AOA

ONLY ADD QUESTIONS.

1. Determine time complexity
2. How is time complexity implemented
3. Time complexity of insertion sort
4. Master theorem
5. Linear loop eg
6. What is big O notation
7. If a function f(n) has time complexity O(n^2) then what it means
8. Divide and conquer + examples
9. Characteristic of quick sort and insertion sort
10. Konse types of sorting kiye
11. Sssp (single source shortest path)
12. Dijkstra algo
13. Bellman ford
14. Fractional knapsack
15. Relax procedure
16. Feasible and optimal solution
17. Greedy algo
18. Travelling salesman problem
19. Multistage graph
20. Dynamic approach
21. Complexity of 0/1 knapsack
22. What is source and sink
23. Outdegree of sink
24. TSP- types of approach
25. Time complexity of merge part of merge sort
26. Time complexity of recurrence problems
27. TYPES OF STRING STRING MATCHING ALGO
28. RABIN KARP ALGO
29. Planar node
30. n queen ka 8 by 8 matrix
31. Floyd warshall max paths in a graph of 4 vertices
32. Other than dynamic, which approach can be used for 0/1 knapsack problem
33. Difference between 0/1 knapsack and fractional knapsack
34. Difference between backtracking approach and Branch and bound
35. Difference between dynamic and backtracking
36. What is a spurious hit?
37. Np hard
38. Application of minimum spanning tree
39. Complexity Binary of

40. Memory optimization?
41. State space tree
42. Dynamic programming and its application?
43. job sequencing with deadline?
44. and working kruskal ka
45. Time complexity of kruskals
46. What is dijkstra and data structure used in dijkstras and how inputs are taken in dijktras
47. What is drawback of rabin karp
48. Branch and Bound
49. Why do we use branch and Bound
50. Worst case scenario for quick sort
51. State space tree of 15 puzzle problem
52. Planar graph
53. E node and live node
54. Time complexity of NP algos
55. Drawback of greedy approach

DBMS

ONLY ADD QUESTIONS

1. Characteristics of DBMS
2. DDL Commands
3. 1NF
4. What is transaction
5. What is transaction control
6. Types of relationship in DBMS
7. Difference between ER model and EER model?
8. Deadlock in DBMS
9. what is an entity, attributes, relation
10. how do you represent strong entity, what is inner join, types of join
11. Constraints with examples
12. Types of attributes and examples
13. Roles of Database Administrator
14. What is data abstraction
15. What are relationships
16. Differentiate between logical/physical schema
17. What is triggers
18. Architecture of DBMS?
19. Differentiate between ER and EER diagram
20. Normalisation (normal forms)
21. What is functional dependency
22. Difference between file system and dbms
23. Transitive dependency
24. Constraints in dbms
25. Serializability,recovery system
26. How to represent weak entity in er diagram
27. What is Col?concurrency Control
28. ACID properties
29. Which command is used to give permission to the user
30. What does the commit command do
31. Advantages of DBMS
32. What are DML commands and its syntax
33. What is the symbol of Project operation in relational algebra
34. What is groupby
35. Different types of users
36. Deadlock and deadlock prevention techniques
37. Log based Recovery

38. What is Naive users
39. What is entities and its type
40. Serializability
41. BCNF , 3NF , Transitive dependency
42. What is candidate key
43. What are types of user
44. What is data dependency
45. What is functional dependency
46. Role of DBA
47. 2NF
48. Attributes
49. serializability
50. Normalization - its type , explain bcnf
51. What is triggers
52. What are advantages and disadvantages of dbms
53. Components of DBMS
54. Disadvantages of dbms
55. (5th module - normalization and 6th module - Transaction ache se padho..sidha wahi se puchti hai mam -.-)
56.

OS

ONLY ADD QUESTIONS

1. What is OS?
2. Banker's Algorithm
3. Memory management
4. Demand paging
5. What is critical section
6. How to solve the critical section problems
7. What is semaphore
8. Types of semaphore
9. Deadlock conditions
10. How to solve circular wait prob
11. PCB
12. Context switching
13. Steps in context switching
14. Types of schedulers
15. Scheduling Algorithms
16. Process and program under execution
17. Name of the OS modules
18. When the producer and consumer are blocked
19. Fragmentation
20. Internal and external fragmentation
21. State process
22. Contiguous and non contiguous memory location with advantage and disadvantage
23. Overview of operating system
24. Goals of OS
25. Page replacement algorithm
26. thrashing
27. Types of OS
28. Disk scheduling
29. Paging and page table
30. What is semaphore
31. What is deadlock
32. CPU bound
33. Real Time OS and Batch OS
34. Scheduling Criterias
35. What is process
36. Burst time
37. Process and its attributes
38. Synchronisation
39. Critical section

SOME AOA ANSWER...(***Might not be correct***)

2. How is time complexity implemented
-->Time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input.

.
5. Linear loop eg
-->A linear loop that contains no other loops, and whose control variable is modified only via additive operations (addition or subtraction), a loop whose execution is directly a function of the number of elements being processed.

6. What is big O notation
-->Big O notation is a mathematical notation that describes the limiting behavior of a function when the argument tends towards a particular value or infinity.

7. If a function f(n) has time complexity O(n^2) then what it means
-->O(n) represents the complexity of a function that increases linearly and in direct proportion to the number of inputs. Similarly O(n^2) means the time taken by the function increases proportional to n^2 as the input size increases.

8. Divide and conquer + examples
--> divide-and-conquer algorithm recursively breaks down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. Merge Sort.

15. Relax procedure
-->Relaxing an edge, (a concept you can find in other shortest-path algorithms as well) is trying to lower the cost of getting to a vertex by using another vertex. where est(S,a) is the current estimate of the distance, and dist(a,b) is the distance between two vertices that are neighbors in the graph.

16. feasible and optimal solution
-->A feasible solution satisfies all the problem's constraints. An optimal solution is a feasible solution that results in the largest possible objective function value when maximizing (or smallest when minimizing).

20. Dynamic approach

-->Dynamic programming approach is similar to divide and conquer in breaking down the problem into smaller and yet smaller possible sub-problems. But unlike, divide and conquer, these sub-problems are not solved independently. Rather, results of these smaller sub-problems are remembered and used for similar or overlapping sub-problems. Mostly, these algorithms are used for optimization.
The following computer problems can be solved using dynamic programming approach –
All pair shortest path by Floyd-Warshall
Shortest path by Dijkstra
Knapsack problem

22. What is source and sink
-->A node is considered a source in a graph if it has in-degree of 0 (no nodes have a source as their destination); likewise, a node is considered a sink in a graph if it has out-degree of 0 (no nodes have a sink as their source).

23. Outdegree of sink
-->a node is considered a sink in a graph if it has out-degree of 0 (no nodes have a sink as their source).

24. TSP- types of approach
-->The Brute-Force Approach (Naive Approach)
Dynamic Programming, The Branch and Bound Method
32. Other than dynamic, which approach can be used for 0/1 knapsack problem
-->Recursion by Brute-Force algorithm OR Exhaustive Search.

33. Difference between 0/1 knapsack and fractional knapsack
--> As the name suggests, the “fractional knapsack” is the one in which we can take objects in fractions, i.e, in decimals (in floating points) whereas the “0/1 knapsack” is the one in which we can take objects in whole numbers (in interger value).

34. Difference between backtracking approach and Branch and bound
-->Backtracking

It is used to find all possible solutions available to a problem.
It traverses the state space tree by DFS(Depth First Search) manner.
It realizes that it has made a bad choice & undoes the last choice by backing up.
It searches the state space tree until it has found a solution.
It involves feasibility function.
Branch-and-Bound

It is used to solve optimization problem.
It may traverse the tree in any manner, DFS or BFS.

It realizes that it already has a better optimal solution that the pre-solution leads to so it abandons that pre-solution.
It completely searches the state space tree to get optimal solution.
It involves a bounding function.

35. Difference between dynamic and backtracking
-->Dynamic Programming (DP) is an algorithmic technique for solving an optimization problem by breaking it down into simpler subproblems and utilizing the fact
that the optimal solution to the overall problem depends upon the optimal solution to its subproblems. It is also about
building a recursive relation between the functions and subproblems and store them in a data structure generally
an array.
Backtracking is an algorithmic-technique for solving problems recursively by trying to build a
solution incrementally, one piece at a time, removing those solutions that fail
to satisfy the constraints of the problem at any point of time (by time, here,
is referred to the time elapsed till reaching any level of the search tree).

36. What is a spurious hit?
-->When the hash value of the pattern matches with the hash value of a window of the text but the window is not the actual pattern then it is called a spurious hit. Spurious hit increases the time complexity of the algorithm. In order to minimize spurious hit, we use modulus. It greatly reduces the spurious hit.

39. Complexity Binary Search???
-->The time complexity of the binary search algorithm is O(log n). The best-case time complexity would be O(1) when the central index would directly match the desired value.

41. State space tree
-->A state space tree is a tree constructed from all of the possible states of the problem as nodes, connected via state transitions from some initial state as root to some terminal state as leaf.

42. Dynamic programming and its application?
-->
Longest Common Subsequence
Travelling Salesman Problem

44. Which sorting algo is used in kruskals method and working kruskal ka
-->An application of the bucket sort in Kruskal's minimal spanning tree algorithm is proposed.

49. Why do we use branch and Bound
-->Branch and bound is an algorithm design paradigm which is generally used for solving combinatorial optimization problems. These problems are typically exponential in terms of time complexity and may require exploring all possible permutations in worst case.

50. Worst case scenario for quick sort
-->The worst case time complexity of a typical implementation of QuickSort is O(n2). The worst case occurs when the picked pivot is always an extreme (smallest or largest) element. This happens when input array is sorted or reverse sorted and either first or last element is picked as pivot.

52. Planar graph
-->a planar graph is a graph that can be embedded in the plane, i.e., it can be drawn on the plane in such a way that its edges intersect only at their endpoints. In other words, it can be drawn in such a way that no edges cross each other.

53. E node and live node
-->Live node is a node that has been generated but whose children have not yet been generated. E-node is a live node whose children are currently being explored. In other words, an E-node is a node currently being expanded. ... All children of a dead node have already been expanded.

Selection: O(n^2) space: O(1)
Insertion: space: O(n^2) best: O(n) space: O(1)
Merge Sort: O (n*log n)
Quick Sort: O(n*log n) worst: O(n^2)
Quick Sort: (n*log n)
Binary Search: O(log n) best: O(1)
Dijkstra: O(ElogV) worst: O(V^2)
Bellman-Ford: O(E.V)
Floyd-Warshall: O(V^3)
Job Sequencing: O(n^2)
Kruskals: O(ElogV)
Prims: O((V+E)logV)
Naive-String: O(n-m+1)
Rabin-Karp: O(mn)
KMP: O(n) linear complexity
LCS:worst:O(2^n)

- Njjjjjjjjjjjjjjjuuuyyuuuymyuuunuyjyuyuyy nhi

All the OP contributors to this sheet. We all did a great job. All those who didn’t and still used it, consider contributing next time.