Question 1

Not yet answered

Marked out of 1.00

What is the maximum number of nodes in a binary tree of height `h` (where height is counted as the number of edges from root to the deepest node)?

- a. (2^h - 1)
- b. (2^{h+1} - 1)
- c. (h log h)
- d. (h^2)

Question 2

Not yet answered

Marked out of 1.00

Consider the following pseudo-code for a function `func(Node root)` applied to a binary tree. What does it compute? Function func(Node root): if root is NULL: return 0 return 1 + func(root.left) + func(root.right)

- a. Maximum depth of the tree
- b. Height of the tree
- c. Number of nodes in the tree
- d. Sum of all node values

Question 3

Not yet answered

Marked out of 1.00

Which of the following is always true for a full binary tree with `n` nodes?

- a. The height of the tree is always `log n`
- b. The tree is always balanced
- c. Every level is completely filled
- d. Every node has either 0 or 2 children

Question 4

Not yet answered

Marked out of 1.00

Given a BST, which of the following elements will always be found in the left subtree of a node with value `x`?

- a. All elements in the tree
- b. Elements less than `x`
- c. Elements equal to `x`
- d. Elements greater than `x`

Question 5

Not yet answered

Marked out of 1.00

What is the output of the following function when applied to a BST? Function findMin(Node root): if root is NULL: return NULL if root.left is NULL: return root.data return findMin(root.left)

- a. The height of the BST
- b. The maximum value in the BST
- c. The minimum value in the BST
- d. The sum of all nodes

Question 6

Not yet answered

Marked out of 1.00

What is the worst-case time complexity of deleting a node in an unbalanced BST with `n` nodes?

- a. O(n)
- b. O(log n)
- c. O(1)
- d. O(n log n)

Question 7

Not yet answered

Marked out of 1.00

Which of the following statements is true for Dijkstra's Algorithm?

- a. It finds the shortest path between all pairs of nodes
- b. It guarantees the shortest path in all cases
- c. It works only for graphs with non-negative weights
- d. It works correctly with negative-weight cycles

Question 8

Not yet answered

Marked out of 1.00

What is the time complexity of Depth-First Search (DFS) on a graph with `V` vertices and `E` edges using an adjacency matrix?

- a. O(E log V)
- b. $O(V^2)$
- c. O(V + E)
- d. O(V)

Question 9

Not yet answered

Marked out of 1.00

Which traversal method should be used to determine if a directed graph contains a cycle?

- a. Breadth-First Search (BFS)
- b. Depth-First Search (DFS) with recursion stack
- c. Kruskal's Algorithm
- d. Dijkstra's Algorithm

Question 10

Not yet answered

Marked out of 1.00

What is the output of the following function when applied to an undirected graph represented as an adjacency list? Function fun(Node start): Queue Q Add start to Q While Q is not empty: Node u = Q.dequeue() print u For each neighbor v of u: If v is not visited: Mark v as visited Add v to Q

- a. Detection of cycles
- b. Breadth-First Traversal
- c. Finding the minimum spanning tree
- d. Depth-First Traversal