Data Structures and Algorithm Concepts

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 Ω(n) notation.
- Worst case is represented by O(n) notation.
- Average case is represented by Θ(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?

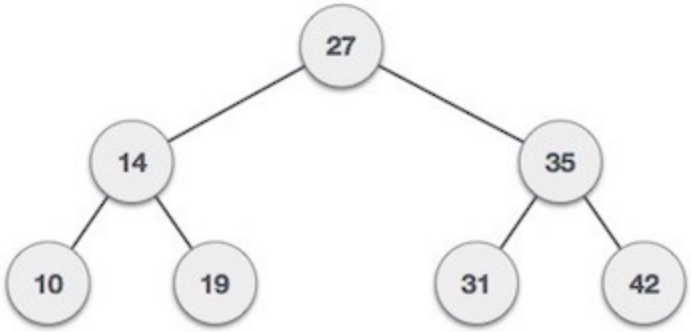
A binary search tree is a binary tree with a special provision where a node's left child must have value less than its parent's value and node's right child must have value greater than it's parent value.

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 :-)

https://www.tutorialspoint.com/data_structures_algorithms/data_structures_algorithms_interview_questions.htm

Top 10 algorithms in Interview Questions

In this post “Top 10 coding problems of important topics with their solutions ” are written. If you are preparing for a coding interview, going through these problems is a must.

Topics :

1. **Graph**
2. **Linked List**
3. **Dynamic Programming**
4. **Sorting And Searching**
5. **Tree / Binary Search Tree**

6. **Number Theory**
7. **BIT Manipulation**
8. **String / Array**

Graph

1. Breadth First Search (BFS)
2. Depth First Search (DFS)
3. Shortest Path from source to all vertices **Dijkstra**
4. Shortest Path from every vertex to every other vertex **Floyd Warshall**
5. To detect cycle in a Graph **Union Find**
6. Minimum Spanning tree **Prim**
7. Minimum Spanning tree **Kruskal**
8. Topological Sort
9. Boggle (Find all possible words in a board of characters)
10. Bridges in a Graph

Linked List

1. Insertion of a node in Linked List (On the basis of some constraints)
2. Delete a given node in Linked List (under given constraints)
3. Compare two strings represented as linked lists
4. Add Two Numbers Represented By Linked Lists
5. Merge A Linked List Into Another Linked List At Alternate Positions
6. Reverse A List In Groups Of Given Size
7. Union And Intersection Of 2 Linked Lists
8. Detect And Remove Loop In A Linked List
9. Merge Sort For Linked Lists
10. Select A Random Node from A Singly Linked List

Dynamic Programming

1. Longest Common Subsequence
2. Longest Increasing Subsequence
3. Edit Distance
4. Minimum Partition
5. Ways to Cover a Distance
6. Longest Path In Matrix
7. Subset Sum Problem

8. Optimal Strategy for a Game
9. 0-1 Knapsack Problem
10. Boolean Parenthesization Problem

Sorting And Searching

1. Binary Search
2. Search an element in a sorted and rotated array
3. Bubble Sort
4. Insertion Sort
5. Merge Sort
6. Heap Sort (Binary Heap)
7. Quick Sort
8. Interpolation Search
9. Find Kth Smallest/Largest Element In Unsorted Array
10. Given a sorted array and a number x, find the pair in array whose sum is closest to x

Tree / Binary Search Tree

1. Find Minimum Depth of a Binary Tree
2. Maximum Path Sum in a Binary Tree
3. Check if a given array can represent Preorder Traversal of Binary Search Tree
4. Check whether a binary tree is a full binary tree or not
5. Bottom View Binary Tree
6. Print Nodes in Top View of Binary Tree
7. Remove nodes on root to leaf paths of length < K
8. Lowest Common Ancestor in a Binary Search Tree
9. Check if a binary tree is subtree of another binary tree
10. Reverse alternate levels of a perfect binary tree

Number Theory

1. Modular Exponentiation
2. Modular multiplicative inverse
3. Primality Test | Set 2 (Fermat Method)
4. Euler's Totient Function
5. Sieve of Eratosthenes
6. Convex Hull
7. Basic and Extended Euclidean algorithms
8. Segmented Sieve

9. Chinese remainder theorem
10. Lucas Theorem

BIT Manipulation

1. Maximum Subarray XOR
2. Magic Number
3. Sum of bit differences among all pairs
4. Swap All Odds And Even Bits
5. Find the element that appears once
6. Binary representation of a given number
7. Count total set bits in all numbers from 1 to n
8. Rotate bits of a number
9. Count number of bits to be flipped to convert A to B
10. Find Next Sparse Number

String / Array

1. Reverse an array without affecting special characters
2. All Possible Palindromic Partitions
3. Count triplets with sum smaller than a given value
4. Convert array into Zig-Zag fashion
5. Generate all possible sorted arrays from alternate elements of two given sorted arrays
6. Pythagorean Triplet in an array
7. Length of the largest subarray with contiguous elements
8. Find the smallest positive integer value that cannot be represented as sum of any subset of a given array
9. Smallest subarray with sum greater than a given value
10. Stock Buy Sell to Maximize Profit

Top 10 Algorithms and Data Structures for Competitive Programming

Company-wise Practice Questions

Interview Corner

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

http://www.geeksforgeeks.org/top-10-algorithms-in-interview-questions/

http://www.crazyforcode.com/data-structure-interview-questions/

1) What is data structure?

Data structures refers to the way data is organized and manipulated. It seeks to find ways to make data access more efficient. When dealing with data structure, we not only focus on one piece of data, but rather different set of data and how they can relate to one another in an organized manner.

2) Differentiate file structure from storage structure.

Basically, the key difference is the memory area that is being accessed. When dealing with the structure that resides the main memory of the computer system, this is referred to as storage structure. When dealing with an auxiliary structure, we refer to it as file structures.

3) When is a binary search best applied?

A binary search is an algorithm that is best applied to search a list when the elements are already in order or sorted. The list is search starting in the middle, such that if that middle value is not the target search key, it will check to see if it will continue the search on the lower half of the list or the higher half. The split and search will then continue in the same manner.

4) What is a linked list?

A linked list is a sequence of nodes in which each node is connected to the node following it. This forms a chain-like link of data storage.

5) How do you reference all the elements in a one-dimension array?

To do this, an indexed loop is used, such that the counter runs from 0 to the array size minus one. In this manner, we are able to reference all the elements in sequence by using the loop counter as the array subscript.

6) In what areas do data structures applied?

Data structures is important in almost every aspect where data is involved. In general, algorithms that involve efficient data structure is applied in the following areas: numerical

analysis, operating system, A.I., compiler design, database management, graphics, and statistical analysis, to name a few.

7) What is LIFO?

LIFO is short for Last In First Out, and refers to how data is accessed, stored and retrieved. Using this scheme, data that was stored last , should be the one to be extracted first. This also means that in order to gain access to the first data, all the other data that was stored before this first data must first be retrieved and extracted.

8) What is a queue?

A queue is a data structures that can simulates a list or stream of data. In this structure, new elements are inserted at one end and existing elements are removed from the other end.

9) What are binary trees?

A binary tree is one type of data structure that has two nodes, a left node and a right node. In programming, binary trees are actually an extension of the linked list structures.

10) Which data structures is applied when dealing with a recursive function?

Recursion, which is basically a function that calls itself based on a terminating condition, makes use of the stack. Using LIFO, a call to a recursive function saves the return address so that it knows how to return to the calling function after the call terminates.

11) What is a stack?

A stack is a data structure in which only the top element can be accessed. As data is stored in the stack, each data is pushed downward, leaving the most recently added data on top.

12) Explain Binary Search Tree

A binary search tree stores data in such a way that they can be retrieved very efficiently. The left subtree contains nodes whose keys are less than the node's key value, while the right subtree contains nodes whose keys are greater than or equal to the node's key value. Moreover, both subtrees are also binary search trees.

13) What are multidimensional arrays?

Multidimensional arrays make use of multiple indexes to store data. It is useful when storing data that cannot be represented using a single dimensional indexing, such as data representation in a board game, tables with data stored in more than one column.

14) Are linked lists considered linear or non-linear data structures?

It actually depends on where you intend to apply linked lists. If you based it on storage, a linked list is considered non-linear. On the other hand, if you based it on access strategies, then a linked list is considered linear.

15) How does dynamic memory allocation help in managing data?

Aside from being able to store simple structured data types, dynamic memory allocation can combine separately allocated structured blocks to form composite structures that expand and contract as needed.

16) What is FIFO?

FIFO is short for First-in, First-out, and is used to represent how data is accessed in a queue. Data has been inserted into the queue list the longest is the one that is removed first.

17) What is an ordered list?

An ordered list is a list in which each node's position in the list is determined by the value of its key component, so that the key values form an increasing sequence, as the list is traversed.

18) What is merge sort?

Merge sort takes a divide-and-conquer approach to sorting data. In a sequence of data, adjacent ones are merged and sorted to create bigger sorted lists. These sorted lists are then merged again to form an even bigger sorted list, which continuous until you have one single sorted list.

19) Differentiate NULL and VOID.

Null is actually a value, whereas Void is a data type identifier. A variable that is given a Null value simply indicates an empty value. Void is used to identify pointers as having no initial size.

20) What is the primary advantage of a linked list?

A linked list is a very ideal data structure because it can be modified easily. This means that modifying a linked list works regardless of how many elements are in the list.

21) What is the difference between a PUSH and a POP?

Pushing and popping applies to the way data is stored and retrieved in a stack. A push denotes data being added to it, meaning data is being "pushed" into the stack. On the other

hand, a pop denotes data retrieval, and in particular refers to the topmost data being accessed.

22) What is a linear search?

A linear search refers to the way a target key is being searched in a sequential data structure. Using this method, each element in the list is checked and compared against the target key, and is repeated until found or if the end of the list has been reached.

23) How does variable declaration affect memory allocation?

The amount of memory to be allocated or reserved would depend on the data type of the variable being declared. For example, if a variable is declared to be of integer type, then 32 bits of memory storage will be reserved for that variable.

24) What is the advantage of the heap over a stack?

Basically, the heap is more flexible than the stack. That's because memory space for the heap can be dynamically allocated and de-allocated as needed. However, memory of the heap can at times be slower when compared to that stack.

25) What is a postfix expression?

A postfix expression is an expression in which each operator follows its operands. The advantage of this form is that there is no need to group sub-expressions in parentheses or to consider operator precedence.

26) What is Data abstraction?

Data abstraction is a powerful tool for breaking down complex data problems into manageable chunks. This is applied by initially specifying the data objects involved and the operations to be performed on these data objects without being overly concerned with how the data objects will be represented and stored in memory.

27) How do you insert a new item in a binary search tree?

Assuming that the data to be inserted is a unique value (that is, not an existing entry in the tree), check first if the tree is empty. If it's empty, just insert the new item in the root node. If it's not empty, refer to the new item's key. If it's smaller than the root's key, insert it into the root's left subtree, otherwise, insert it into the root's right subtree.

28) How does a selection sort work for an array?

The selection sort is a fairly intuitive sorting algorithm,, though not necessarily efficient. To perform this, the smallest element is first located and switched with the element at subscript

zero, thereby placing the smallest element in the first position. The smallest element remaining in the subarray is then located next with subscripts 1 through n-1 and switched with the element at subscript 1, thereby placing the second smallest element in the second position. The steps are repeated in the same manner till the last element.

29) How do signed and unsigned numbers affect memory?

In the case of signed numbers, the first bit is used to indicate whether positive or negative, which leaves you with one bit short. With unsigned numbers, you have all bits available for that number. The effect is best seen in the number range (unsigned 8 bit number has a range 0-255, while 8-bit signed number has a range -128 to +127.

30) What is the minimum number of nodes that a binary tree can have?

A binary tree can have a minimum of zero nodes, which occurs when the nodes have NULL values. Furthermore, a binary tree can also have 1 or 2 nodes.

31) What are dynamic data structures?

Dynamic data structures are structures that expand and contract as a program runs. It provides a flexible means of manipulating data because it can adjust according to the size of the data.

32) In what data structures are pointers applied?

Pointers that are used in linked list have various applications in data structure. Data structures that make use of this concept include the Stack, Queue, Linked List and Binary Tree.

33) Do all declaration statements result in a fixed reservation in memory?

Most declarations do, with the exemption of pointers. Pointer declaration does not allocate memory for data, but for the address of the pointer variable. Actual memory allocation for the data comes during run-time.

34) What are ARRAYs?

When dealing with arrays, data is stored and retrieved using an index that actually refers to the element number in the data sequence. This means that data can be accessed in any order. In programming, an array is declared as a variable having a number of indexed elements.

35) What is the minimum number of queues needed when implementing a priority queue?

The minimum number of queues needed in this case is two. One queue is intended for sorting priorities while the other queue is intended for actual storage of data.

36) Which sorting algorithm is considered the fastest?

There are many types of sorting algorithms: quick sort, bubble sort, balloon sort, radix sort, merge sort, etc. Not one can be considered the fastest because each algorithm is designed for a particular data structure and data set. It would depend on the data set that you would want to sort.

37) Differentiate STACK from ARRAY.

Data that is stored in a stack follows a LIFO pattern. This means that data access follows a sequence wherein the last data to be stored will the first one to be extracted. Arrays, on the other hand, does not follow a particular order and instead can be accessed by referring to the indexed element within the array.

38) Give a basic algorithm for searching a binary search tree.

1. if the tree is empty, then the target is not in the tree, end search
2. if the tree is not empty, the target is in the tree
3. check if the target is in the root item
4. if target is not in the root item, check if target is smaller than the root's value
5. if target is smaller than the root's value, search the left subtree
6. else, search the right subtree

39) What is a dequeue?

A dequeue is a double-ended queue. This is a structure wherein elements can be inserted or removed from either end.

40) What is a bubble sort and how do you perform it?

A bubble sort is one sorting technique that can be applied to data structures such as an array. It works by comparing adjacent elements and exchanges their values if they are out of order. This method lets the smaller values "bubble" to the top of the list, while the larger value sinks to the bottom.

41) What are the parts of a linked list?

A linked list typically has two parts: the head and the tail. Between the head and tail lie the actual nodes, with each node being linked in a sequential manner.

42) How does selection sort work?

Selection sort works by picking the smallest number from the list and placing it at the front. This process is repeated for the second position towards the end of the list. It is the simplest sort algorithm.

43) What is a graph?

A graph is one type of data structure that contains a set of ordered pairs. These ordered pairs are also referred to as edges or arcs, and are used to connect nodes where data can be stored and retrieved.

44) Differentiate linear from non linear data structure.

Linear data structure is a structure wherein data elements are adjacent to each other. Examples of linear data structure include arrays, linked lists, stacks and queues. On the other hand, non-linear data structure is a structure wherein each data element can connect to more than two adjacent data elements. Examples of non linear data structure include trees and graphs.

45) What is an AVL tree?

An AVL tree is a type of binary search tree that is always in a state of partially balanced. The balance is measured as a difference between the heights of the subtrees from the root. This self-balancing tree was known to be the first data structure to be designed as such.

46) What are doubly linked lists?

Doubly linked lists are a special type of linked list wherein traversal across the data elements can be done in both directions. This is made possible by having two links in every node, one that links to the next node and other one that links to the previous node.

47) What is Huffman’s algorithm?

Huffman’s algorithm is associated in creating extended binary trees that has minimum weighted path lengths from the given weights. It makes use of a table that contains frequency of occurrence for each data element.

48) What is Fibonacci search?

Fibonacci search is a search algorithm that applies to a sorted array. It makes use of a divide-and-conquer approach that can greatly reduce the time needed in order to reach the target element.

49) Briefly explain recursive algorithm.

Recursive algorithm targets a problem by dividing it into smaller, manageable sub-problems. The output of one recursion after processing one sub-problem becomes the input to the next recursive process.

50) How do you search for a target key in a linked list?

To find the target key in a linked list, you have to apply sequential search. Each node is traversed and compared with the target key, and if it is different, then it follows the link to the next node. This traversal continues until either the target key is found or if the last node is reached.

http://career.guru99.com/top-50-data-structure-interview-questions/