# MCQs from Unit-3 (Linked Lists)

### 1. What is a linked list?

\*\*a)\*\* A linear collection of nodes

\*\*b)\*\* A non-linear collection of data

\*\*c)\*\* A stack-based structure

\*\*d)\*\* A recursive structure

\*\*Answer:\*\* a) A linear collection of nodes

### 2. What does a node in a linked list contain?

\*\*a)\*\* Data only

\*\*b)\*\* Address of the next node only

\*\*c)\*\* Data and pointer to the next node

\*\*d)\*\* None of the above

\*\*Answer:\*\* c) Data and pointer to the next node

### 3. What does the `NULL` pointer signify in a linked list?

\*\*a)\*\* Beginning of the list

\*\*b)\*\* End of the list

\*\*c)\*\* Middle of the list

\*\*d)\*\* A cyclic link

\*\*Answer:\*\* b) End of the list

### 4. Which of the following represents a self-referential structure?

\*\*a)\*\* Array

\*\*b)\*\* Linked list node

\*\*c)\*\* Queue

\*\*d)\*\* Stack

\*\*Answer:\*\* b) Linked list node

### 5. What is the advantage of linked lists over arrays?

\*\*a)\*\* Fixed size

\*\*b)\*\* Sequential memory allocation

\*\*c)\*\* Dynamic memory allocation

\*\*d)\*\* Faster random access

\*\*Answer:\*\* c) Dynamic memory allocation

### 6. What pointer is used to represent the first node of a linked list?

\*\*a)\*\* LAST

\*\*b)\*\* FIRST

\*\*c)\*\* START

\*\*d)\*\* HEAD

\*\*Answer:\*\* c) START

### 7. Which of the following is not a type of linked list?

\*\*a)\*\* Singly linked list

\*\*b)\*\* Circular linked list

\*\*c)\*\* Doubly linked list

\*\*d)\*\* Static linked list

\*\*Answer:\*\* d) Static linked list

### 8. How does a circular linked list differ from a singly linked list?

\*\*a)\*\* It has two pointers in each node

\*\*b)\*\* The last node points to the first node

\*\*c)\*\* It is stored in sequential memory

\*\*d)\*\* It is used only for stacks

\*\*Answer:\*\* b) The last node points to the first node

### 9. In a doubly linked list, what does each node contain?

\*\*a)\*\* Data and a pointer to the next node

\*\*b)\*\* Data and two pointers (next and previous)

\*\*c)\*\* Only data

\*\*d)\*\* Data and memory location

\*\*Answer:\*\* b) Data and two pointers (next and previous)

### 10. What is the main advantage of a doubly linked list over a singly linked list?

\*\*a)\*\* Easier deletion and insertion at any position

\*\*b)\*\* Requires less memory

\*\*c)\*\* Faster traversal

\*\*d)\*\* Dynamic memory allocation

\*\*Answer:\*\* a) Easier deletion and insertion at any position

### 11. What is a circular doubly linked list?

\*\*a)\*\* A list where the last node points to the first node

\*\*b)\*\* A list with two circular connections

\*\*c)\*\* A list with a node pointing only to the next node

\*\*d)\*\* A circular array representation

\*\*Answer:\*\* a) A list where the last node points to the first node

### 12. What is the condition for underflow in a linked list?

\*\*a)\*\* START is NULL

\*\*b)\*\* Memory overflow

\*\*c)\*\* The pointer points to the previous node

\*\*d)\*\* NEXT pointer points to the first node

\*\*Answer:\*\* a) START is NULL

### 13. Which operation adds a new node at the beginning of a linked list?

\*\*a)\*\* Deletion

\*\*b)\*\* Push

\*\*c)\*\* Prepend

\*\*d)\*\* Enqueue

\*\*Answer:\*\* c) Prepend

### 14. What data structure is used to implement a stack using linked lists?

\*\*a)\*\* Queue

\*\*b)\*\* Singly linked list

\*\*c)\*\* Doubly linked list

\*\*d)\*\* Binary tree

\*\*Answer:\*\* b) Singly linked list

### 15. What is the time complexity of inserting a node at the beginning of a singly linked list?

\*\*a)\*\* O(1)

\*\*b)\*\* O(n)

\*\*c)\*\* O(n^2)

\*\*d)\*\* O(log n)

\*\*Answer:\*\* a) O(1)

### 16. How does a circular doubly linked list differ from a simple doubly linked list?

\*\*a)\*\* It has more pointers

\*\*b)\*\* Its last node points to the first node

\*\*c)\*\* It requires sorted data

\*\*d)\*\* It cannot be traversed backwards

\*\*Answer:\*\* b) Its last node points to the first node

### 17. What happens when you delete a node in a linked list?

\*\*a)\*\* The entire list is deleted

\*\*b)\*\* Memory is freed and returned to the free pool

\*\*c)\*\* The linked list structure changes to an array

\*\*d)\*\* Nothing significant happens

\*\*Answer:\*\* b) Memory is freed and returned to the free pool

### 18. What is the purpose of the `AVAIL` pointer?

\*\*a)\*\* To track available memory for new nodes

\*\*b)\*\* To track the last node of the linked list

\*\*c)\*\* To track circular connections

\*\*d)\*\* To maintain the order of elements

\*\*Answer:\*\* a) To track available memory for new nodes

### 19. Which linked list type is best for implementing a queue?

\*\*a)\*\* Singly linked list

\*\*b)\*\* Doubly linked list

\*\*c)\*\* Circular singly linked list

\*\*d)\*\* None of the above

\*\*Answer:\*\* c) Circular singly linked list

### 20. How are sparse matrices represented in linked lists?

\*\*a)\*\* Using a row-column-value structure

\*\*b)\*\* Using a 2D array

\*\*c)\*\* By storing only non-zero values in random order

\*\*d)\*\* By creating a binary tree for elements

\*\*Answer:\*\* a) Using a row-column-value structure

### 21. What is the structure of a node in a sparse matrix using linked lists?

\*\*a)\*\* Data and two pointers

\*\*b)\*\* Row, column, value, and two pointers

\*\*c)\*\* Data and one pointer

\*\*d)\*\* Row, data, and column only

\*\*Answer:\*\* b) Row, column, value, and two pointers

### 22. What type of traversal is required for processing a circular linked list?

\*\*a)\*\* Sequential traversal until NULL is reached

\*\*b)\*\* Sequential traversal starting and ending at the same node

\*\*c)\*\* Level-order traversal

\*\*d)\*\* Depth-first traversal

\*\*Answer:\*\* b) Sequential traversal starting and ending at the same node

### 23. What operation is used to insert an element at the rear of a queue implemented with a linked list?

\*\*a)\*\* Push

\*\*b)\*\* Enqueue

\*\*c)\*\* Append

\*\*d)\*\* Prepend

\*\*Answer:\*\* b) Enqueue

### 24. What is the structure of a node in a stack implemented using linked lists?

\*\*a)\*\* Value and two pointers

\*\*b)\*\* Value and a pointer to the next node

\*\*c)\*\* Value only

\*\*d)\*\* Pointer to both top and bottom of the stack

\*\*Answer:\*\* b) Value and a pointer to the next node

### 25. How can the insertion of a new node be achieved in a doubly linked list?

\*\*a)\*\* Change only the next pointer

\*\*b)\*\* Change both the previous and next pointers

\*\*c)\*\* Delete the entire list and reinsert

\*\*d)\*\* Change only the previous pointer

\*\*Answer:\*\* b) Change both the previous and next pointers

### 26. Which linked list type is suitable for bidirectional traversal?

\*\*a)\*\* Singly linked list

\*\*b)\*\* Doubly linked list

\*\*c)\*\* Circular singly linked list

\*\*d)\*\* Queue

\*\*Answer:\*\* b) Doubly linked list

### 27. What happens to the `TOP` pointer in a stack when you perform a pop operation?

\*\*a)\*\* It is reset to NULL

\*\*b)\*\* It is decremented by 2

\*\*c)\*\* It points to the next element in the stack

\*\*d)\*\* It is incremented by 1

\*\*Answer:\*\* c) It points to the next element in the stack

### 28. What is the main characteristic of a circular singly linked list?

\*\*a)\*\* It is always sorted

\*\*b)\*\* It cannot contain more than 10 nodes

\*\*c)\*\* Its last node points back to the first node

\*\*d)\*\* Its first node points to the last node

\*\*Answer:\*\* c) Its last node points back to the first node

### 29. What is the drawback of using a singly linked list for implementing a stack?

\*\*a)\*\* Increased memory usage

\*\*b)\*\* Limited to forward traversal

\*\*c)\*\* High insertion time

\*\*d)\*\* Cannot store large datasets

\*\*Answer:\*\* b) Limited to forward traversal

### 30. How can you efficiently manage free space in memory for linked lists?

\*\*a)\*\* By using a `START` pointer

\*\*b)\*\* By using an `AVAIL` pointer to track free blocks

\*\*c)\*\* By deallocating all nodes

\*\*d)\*\* By creating a circular structure

\*\*Answer:\*\* b) By using an `AVAIL` pointer to track free blocks

---