above

Answer. prints the linked list

Q.39 Which of the following is not a type of Linked List ?

- a. Hybrid Linked List
- b. Circular Linked List
- c. Doubly Linked List
- d. Singly Linked List

Answer. Hybrid Linked List

Q.40 A linear list in which the pointer points only to the successive node is

- a. Circular Linked List
- b. Doubly Linked List

c. Singly Linked List

d. None of the above

Answer. Singly Linked List