Comprehensive Viva Questions and Answers - Data Structures (KTU 2019 Syllabus)

Module 1: Basic Concepts

1. What is a Data Structure?

A data structure is a particular way of organizing and storing data in a computer so that it can be accessed and modified efficiently.

2. What are the types of Data Structures?

- Primitive: int, char, float, etc.
- Non-Primitive:
  - Linear: Arrays, Linked Lists, Stacks, Queues
  - Non-Linear: Trees, Graphs

3. What is an Algorithm?

An algorithm is a finite set of instructions that performs a specific task.

4. What is Time Complexity?

Time complexity is the computational complexity that describes the amount of time it takes to run an algorithm.

5. What is Big O Notation?

Big O is used to describe the upper bound of an algorithm in the worst-case scenario.

Module 2: Arrays, Stacks, and Queues

6. What is an Array?

An array is a collection of elements stored at contiguous memory locations.

7. What is a Stack?

A stack is a linear data structure which follows the LIFO (Last In First Out) principle.

8. Applications of Stack:

- Expression evaluation

- Syntax parsing

- Backtracking

9. What is a Queue?

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

10. What is a Circular Queue?

A circular queue connects the last position back to the first to make a circle.

Module 3: Linked Lists

11. What is a Linked List?

A linked list is a collection of nodes where each node contains data and a pointer to the next node.

12. Types of Linked Lists:

- Singly Linked List

- Doubly Linked List

- Circular Linked List

13. Advantages of Linked Lists:

- Dynamic size

- Efficient insertions/deletions

14. Applications of Linked Lists:

- Dynamic memory allocation

- Implementing stacks and queues

Module 4: Trees

15. What is a Tree?

A tree is a non-linear data structure consisting of nodes, with one node as the root and others as children.

16. What is a Binary Search Tree (BST)?

A BST is a tree in which each node has at most two children and the left child is smaller, the right is larger.

17. Tree Traversals:

- Inorder

- Preorder

- Postorder

18. What is an AVL Tree?

An AVL tree is a self-balancing binary search tree.

Module 5: Graphs

19. What is a Graph?

A graph is a collection of nodes (vertices) connected by edges.

20. Graph Representations:

- Adjacency Matrix
- Adjacency List

21. What is DFS?

Depth First Search is a graph traversal method using a stack (or recursion).

22. What is BFS?

Breadth First Search is a traversal method using a queue.

23. Applications of Graphs:

- Network routing
- Social networks

Module 6: Hashing and Searching

24. What is Hashing?

Hashing is a technique to convert a range of key values into a range of indexes.

25. Collision Resolution Techniques:

- Chaining
- Linear Probing

26. What is Linear Search?

Linear search checks each element one by one.

27. What is Binary Search?

Binary search divides the array into halves to find the target. It requires sorted input.

Module 7: Sorting

28. Sorting Algorithms:

- Bubble Sort
- Selection Sort
- Insertion Sort
- Merge Sort
- Quick Sort

29. Time Complexities:

Bubble: Best O(n), Worst O(n^2), Avg O(n^2)

Insertion: Best O(n), Worst O(n^2), Avg O(n^2)

Merge: O(n log n)

Quick: Best O(n log n), Worst O(n^2), Avg O(n log n)

Module 8: Applications and Programming

30. Applications of Arrays: Used in implementing other data structures like matrices.

31. Applications of Trees: Used in databases, file systems, and expression parsing.

32. Applications of Graphs: Shortest path problems, web page ranking.

33. Applications of Hashing: Lookup tables, databases, caches.

34. Write a program to reverse a linked list.

35. Write an algorithm for infix to postfix conversion.