**CS3301 – Data Structures**

**UNIT I – LISTS**

1. **Q:** What is an Abstract Data Type (ADT)?  
   **A:** An ADT is a model for data structures that defines the data and the operations on that data without specifying implementation details.
2. **Q:** Define List ADT.  
   **A:** List ADT is a collection of ordered elements where elements can be inserted, deleted, or retrieved by their position.
3. **Q:** What is the difference between a singly and doubly linked list?  
   **A:** A singly linked list has nodes pointing to the next node only, while a doubly linked list has nodes pointing to both next and previous nodes.
4. **Q:** What is a circular linked list?  
   **A:** In a circular linked list, the last node points back to the first node, forming a circle.
5. **Q:** Mention one application of a linked list.  
   **A:** Linked lists are used to implement dynamic memory allocation and symbolic expressions in compilers.
6. **Q:** What is a polynomial ADT?  
   **A:** Polynomial ADT represents polynomials as a list of terms with coefficients and powers for operations like addition and multiplication.
7. **Q:** Define Radix Sort.  
   **A:** Radix sort is a non-comparative sorting algorithm that sorts numbers digit by digit starting from the least significant digit.
8. **Q:** What is the main advantage of using linked lists over arrays?  
   **A:** Linked lists allow dynamic memory allocation and easy insertion/deletion.
9. **Q:** What is a multilist?  
   **A:** A multilist is a linked list in which each node may point to more than one list (used for representing complex structures like sparse matrices).
10. **Q:** How is array-based list implementation different from linked list implementation?  
    **A:** In array-based lists, memory is allocated statically and contiguously, whereas linked lists use dynamic, non-contiguous memory.

**UNIT II – STACKS AND QUEUES**

**10 Two-Marks Q&A**

1. **Q:** What is Stack ADT?  
   **A:** Stack ADT is a collection of elements with LIFO (Last In, First Out) behavior.
2. **Q:** List two operations of a stack.  
   **A:** Push (insertion) and Pop (deletion).
3. **Q:** Define Queue ADT.  
   **A:** Queue ADT is a collection of elements with FIFO (First In, First Out) behavior.
4. **Q:** What is a circular queue?  
   **A:** A circular queue connects the last position back to the first to make a circle for efficient memory use.
5. **Q:** What is DeQueue?  
   **A:** A DeQueue (Double-Ended Queue) allows insertion and deletion from both ends.
6. **Q:** What is the use of stacks in evaluating expressions?  
   **A:** Stacks are used in converting infix to postfix expressions and evaluating postfix expressions.
7. **Q:** What is balancing of symbols?  
   **A:** It refers to checking if all opening brackets in an expression have corresponding closing brackets, often using stacks.
8. **Q:** What is a common application of queues?  
   **A:** Queues are used in scheduling tasks, like print queues and CPU scheduling.
9. **Q:** What is meant by function calls using stacks?  
   **A:** Function call and return operations in programming languages use stack memory for storing return addresses and local variables.
10. **Q:** What is the difference between a stack and a queue?  
    **A:** Stack uses LIFO; queue uses FIFO for inserting and removing elements.

**UNIT III – TREES**

**10 Two-Marks Q&A**

1. **Q:** Define Tree ADT.  
   **A:** Tree ADT is a hierarchical data structure with a root node and subtrees of children represented as a set of linked nodes.
2. **Q:** What are the three types of binary tree traversals?  
   **A:** In-order, Pre-order, and Post-order traversals.
3. **Q:** What is a Binary Search Tree (BST)?  
   **A:** A BST is a binary tree where left child < root < right child for every node.
4. **Q:** Define AVL Tree.  
   **A:** AVL Tree is a self-balancing binary search tree where the height difference of left and right subtrees is at most 1.
5. **Q:** What is an Expression Tree?  
   **A:** An expression tree is a binary tree where internal nodes represent operators and leaves represent operands.
6. **Q:** Define a heap.  
   **A:** A heap is a complete binary tree where every parent node is greater (max heap) or smaller (min heap) than its children.
7. **Q:** What is a priority queue?  
   **A:** A priority queue is an abstract data structure where each element has a priority, and elements with higher priority are dequeued first.
8. **Q:** What is the time complexity of searching in a BST?  
   **A:** Average case: O(log n); Worst case (unbalanced): O(n).
9. **Q:** What is a balanced tree?  
   **A:** A tree is balanced if the height difference of left and right subtrees of every node is bounded (like in AVL Trees).
10. **Q:** What is the main advantage of AVL trees?  
    **A:** They maintain balance, ensuring efficient operations like insertion, deletion, and search.

**UNIT IV – MULTIWAY SEARCH TREES AND GRAPHS**

**10 Two-Marks Q&A**

1. **Q:** What is a B-Tree?  
   **A:** A B-Tree is a self-balancing search tree that maintains sorted data and allows searches, insertions, deletions in logarithmic time.
2. **Q:** What is a B+ Tree?  
   **A:** A B+ Tree is an extension of B-Tree where all values are found at leaf level and internal nodes only store keys.
3. **Q:** Define Graph ADT.  
   **A:** Graph ADT consists of a set of vertices and a set of edges connecting pairs of vertices.
4. **Q:** What are the two common graph representations?  
   **A:** Adjacency matrix and adjacency list.
5. **Q:** What is a connected graph?  
   **A:** A graph is connected if there is a path between every pair of vertices.
6. **Q:** Define Breadth-First Search (BFS).  
   **A:** BFS is a graph traversal method that visits all vertices level by level.
7. **Q:** What is an Euler circuit?  
   **A:** An Euler circuit is a path that starts and ends at the same vertex and visits every edge exactly once.
8. **Q:** What is Topological Sort?  
   **A:** It is the linear ordering of vertices of a Directed Acyclic Graph (DAG) such that for every directed edge u → v, u comes before v.
9. **Q:** What is Dijkstra’s Algorithm used for?  
   **A:** It is used to find the shortest path from a source vertex to all other vertices in a graph.
10. **Q:** Name two algorithms for Minimum Spanning Tree.  
    **A:** Prim’s algorithm and Kruskal’s algorithm.

**UNIT V – SEARCHING, SORTING AND HASHING TECHNIQUES**

**10 Two-Marks Q&A**

1. **Q:** What is Linear Search?  
   **A:** It checks each element one by one to find the target value.
2. **Q:** Define Binary Search.  
   **A:** It searches a sorted array by repeatedly dividing the search interval in half.
3. **Q:** What is Bubble Sort?  
   **A:** It repeatedly swaps adjacent elements if they are in the wrong order.
4. **Q:** What is Selection Sort?  
   **A:** It selects the smallest element and places it in its correct position in each iteration.
5. **Q:** What is Insertion Sort?  
   **A:** It builds the sorted array one element at a time by inserting elements into their correct positions.
6. **Q:** What is Shell Sort?  
   **A:** It is an optimization over insertion sort that compares elements far apart, reducing the gap over time.
7. **Q:** What is Merge Sort?  
   **A:** A divide-and-conquer algorithm that splits the list into halves, sorts, and merges them.
8. **Q:** What is Hashing?  
   **A:** Hashing is a technique to map data of arbitrary size to fixed-size values (hash values).
9. **Q:** What is Separate Chaining?  
   **A:** A method of handling hash collisions using linked lists at each bucket.
10. **Q:** What is Rehashing?  
    **A:** Rehashing is the process of resizing and reassigning values to a new hash table when load factor increases.

Let me know if you need 13-mark or 16-mark questions and answers for this subject too!