DSA with JavaScript - MCQs

Arrays & Searching

1. What is the time complexity of searching an element in an unsorted array using linear search?

   A) O(1) **B)** O(log N) **C)** O(N) **D)** O(N^2)

   Answer: C) O(N)

2. What is the time complexity of binary search?

   A) O(1) **B)** O(N) **C)** O(log N) **D)** O(N log N)

   Answer: C) O(log N)

3. Which search algorithm is used when the array is rotated but sorted?

   A) Linear Search **B)** Binary Search **C)** Jump Search **D)** Interpolation Search

   Answer: B) Binary Search

4. What is the best-case time complexity of binary search?

   A) O(1) **B)** O(N) **C)** O(log N) **D)** O(N log N)

   Answer: A) O(1)

Sorting Algorithms

5. Which sorting algorithm has the worst-case time complexity of O(N^2)?

   A) Merge Sort **B)** Quick Sort **C)** Bubble Sort **D)** Heap Sort

   Answer: C) Bubble Sort

6. What is the best-case time complexity of QuickSort?

   A) O(N) **B)** O(N log N) **C)** O(N^2) **D)** O(log N)

   Answer: B) O(N log N)

7. What is the worst-case time complexity of QuickSort?

   A) O(N) **B)** O(N log N) **C)** O(N^2) **D)** O(log N)

   Answer: C) O(N^2)

8. Which sorting algorithm is best for nearly sorted arrays?

   A) Quick Sort **B)** Heap Sort **C)** Bubble Sort **D)** Insertion Sort

   Answer: D) Insertion Sort

9. What is the space complexity of Merge Sort?

   A) O(1) **B)** O(N) **C)** O(log N) **D)** O(N log N)

   Answer: B) O(N)

Stacks & Queues

10. Which data structure follows Last In, First Out (LIFO)?

    A) Queue **B)** Stack **C)** Linked List **D)** Graph

    Answer: B) Stack

11. Which data structure follows First In, First Out (FIFO)?

    A) Queue **B)** Stack **C)** Tree **D)** Heap

    Answer: A) Queue

12. How can you implement a queue using two stacks?

    A) Using push and pop operations

    B) Using only pop operations

    C) Using shift and unshift

    D) Using priority queues

    Answer: A) Using push and pop operations

13. Which data structure is used in Breadth-First Search (BFS)?

    A) Stack **B)** Queue **C)** Priority Queue **D)** Set

    Answer: B) Queue

Linked Lists

14. What is the time complexity of inserting an element at the beginning of a singly linked list?

    A) O(1) **B)** O(N) **C)** O(log N) **D)** O(N log N)

    Answer: A) O(1)

15. How do you traverse a singly linked list?

    A) Using a loop

    B) Using recursion

    C) Using a queue

    D) Both A & B

    Answer: D) Both A & B

16. What is the best way to reverse a linked list?

    A) Using recursion

    B) Using an array

    C) Using a queue

    D) None of the above

    Answer: A) Using recursion

Recursion & Dynamic Programming

17. Which data structure is used to implement recursion?

    A) Queue **B)** Stack **C)** Heap **D)** Graph

    Answer: B) Stack

18. What is the base case in recursion?

    A) The initial step

    B) The final step

    C) The step where recursion stops

    D) The step where recursion starts

    Answer: C) The step where recursion stops

19. What is the time complexity of the Fibonacci sequence using recursion?

    A) O(N) **B)** O(N log N) **C)** O(2^N) **D)** $O(N^2)$

    Answer: C) O(2^N)

Hashing & Sets

20. Which JavaScript data structure allows only unique values?

    A) Array **B)** Map **C)** Set **D)** Object

    Answer: C) Set

21. Which JavaScript data structure stores key-value pairs?

    A) Array **B)** Set **C)** Map **D)** Stack

    Answer: C) Map

22. What is the time complexity of searching in a HashMap?

    A) O(1) **B)** O(N) **C)** O(log N) **D)** O(N log N)

    Answer: A) O(1)

Graph Algorithms

23. Which data structure is used for Depth-First Search (DFS)?

    A) Queue **B)** Stack **C)** Heap **D)** Priority Queue

    Answer: B) Stack

24. What is the time complexity of Dijkstra's Algorithm using a priority queue?

    A) O(N) **B)** O(N log N) **C)** $O(N^2)$ **D)** O(1)

    Answer: B) O(N log N)

Miscellaneous

25. Which JavaScript method is used to remove the last element from an array?

    A) push() **B)** pop() **C)** shift() **D)** unshift()

    Answer: B) pop()

26. Which technique is used in Binary Search Algorithm?

    A) Divide and Conquer **B)** Brute Force **C)** Dynamic Programming **D)** Greedy Algorithm

    Answer: A) Divide and Conquer

27. Which data structure is best for implementing an LRU Cache?

    A) Array **B)** Stack **C)** Queue **D)** HashMap with Doubly Linked List

    Answer: D) HashMap with Doubly Linked List

This list contains 27 questions so far. Would you like me to add more, or modify any specific sections?

😊

more mcqs in this porsn

Here are more MCQs for the **Stacks, Queues, Linked Lists, Recursion, Hashing, and Graph Algorithms** sections, making the total closer to 50 questions.

Stacks & Queues

28. Which operation is used to remove an element from the top of a stack?

    A) push() **B)** pop() **C)** shift() **D)** unshift()

    Answer: B) pop()

29. How can a queue be efficiently implemented?

    A) Using an array

    B) Using a linked list

    C) Using two stacks

    D) Both B and C

    Answer: D) Both B and C

30. Which of the following is not a queue operation?

    A) enqueue() **B)** dequeue() **C)** front() **D)** push()

    Answer: D) push()

31. Which data structure is used for the **Undo/Redo** feature in a text editor?

    A) Queue **B)** Stack **C)** Heap **D)** Tree

    Answer: B) Stack

32. How do you check if a stack is empty?

    A) `stack.length == 0`

B) `stack.size() == 0`

C) `stack.isEmpty()`

D) `stack.peek() == null`

Answer: A) `stack.length == 0`

33. Which operation in a circular queue requires updating both front and rear pointers?

    A) enqueue() **B)** dequeue() **C)** isEmpty() **D)** isFull()

    Answer: A) enqueue()

Linked Lists

34. What is the time complexity of inserting an element at the end of a singly linked list?

    A) O(1) **B)** O(N) **C)** O(log N) **D)** O(N^2)

    Answer: B) O(N)

35. How do you remove the last node of a singly linked list?

    A) By setting the second last node's next pointer to null

    B) By deleting the entire list

    C) By shifting all elements

    D) By reversing the list

    Answer: A) By setting the second last node's next pointer to null

36. Which of the following is true about a doubly linked list?

    A) Each node has a reference to only the next node

    B) Each node has references to both next and previous nodes

    C) It can only be traversed in one direction

    D) It has a fixed size

    Answer: B) Each node has references to both next and previous nodes

37. Which is an advantage of a linked list over an array?

    A) Constant-time access to elements

    B) Efficient memory usage

    C) Easier insertion and deletion

    D) Both B and C

    Answer: D) Both B and C

38. How do you check if a linked list contains a cycle?

    A) Using two pointers (slow and fast)

    B) Using recursion

    C) Sorting the list

    D) Traversing the list twice

    Answer: A) Using two pointers (slow and fast)

Recursion

39. What is the base condition in recursion?

    A) The step where recursion starts

    B) The step where recursion ends

    C) The first recursive function call

    D) The deepest level of recursion

    Answer: B) The step where recursion ends

40. What is the space complexity of a recursive function (without tail recursion)?

    A) O(N) **B)** O(log N) **C)** O(1) **D)** O(N^2)

    Answer: A) O(N)

41. What is tail recursion?

    A) A recursive function where the last operation is a recursive call

    B) A function that calls itself multiple times

    C) A function that does not use recursion

    D) A function that has multiple base cases

    Answer: A) A recursive function where the last operation is a recursive call

42. What is the main advantage of using recursion?

    A) Reduces code complexity

    B) Improves execution speed

    C) Uses less memory

    D) Always runs in constant time

    Answer: A) Reduces code complexity

Hashing & Sets

43. What is the average time complexity of retrieving an element from a JavaScript Set?

    A) O(1) **B)** O(N) **C)** O(log N) **D)** O(N log N)

    Answer: A) O(1)

44. Which of the following methods checks if an element exists in a JavaScript Set?

    A) `set.has(value)`

    B) `set.contains(value)`

    C) `set.exists(value)`

    D) `set.lookup(value)`

    Answer: A) `set.has(value)`

45. What is a Hash Collision?

    A) When two different keys map to the same value

    B) When two different values have the same key

    C) When a hash function runs indefinitely

    D) When a HashMap is empty

    Answer: A) When two different keys map to the same value

46. What is the best way to handle hash collisions?

    A) Chaining (Linked List)

    B) Linear Probing

    C) Quadratic Probing

    D) All of the above

    Answer: D) All of the above

Graph Algorithms

47. What is the time complexity of DFS (Depth-First Search) in an adjacency list?

    A) O(V + E) **B)** O(V^2) **C)** O(log V) **D)** O(V log V)

    Answer: A) O(V + E)

48. Which data structure is used for Kruskal's Algorithm?

    A) Heap **B)** Disjoint Set **C)** Stack **D)** Queue

    Answer: B) Disjoint Set

49. What is the best data structure to represent a weighted graph?

    A) Adjacency List **B)** Adjacency Matrix **C)** Incidence Matrix **D)** None of the above

    Answer: A) Adjacency List

50. Which graph traversal algorithm finds the shortest path in an unweighted graph?

    A) Depth-First Search (DFS)

B) Breadth-First Search (BFS)

C) Dijkstra's Algorithm

D) Bellman-Ford Algorithm

Answer: B) Breadth-First Search (BFS)