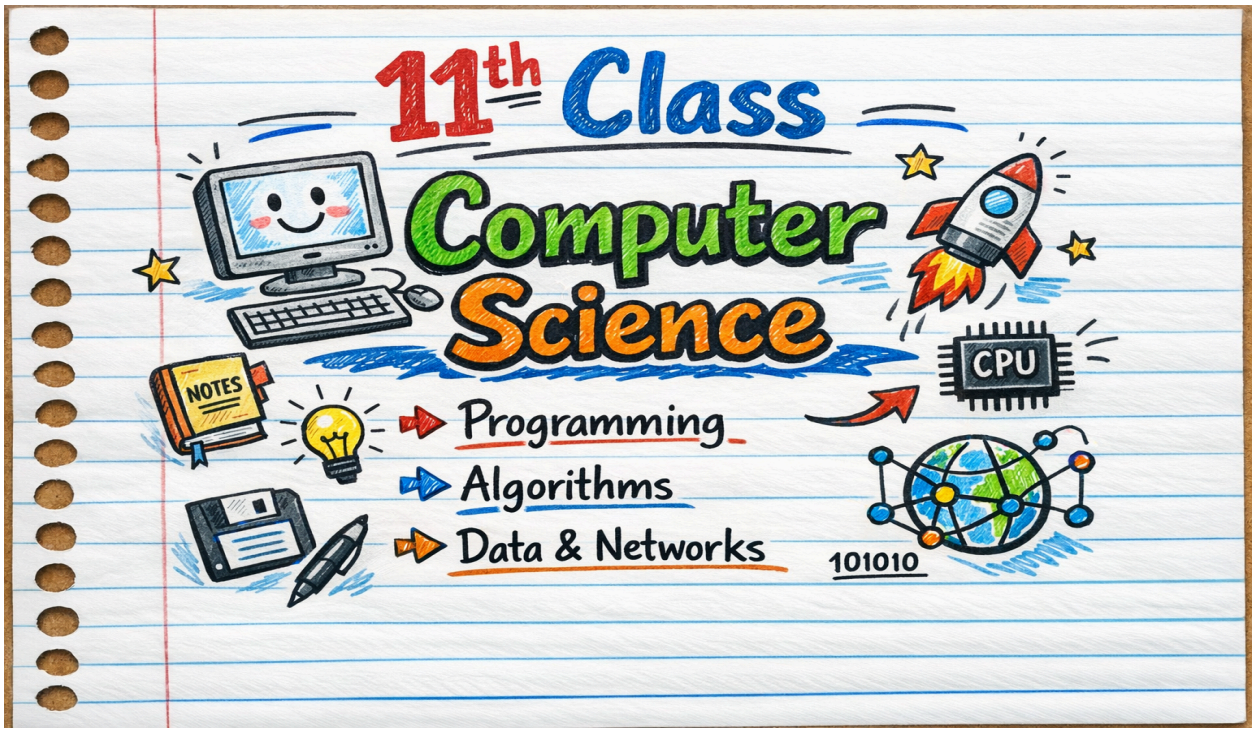

Class: 11th

Subject: Computer Science

Unit 4: Computational Structures

❖ Important MCQs:

1. Which statement best defines a Python list?

(a) A collection of unordered elements

(b) A sequence of indexed elements stored in order ✅

(c) A fixed-size memory structure

(d) A numeric variable

2. Which feature of a list allows efficient access to elements?

(a) Dynamic size

(b) Index-based access ✅

(c) Mutability

(d) Looping

3. Why are lists called dynamic structures?

(a) They are sorted automatically

(b) Their size can change during execution ✅

(c) They store only numbers

(d) They cannot be modified

4. What is the result of inserting an element at index 0?

(a) Element is added at end

(b) Element is added at middle

(c) Element is added at beginning ✅

(d) Element replaces last item

5. Which operation is most suitable for removing a specific value?

(a) pop()

(b) remove() ✅

(c) insert()

(d) append()

6. Which operation should be used when the index of element is known?

(a) remove()

(b) pop() ✅

(c) search()

(d) insert()

7. What is the key difference between remove() and pop()?

(a) remove() uses index, pop() uses value

(b) remove() uses value, pop() uses index ✅

(c) Both are same

(d) Both remove all elements

8. What is the primary purpose of using the in keyword?

(a) Insert element

(b) Delete element

(c) Check existence of element ✅

(d) Sort list

9. Which scenario best represents list usage?

(a) Storing a single value

(b) Managing a collection of items like tasks ✅

(c) Performing calculations

(d) Loop control

10. Why is order important in a list?

(a) For memory allocation

(b) For maintaining sequence of elements ✅

(c) For faster execution

(d) For sorting automatically

11. Which structure is most suitable for implementing LIFO behavior?

(a) Queue

(b) List

(c) Stack ✅

(d) Graph

12. Why is a stack called a restricted data structure?

(a) Limited memory

(b) Only one type of data allowed

(c) Operations are restricted to one end ✅

(d) Cannot grow dynamically

13. Which operation is performed when adding an element to a stack?

(a) Pop

(b) Push ✅

(c) Insert

(d) Remove

14. What happens when pop operation is performed on a stack?

(a) First element is removed

(b) Last inserted element is removed ✅

(c) Middle element is removed

(d) Entire stack is cleared

15. Which real-world situation follows stack behavior?

(a) Queue at bank

(b) Stack of plates/books ✅

(c) Railway system

(d) Tree structure

16. Why are lists used to implement stacks?

(a) They are fixed in size

(b) They support insertion and deletion at one end efficiently ✅

(c) They are immutable

(d) They are unordered

17. Which of the following best describes stack operations?

(a) Insert anywhere

(b) Delete anywhere

(c) Insert and delete only at top ✅

(d) Access random elements

18. What will happen if pop() is used on an empty stack?

(a) Returns None

(b) Returns 0

(c) Raises an error ✅

(d) Adds element

19. Which list property is most important for stack implementation?

(a) Ordered collection

(b) Dynamic size

(c) Ability to add/remove elements ✅

(d) Indexing

20. Which concept ensures the last added element is removed first?

(a) FIFO

(b) LIFO ✅

(c) Sequential access

(d) Random access

21. What is a queue data structure?

(a) A random collection of data

(b) A linear structure following FIFO principle ✅

(c) A hierarchical structure

(d) A fixed-size array

22. What does FIFO stand for?

(a) First In First Out ✅

(b) First Input Final Output

(c) Fast Input Fast Output

(d) Final In First Out

23. In a queue, where are elements added?

(a) Front

(b) Middle

(c) Back (rear) ✅

(d) Anywhere

24. In a queue, from where are elements removed?

(a) Back

(b) Middle

(c) Front ✅

(d) Anywhere

25. Which operation is used to add an element in a queue?

(a) Push

(b) Insert

(c) Enqueue ✅

(d) Append

26. Which operation is used to remove an element from a queue?

(a) Pop

(b) Delete

(c) Dequeue ✅

(d) Remove

27. Which real-life example best represents a queue?

(a) Stack of books

(b) Line at a bank counter ✅

(c) Tree structure

(d) Graph

28. What does the front element of a queue represent?

(a) Last added element

(b) First added element ✅

(c) Middle element

(d) Random element

29. Which operation allows checking the front element without removing it?

(a) Dequeue

(b) Enqueue

(c) Peek / Front retrieval ✅

(d) Insert

30. Which additional operation checks whether the queue has no elements?

(a) isFull()

(b) isEmpty() ✅

(c) clear()

(d) size()

31. What is a tree data structure?

(a) Linear sequence of elements

(b) Hierarchical structure of nodes ✅

(c) Random data structure

(d) Fixed array

32. What is the topmost node of a tree called?

(a) Leaf

(b) Parent

(c) Root node ✅

(d) Edge

33. What are the individual elements in a tree called?

(a) Edges

(b) Nodes ✅

(c) Roots

(d) Branches

34. What connects nodes in a tree?

(a) Links

(b) Paths

(c) Edges ✅

(d) Lines

35. A node with no children is called:

(a) Root

(b) Parent

(c) Leaf node ✅

(d) Branch

36. What does the height of a tree represent?

(a) Number of nodes

(b) Number of edges

(c) Longest path from root to leaf ✅

(d) Width of tree

37. When is a tree considered balanced?

(a) When it has many nodes

(b) When left and right branches have nearly equal height ✅

(c) When it has only one node

(d) When all nodes are equal

38. Why are trees used instead of lists for hierarchical data?

(a) Lists are faster

(b) Trees represent parent-child relationships clearly ✅

(c) Trees are smaller

(d) Lists cannot store data

39. Which application uses pre-order traversal?

(a) File deletion

(b) File backup (directories first) ✅

(c) Sorting

(d) Searching

40. Which traversal is used for deleting file systems correctly?

(a) Pre-order

(b) In-order

(c) Post-order ✅

(d) Level-order

41. What is a graph in data structures?

(a) A linear sequence

(b) A hierarchical structure

(c) A set of vertices connected by edges ✅

(d) A numeric array

42. What are vertices in a graph?

(a) Connections

(b) Nodes or points ✅

(c) Paths

(d) Weights

43. What do edges represent in a graph?

(a) Nodes

(b) Relationships between vertices ✅

(c) Data values

(d) Loops

44. Graphs are mainly used to represent:

(a) Linear data

(b) Hierarchical systems

(c) Networks of connections ✅

(d) Arrays

45. In a map of cities, cities are represented as:

(a) Edges

(b) Paths

(c) Vertices ✅

(d) Weights

46. In a map, roads between cities represent:

(a) Nodes

(b) Vertices

(c) Edges ✅

(d) Degrees

47. Which structure does NOT have a root node?

(a) Tree

(b) Graph ✅

(c) List

(d) Stack

48. Which statement is true about graphs?

(a) Always hierarchical

(b) Always linear

(c) Can form complex relationships ✅

(d) Have fixed structure

49. What is a key difference between tree and graph?

(a) Graph has root

(b) Tree has cycles

(c) Graph allows cycles ✅

(d) Tree has multiple paths

50. In a tree, between any two nodes there is:

(a) Multiple paths

(b) No path

(c) Exactly one path ✅

(d) Infinite paths

51. In a graph, between two vertices there can be:

(a) Only one path

(b) No path

(c) Multiple paths ✅

(d) Fixed path

52. What is the degree of a vertex?

(a) Number of nodes

(b) Number of edges connected to it ✅

(c) Number of paths

(d) Weight value

53. If a vertex connects to 4 edges, its degree is:

(a) 2

(b) 3

(c) 4 ✅

(d) 5

54. What does weight in a graph represent?

(a) Number of nodes

(b) Cost or distance between vertices ✅

(c) Number of edges

(d) Direction

55. Which graph has values like distance or cost on edges?

(a) Directed graph

(b) Undirected graph

(c) Weighted graph ✅

(d) Tree

56. What does direction in graphs indicate?

(a) Size of graph

(b) Flow of connection between vertices ✅

(c) Number of edges

(d) Weight of nodes

57. In a directed graph, edges are:

(a) Two-way

(b) One-way ✅

(c) Circular

(d) Random

58. In an undirected graph, edges are:

(a) One-way

(b) Two-way ✅

(c) Weighted only

(d) Directed only

59. Which graph allows movement in both directions?

(a) Directed graph

(b) Weighted graph

(c) Undirected graph ✅

(d) Tree

60. Which graph restricts movement to a specific direction?

(a) Undirected graph

(b) Weighted graph

(c) Directed graph ✅

(d) Tree

61. Which example represents a directed graph?

(a) Friendship network

(b) Two-way road

(c) One-way street system ✅

(d) Family tree

62. Which example represents an undirected graph?

(a) One-way road

(b) Friendship relationship ✅

(c) Traffic signals

(d) Decision tree

63. Which example represents a weighted graph?

(a) Social media

(b) Family tree

(c) Map with distances ✅

(d) Queue

64. Why are graphs more flexible than trees?

(a) They have roots

(b) They are linear

(c) They allow multiple connections and cycles ✅

(d) They are smaller

65. Which structure is best for hierarchical data?

(a) Graph

(b) Tree ✅

(c) Queue

(d) Stack

66. Which structure is best for network representation?

(a) List

(b) Tree

(c) Graph ✅

(d) Stack

67. Which of the following allows cycles?

(a) Tree

(b) List

(c) Graph ✅

(d) Stack

68. What type of graph shows cost or distance?

(a) Directed

(b) Undirected

(c) Weighted ✅

(d) Binary

69. In graphs, connections between vertices are called:

(a) Nodes

(b) Edges ✅

(c) Degrees

(d) Paths

70. Which of the following best describes a graph?

(a) Linear structure

(b) Hierarchical structure

(c) Network of interconnected nodes ✅

(d) Fixed array

❖ Important Short Questions:

1. What is a list?

A list is a built-in data structure in Python used to store multiple items in a single variable. These items can be of the same type or different types (numbers, strings, etc.). A list keeps elements in a specific order and allows easy access using indexes.

Example:

Python

```
items = ["pen", "book", "eraser", "scale"]

print(items)
```

2. How are lists created in Python?

Lists are created using square brackets []. Elements are written inside brackets and separated by commas. A list can be empty or can contain multiple values.

Example:

Python

```
numbers = [10, 20, 30, 40]

names = ["Ali", "Ahmed", "Sara"]

empty_list = []
```

3. What is indexing?

Indexing means accessing elements of a list using their position number. Each element has a unique index starting from 0. Indexing helps us retrieve or modify specific elements in a list.

Example:

Python

```
items = ["A", "B", "C", "D"]
```

```
print(items[0])  # A

print(items[2])  # C
```

4. What is the first index of a list?

The first index of a list is always 0 in Python. This means the first element is accessed using index 0, not 1.

Example:

Python

```
items = ["apple", "banana", "mango"]

print(items[0])  # apple
```

5. What is dynamic size?

Dynamic size means that a list in Python can increase or decrease in size during program execution. We can add new elements or remove existing ones without defining a fixed size.

Example:

Python

```
nums = [1, 2, 3]

nums.append(4)   # adding element

nums.remove(2)   # removing element

print(nums)      # [1, 3, 4]
```

6. What is ordered collection?

Ordered collection means that the elements in a list remain in the same order in which they are added. The position of each element is fixed unless changed manually.

Example:

Python

```
data = ["first", "second", "third"]

print(data)
```

7. Name function used to add element at position.

The insert() function is used to add an element at a specific position (index) in a list. It shifts existing elements to make space for the new item.

Example:

Python

```
items = ["A", "C", "D"]

items.insert(1, "B")

print(items)  # ['A', 'B', 'C', 'D']
```

8. Which function removes value from list?

The remove() function is used to delete the first occurrence of a specific value from a list. It does not use index; it uses the actual value.

Example:

Python

```
items = ["pen", "book", "eraser"]

items.remove("book")

print(items)  # ['pen', 'eraser']
```

9. What does pop() do?

The pop() function removes an element from a list. By default, it removes the last element, but you can also specify an index to remove a specific element.

Example:

Python

```
items = ["A", "B", "C"]

items.pop()    # removes C

print(items)   # ['A', 'B']
```

With index:

Python

```
items.pop(0)   # removes A
```

10. Which keyword is used for searching in list?

The in keyword is used to check whether an element exists in a list or not. It returns True if found, otherwise False.

Example:

Python

```
items = ["pen", "book", "eraser"]

if "book" in items:

    print("Item found")

else:

    print("Not found")
```

11. What is a stack?

A stack is a linear data structure in which elements are added and removed from one end called the top. It follows the LIFO (Last In First Out) principle.

Example:

Stack of books where the last book placed on top is removed first.

12. What does LIFO stand for?

LIFO stands for Last In First Out. It means the element that is inserted last will be removed first.

Example:

In a stack of plates, the last plate placed on top is the first one to be taken out.

13. What is push operation?

Push is an operation in a stack used to add an element at the top of the stack.

Example:

Python

```
stack = []
stack.append("Book A")  # push operation
print(stack)
```

14. What is pop operation?

Pop is an operation used to remove the top element from the stack.

Example:

Python

```
stack = ["Book A", "Book B"]
stack.pop()  # removes Book B
print(stack)
```

15. Where are stack operations performed?

All stack operations (push and pop) are performed at the top of the stack only.

Example:

In a stack of plates, both adding and removing plates happen from the top only.

16. What is a queue?

A queue is a linear data structure in which elements are arranged in a sequence. In a queue, insertion happens at one end called the rear, and deletion happens at the other end called the front. It follows the FIFO (First In First Out) principle.

Example:

A queue at a bus stop where the first person who joins the line is the first to get on the bus.

17. What does FIFO stand for?

FIFO stands for First In First Out. It means that the element which is inserted first into the queue will be removed first.

Example:

In a bank queue, the customer who arrives first will be served first.

18. What is enqueue?

Enqueue is an operation used in a queue to add a new element at the rear (back end) of the queue. It increases the size of the queue.

Example:

Python

```
queue = []

queue.append("Ali")   # enqueue operation

queue.append("Ahmed")

print(queue)
```

Explanation:

Here, new elements are added at the end of the queue.

19. What is dequeue?

Dequeue is an operation used to remove an element from the front of the queue. It decreases the size of the queue.

Example:

Python

```
queue = ["Ali", "Ahmed", "Sara"]

queue.pop(0)   # dequeue operation

print(queue)
```

Explanation:

The first element (Ali) is removed from the front.

20. Where are elements added in a queue?

Elements are always added at the rear (back end) of the queue using the enqueue operation.

Example:

In a line at a ticket counter, new people join at the end of the line.

21. From where are elements removed in a queue?

Elements are always removed from the front of the queue using the dequeue operation.

Example:

In a bank queue, the person at the front is served and leaves first.

22. What is a tree data structure?

A tree is a non-linear, hierarchical data structure in which data is organized in the form of nodes connected by edges. It starts from a single node called the root node, and branches into sub-nodes.

Example:

A family tree or file system structure in a computer.

22. What is a root node?

The root node is the topmost node of a tree. It is the starting point of the tree from which all other nodes are connected.

Example:

In a family tree, the oldest ancestor is the root node.

23. What is a leaf node?

A leaf node is a node that has no children (no further branches). It is the last level of a tree.

Example:

In a family tree, children with no further descendants are leaf nodes.

24. What are edges in a tree?

Edges are the lines or connections between nodes in a tree. They show the relationship between parent and child nodes.

Example:

In a family tree, the connection between parent and child represents an edge.

25. What is height of a tree?

The height of a tree is the longest path from the root node to the farthest leaf node. It represents how deep the tree is.

Example:

A tree with many levels of branches has greater height.

26. What is a balanced tree?

A balanced tree is a tree in which the left and right subtrees of every node have almost equal height. This makes operations more efficient.

Example:

A perfectly organized family tree where both sides have equal generations.

27. Name one application of trees.

One important application of trees is in file systems, where folders and files are arranged in a hierarchical structure.

Example:

Computer file explorer (C drive → folders → subfolders → files).

28. What is a graph?

A graph is a non-linear data structure consisting of a set of vertices (nodes) connected by edges. It is used to represent networks and relationships between objects.

Example:

A map of cities connected by roads or a social network.

29. What are vertices?

Vertices (nodes) are the individual points or objects in a graph. They represent entities like cities, people, or locations.

Example:

In a social network, each person is a vertex.

30. What are edges?

Edges are the connections between vertices. They represent relationships or links between nodes.

Example:

A friendship link between two people is an edge.

31. What is degree of a vertex?

The degree of a vertex is the number of edges connected to it.

Example:

If a city is connected to 3 other cities, its degree is 3.

32. What is a weighted graph?

A weighted graph is a graph in which each edge has a value (weight) such as distance, cost, or time.

Example:

Road map where each road has a distance like 10 km, 20 km, etc.

33. What is a directed graph?

A directed graph is a graph in which edges have a direction, meaning movement is allowed only in one direction.

Example:

One-way roads where traffic moves in only one direction.

34. What is an undirected graph?

An undirected graph is a graph in which edges have no direction, so movement is possible in both directions.

Example:

Friendship between two people (both are friends with each other).

35. Can graphs have cycles?

Yes, graphs can have cycles. A cycle occurs when you can start from a vertex and return to the same vertex through different edges.

Example:

A road network where you can travel in a loop and return to the starting city.

36. Give one example of a graph.

One example of a graph is a social network where people are connected through friendships.

37. What do edges represent in a graph?

Edges represent the relationship or connection between two vertices. They show how nodes are linked.

Example:

In a map, roads between cities represent edges.

Note:

This chapter is designed to provide a solid foundation of knowledge, with the goal of deepening understanding and encouraging further exploration of the subject. The content has been carefully selected to support effective learning and inspire students to engage with the topic more deeply.

Author: Muhammad Asghar

Purpose: To contribute to education by offering insightful, valuable content that enhances learning and understanding.