DSA Interview Questions

Dear readers, these **Data Structures & Algorithms Interview Questions** have been designed specially to get you acquainted with the nature of questions you may encounter during your interview for the subject of **Data Structures & Algorithms**. As per my experience good interviewers hardly plan to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer:

What is data-structure?

Data structure is a way of defining, storing & retriving of data in a structural & systemetic way. A data structure may contain different type of data items.

What are various data-structures available?

Data structure availability may vary by programming languages. Commonly available data structures are list, arrays, stack, queues, graph, tree etc.

What is algorithm?

Algorithm is a step by step procedure, which defines a set of instructions to be executed in certain order to get the desired output.

Why we need to do algorithm analysis?

A problem can be solved in more than one ways. So, many solution algorithms can be derived for a given problem. We analyze available algorithms to find and implement the best suitable algorithm.

What are the criteria of algorithm analysis?

An algorithm are generally analyzed on two factors − time and space. That is, how much **execution** time and how much **extra space** required by the algorithm.

What is asymptotic analysis of an algorithm?

Asymptotic analysis of an algorithm, refers to defining the mathematical boundation/framing of its run-time performance. Using asymptotic analysis, we can very well conclude the best case, average case and worst case scenario of an algorithm.

What are asymptotic notations?

Asymptotic analysis can provide three levels of mathematical binding of execution time of an algorithm −

- Best case is represented by $\Omega(n)$ notation.
- Worst case is represented by $O(n)$ notation.
- Average case is represented by $\Theta(n)$ notation.

What is linear data structure?

A linear data-structure has sequentially arranged data items. The next time can be located in the next memory address. It is stored and accessed in a sequential manner. Array and list are example of linear data structure.

What are common operations that can be performed on a data-structure?

The following operations are commonly performed on any data-structure −

- **Insertion** − adding a data item
- **Deletion** − removing a data item
- **Traversal** − accessing and/or printing all data items
- **Searching** − finding a particular data item
- **Sorting** − arranging data items in a pre-defined sequence

Briefly explain the approaches to develop algorithms.

There are three commonly used approaches to develop algorithms −

- **Greedy Approach** − finding solution by choosing next best option
- **Divide and Conquer** − diving the problem to a minimum possible sub-problem and solving them independently
- **Dynamic Programming** − diving the problem to a minimum possible sub-problem and solving them combinedly

Give some examples greedy algorithms.

The below given problems find their solution using greedy algorithm approach −

- Travelling Salesman Problem
- Prim's Minimal Spanning Tree Algorithm

- Kruskal's Minimal Spanning Tree Algorithm

- Dijkstra's Minimal Spanning Tree Algorithm

- Graph - Map Coloring

- Graph - Vertex Cover

- Knapsack Problem

- Job Scheduling Problem

What are some examples of divide and conquer algorithms?

The below given problems find their solution using divide and conquer algorithm approach −

- Merge Sort
- Quick Sort
- Binary Search
- Strassen's Matrix Multiplication
- Closest pair (points)

What are some examples of dynamic programming algorithms?

The below given problems find their solution using divide and conquer algorithm approach −

- Fibonacci number series
- Knapsack problem
- Tower of Hanoi
- All pair shortest path by Floyd-Warshall
- Shortest path by Dijkstra
- Project scheduling

What is a linked-list?

A linked-list is a list of data-items connected with links i.e. pointers or references. Most modern high-level programming language does not provide the feature of directly accessing memory location, therefore, linked-list are not supported in them or available in form of inbuilt functions.

What is stack?

In data-structure, stack is an Abstract Data Type (ADT) used to store and retrieve values in Last In First Out method.

Why do we use stacks?

Stacks follows LIFO method and addition and retrieval of a data item takes only O(n) time. Stacks are used where we need to access data in the reverse order or their arrival. Stacks are used commonly in recursive function calls, expression parsing, depth first traversal of graphs etc.

What operations can be performed on stacks?

The below operations can be performed on a stack −

- **push()** − adds an item to stack

- **pop()** − removes the top stack item

- **peek()** − gives value of top item without removing it

- **isempty()** − checks if stack is empty

- **isfull()** − checks if stack is full

What is a queue in data-structure?

Queue is an abstract data structure, somewhat similar to stack. In contrast to stack, queue is opened at both end. One end is always used to insert data (enqueue) and the other is used to remove data (dequeue). Queue follows First-In-First-Out methodology, i.e., the data item stored first will be accessed first.

Why do we use queues?

As queues follows FIFO method, they are used when we need to work on data-items in exact sequence of their arrival. Every operating system maintains queues of various processes. Priority queues and breadth first traversal of graphs are some examples of queues.

What operations can be performed on Queues?

The below operations can be performed on a stack −

- **enqueue()** − adds an item to rear of the queue

- **dequeue()** − removes the item from front of the queue

- **peek()** − gives value of front item without removing it

- **isempty()** − checks if stack is empty

- **isfull()** − checks if stack is full

What is linear searching?

Linear search tries to find an item in a sequentially arranged data type. These sequentially arranged data items known as array or list, are accessible in incrementing memory location. Linear

search compares expected data item with each of data items in list or array. The average case time complexity of linear search is O(n) and worst case complexity is $O(n^2)$. Data in target arrays/lists need not to be sorted.

What is binary search?

A binary search works only on sorted lists or arrays. This search selects the middle which splits the entire list into two parts. First the middle is compared.

This search first compares the target value to the mid of the list. If it is not found, then it takes decision on whether.

What is bubble sort and how bubble sort works?

Bubble sort is comparison based algorithm in which each pair of adjacent elements is compared and elements are swapped if they are not in order. Because the time complexity is $O(n^2)$, it is not suitable for large set of data.

Tell me something about 'insertion sort'?

Insertion sort divides the list into two sub-list, sorted and unsorted. It takes one element at time and finds it appropriate location in sorted sub-list and insert there. The output after insertion is a sorted sub-list. It iteratively works on all the elements of unsorted sub-list and inserts them to sorted sub-list in order.

What is selection sort?

Selection sort is in-place sorting technique. It divides the data set into two sub-lists: sorted and unsorted. Then it selects the minimum element from unsorted sub-list and places it into the sorted list. This iterates unless all the elements from unsorted sub-list are consumed into sorted sub-list.

How insertion sort and selection sorts are different?

Both sorting techniques maintains two sub-lists, sorted and unsorted and both take one element at a time and places it into sorted sub-list. Insertion sort works on the current element in hand and places it in the sorted array at appropriate location maintaining the properties of insertion sort. Whereas, selection sort searches the minimum from the unsorted sub-list and replaces it with the current element in hand.

What is merge sort and how it works?

Merge sort is sorting algorithm based on divide and conquer programming approach. It keeps on dividing the list into smaller sub-list until all sub-list has only 1 element. And then it merges them in

a sorted way until all sub-lists are consumed. It has run-time complexity of O(n log n) and it needs O(n) auxiliary space.

What is shell sort?

Shell sort can be said a variant of insertion sort. Shell sort divides the list into smaller sublist based on some **gap** variable and then each sub-list is sorted using insertion sort. In best cases, it can perform upto O(n log n).

How quick sort works?

Quick sort uses divide and conquer approach. It divides the list in smaller 'partitions' using 'pivot'. The values which are smaller than the pivot are arranged in the left partition and greater values are arranged in the right partition. Each partition is recursively sorted using quick sort.

What is a graph?

A graph is a pictorial representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are represented by points termed as vertices, and the links that connect the vertices are called edges.

How depth first traversal works?

Depth First Search algorithm(DFS) traverses a graph in a depthward motion and uses a stack to remember to get the next vertex to start a search when a dead end occurs in any iteration.

How breadth first traversal works?

Breadth First Search algorithm(BFS) traverses a graph in a breadthwards motion and uses a queue to remember to get the next vertex to start a search when a dead end occurs in any iteration.

What is a tree?

A tree is a minimally connected graph having no loops and circuits.

What is a binary tree?

A binary tree has a special condition that each node can have two children at maximum.

What is a binary search tree?

A binary search tree is a binary tree with a special provision where a node's left child must have value less than its parent's value and node's right child must have value greater than it's parent

value.

What is tree traversal?

Tree traversal is a process to visit all the nodes of a tree. Because, all nodes are connected via edges (links) we always start from the root (head) node. There are three ways which we use to traverse a tree −

- In-order Traversal
- Pre-order Traversal
- Post-order Traversal

See the below image of a binary search tree, and traverse it using all available methods −
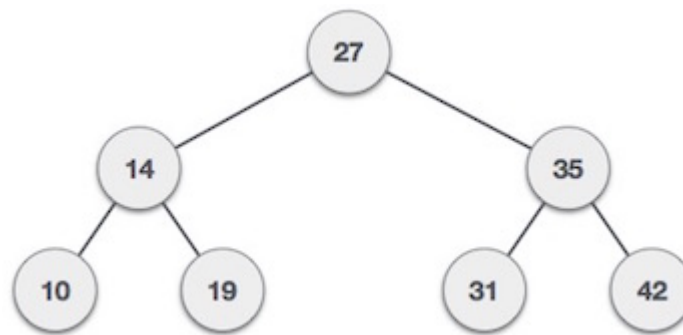
What is an AVL Tree?

AVL trees are height balancing binary search tree. AVL tree checks the height of left and right sub-trees and assures that the difference is not more than 1. This difference is called Balance Factor.

```
BalanceFactor = height(left-sutree) - height(right-sutree)
```

What is a spanning tree?

A spanning tree is a subset of Graph G, which has all the vertices covered with minimum possible number of edges. A spanning tree does not have cycles and it can not be disconnected.

How many spanning trees can a graph has?

It depends on how connected the graph is. A complete undirected graph can have maximum n^{n-1} number of spanning trees, where n is number of nodes.

How Kruskal's algorithm works?

This algorithm treats the graph as a forest and every node it as an individual tree. A tree connects to another only and only if it has least cost among all available options and does not violate MST properties.

How Prim's algorithm finds spanning tree?

Prim's algorithm treats the nodes as a single tree and keeps on adding new nodes to the spanning tree from the given graph.

What is a minimum spanning tree (MST)?

In a weighted graph, a minimum spanning tree is a spanning tree that has minimum weight that all other spanning trees of the same graph.

What is a heap in data structure?

Heap is a special balanced binary tree data structure where root-node key is compared with its children and arranged accordingly. A min-heap, a parent node has key value less than its childs and a max-heap parent node has value greater than its childs.

What is a recursive function?

A recursive function is one which calls itself, directly or calls a function that in turn calls it. Every recursive function follows the recursive properties − **base criteria** where functions stops calling itself and **progressive approach** where the functions tries to meet the base criteria in each iteration.

What is tower of hanoi?

Tower of Hanoi, is a mathematical puzzle which consists of three tower (pegs) and more than one rings. All rings are of different size and stacked upon each other where the large disk is always below the small disk. The aim is to move the tower of disk from one peg to another, without breaking its properties.

What is fibonacci series?

Fibonacci Series generates subsequent number by adding two previous numbers. For example − 0 1 1 2 3 5 8 13.

What is hashing?

Hashing is a technique to convert a range of key values into a range of indexes of an array. By using hash tables, we can create an associative data storage where data index can be find by providing its key values.

What is interpolation search technique?

Interpolation search is an improved variant of binary search. This search algorithm works on the probing position of required value.

What is the prefix and post fix notation of (a + b) * (c + d) ?

Prefix Notation − * + a b + c d

Postfix Notation − a b + c d + *

What is Next ?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)