Top 30 DSA Questions and Answers

1. What is an array?

An array is a collection of elements stored in contiguous memory locations, accessible by index.

2. What is a linked list?

A linked list is a linear data structure where each element (node) contains data and a pointer to the next node.

3. What is a stack?

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

4. What is a queue?

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

5. What is a binary tree?

A binary tree is a tree data structure where each node has at most two children: left and right.

6. What is a binary search tree (BST)?

A BST is a binary tree where the left subtree contains smaller values and the right subtree larger values.

7. What is a graph?

A graph is a collection of nodes (vertices) and edges that connect pairs of nodes.

8. What is recursion?

Recursion is a programming technique where a function calls itself to solve a smaller instance of the problem.

9. What is dynamic programming?

Dynamic Programming is a technique to solve problems by breaking them into overlapping subproblems and storing results.

10. What is the difference between BFS and DFS?

BFS explores nodes level by level, while DFS explores as far as possible along one branch before backtracking.

11. What is hashing?

Hashing is a technique to map data of arbitrary size to fixed-size values (hash codes) using a hash function.

12. What is a heap?

A heap is a complete binary tree that satisfies the heap property: parent is either greater or smaller

than its children.

13. What is a priority queue?

A priority queue is an abstract data type where each element has a priority, and the highest priority element is served first.

14. What is a trie?

A trie is a tree-like data structure used for storing strings where nodes represent common prefixes.

15. What is the time complexity of binary search?

The time complexity of binary search is O(log n).

16. What is the difference between merge sort and quick sort?

Merge sort divides the list into halves and merges sorted halves, while quick sort partitions around a pivot.

17. What is a circular linked list?

A circular linked list is one where the last node points back to the first node.

18. What is a doubly linked list?

A doubly linked list has nodes with two pointers, one pointing to the next node and another to the previous node.

19. What is big O notation?

Big O notation describes the upper bound of an algorithm's time or space complexity.

20. What is a greedy algorithm?

A greedy algorithm builds a solution step by step by choosing the locally optimal choice at each step.

21. What is backtracking?

Backtracking is a technique to solve problems recursively by trying partial solutions and undoing them if needed.

22. What is memoization?

Memoization is an optimization technique that stores the results of expensive function calls.

23. What is the difference between array and linked list?

Arrays have fixed size and direct access, while linked lists are dynamic but need sequential access.

24. What is a spanning tree?

A spanning tree is a subset of a graph that includes all vertices with minimum possible edges and no cycles.

25. What is Dijkstra's algorithm?

Dijkstra's algorithm finds the shortest path from a source node to all other nodes in a weighted

graph.

26. What is a minimum spanning tree (MST)?

An MST connects all vertices in a weighted graph with the minimum total edge weight and no cycles.

27. What is a topological sort?

Topological sort is a linear ordering of vertices in a directed acyclic graph such that for every directed edge (u, v), u comes before v.

28. What is the difference between BFS and Dijkstra's algorithm?

BFS is used for unweighted graphs, while Dijkstra's algorithm is used for weighted graphs.

29. What is a segment tree?

A segment tree is a binary tree used for storing intervals or segments and allows efficient range queries and updates.

30. What is the difference between recursion and iteration?

Recursion calls itself to solve subproblems, while iteration repeatedly executes a set of instructions using loops.