DS INTERVIEW QUESTIONS

Most asked Interview Questions

BY : CODE OF GEEKS

CODE OF GEEKS

Data Structures Interview Questions

This e-book contains **most frequently asked questions** on **Data Structures**.

Prepared by **CODE OF GEEKS**

In case of bug-reporting, please mail us at

support@codeofgeeks.com

SET 1 : Q1-Q5

Q. What is Data?

A. Data is a raw material for data processing. It refers to unprocessed information.

Q. What is information?

A. It is the data that has been processed in the meaningful form to the one who receives it.

Q. What is Data Structure?

A. It is the specified format for organizing and storing data.

Q. What are the different type of data structures?

A. Data Structures are divided into two categories:

1. Primitive Data Structure (int, char, float, double etc)

2. Non Primitive Data Structure

Non Primitive datatype are further divided into two categories :

a. Linear (Array, linked list, stack, queue)

b. Non Linear (Tree,Graph)

Q. What are the different operations applied on Data Structures?

A. Traversing, Searching, Insertion, deletion, sorting and searching.

SET 2 : Q6-Q10

Q. Give some areas where data structures are used?

A. Data Structure provide us means to manage large amounts of data efficiently for uses such as large databases and internet indexing services.

Q. Name the type of data structure used in following:

1. Hierarchical Data Model : Tree

2. RDBMS : Array

3. Network Data Model : Graph

Q. What is an Algorithm?

A. Algorithm denotes a sequence of steps to solve a particular problem.

Q. What are different approaches to develop algorithms?

A. Greedy, Divide and Conquer, Dynamic Programming.

Q. What is Hashing?

A. Hashing is the process of mapping a given value with a particular key for faster access of elements. Efficiency of mapping depends of the efficiency of the hash function used.

SET 3 : Q11-Q15

Q. Explain Greedy algorithm?

A. Algorithms following greedy approach build up solution step by step. It is mostly used in optimization problems. It makes optimal choice at each step, to solve entire problem.

Example: Dijikstra Algo, Prim’s Algo, Kruskal.

Q. Explain Divide & Conquer algorithm?

A. Algorithms following D&C approach works in two steps-Divide & Combine. Atfirst we divide the problem into subparts, solve them individually and then combine the result of all subparts to get a collective solution.

Example: Binary Search, Merge Sort.

Q. Explain Dynamic Programming?

A. DP is used to find the most optimized solution by eliminating the standard recursive calls.

Example: Finding fibonacci series.

Q. What are the parameters that are cared for an algorithm?

A. Time Complexity and Space Complexity.

Q. What are Abstract Datatypes?

A. ADTs are the special datatypes constructed from the collection of data items.

Example : Array, Linked list, Queue.

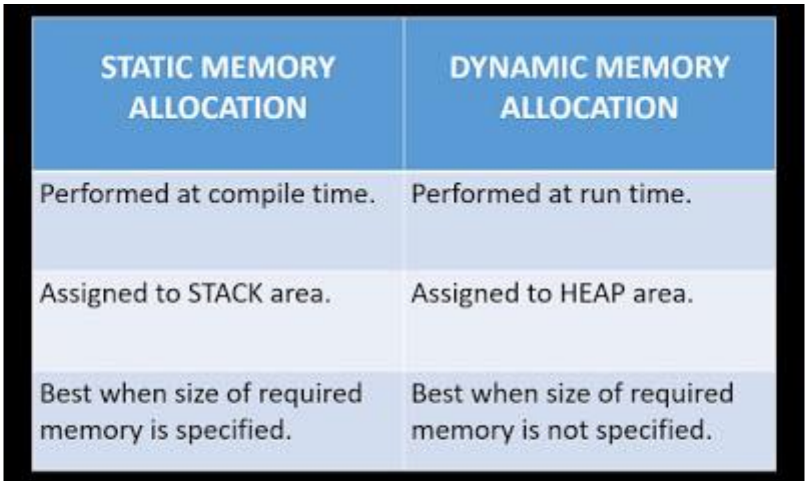
SET 4 : Q16-Q20

Q. What is the need of Data Structure?

A. It tells how data can be stored and accessed in its elementary level. It allows us to manage huge amount of data efficiently. It provides different techniques for searching and sorting data.

Q. Differentiate between Static memory allocation and dynamic memory allocation.

A.



STATIC MEMORY ALLOCATION	DYNAMIC MEMORY ALLOCATION
Performed at compile time.	Performed at run time.
Assigned to STACK area.	Assigned to HEAP area.
Best when size of required memory is specified.	Best when size of required memory is not specified.

Q. Explain the role of malloc(), calloc(), realloc() and free()in dynamic memory allocation.

A. 1. malloc() is one of the functions used for dynamic memory allocation. It takes up single arguments, which denotes the number of bytes to be allocated.

malloc() is faster than calloc().

Syntax : int *p = (int *)malloc(sizeof(int))

2. calloc() is one of the functions used for contiguous dynamic memory allocation. It takes up two arguments, in which first argument denotes the

number of bytes to be allocated, and second argument denotes size of each block. It initializes allocated memory by 0.

3. The realloc() function is used to resize allocated memory without losing old data.

Syntax: void *realloc(void *p, size_t newsize);

4. free() is use to free the memory block that had been allocated dynamically.

Q. What is Array?

A. Array refers to the collection of similar data items.

Syntax – int a[10];

Here, a is the array having size 10.

Q. What is 2D Array?

A. 2D array or 2 dimensional array is array of arrays. It is used to store the data in tabular form-in the terms of rows and columns.

It is represented as int a[m][n], where m denotes number of rows and n denotes number of columns.

SET 5 : Q21-Q25

Q. What is Linked List?

A. Linked list is a list of data elements linked to one another. In linked list, each element consists of a node, with two field each :

a. data field (variable that denotes the content of node)

b. next (pointer variable that stores the address of next node)

Types of Linked List :

1. Singly LL

2. Doubley LL

3. Circular LL

4. Circular Doubley LL

Q. How linked list is better than array?

A. 1. Array is static and Linked list is dynamic.

2. Linked list avoids memory wastage.

Q. What do you mean by Stack?

A. Stack is linear data structure in which insertion and deletion is done only from one end. It follows LIFO (Last In First Out) order.

Q. What are the operations that we can apply on stack?

A. Push : Insertion of element on top

Pop : Deletion of element on top

Q. What is Stack Overflow & Stack Underflow?

A. Stack Overflow : Condition when array is full and user requests for another insertion.

Stack Underflow : Condition when array is empty and user requests for deletion operation.

SET 6 : Q26-Q30

Q. In how ways we can implement stack?

A. Two ways : using arrays and linked list

Q. What is peek() operation in Stack?

A. It returns top most element of Stack.

Q. What is the condition of Stack Overflow?

A. top=n-1, where n is the number of elements in stack.

Q. Give some applications of stack?

A. 1. For data reversal.

2. Evaluating arithmetic operations.

3. To calculate postfix expressions.

4. For parsing.

5. For simulation of recursion.

Q. What is recursion?

A. Recursion is the process in which a function is called by itself again & again.

Example : Recursive solution of printing a factorial of a number

fact(n)

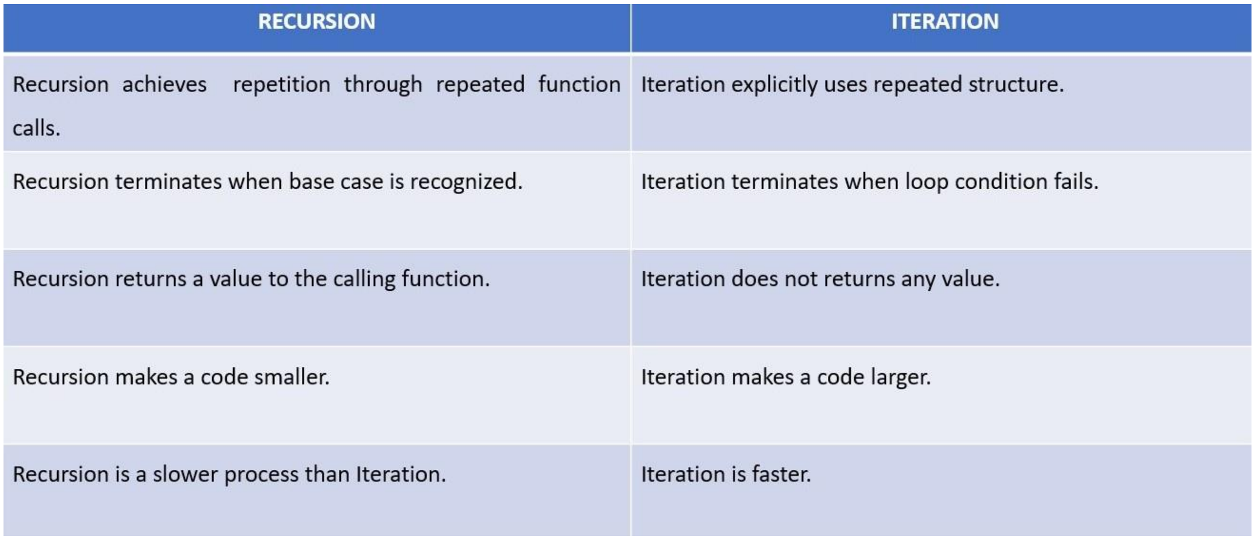
{

```
if(n==0)
{
return 1;
}
else
{
return n*fact(n-1);
}
}
```

SET 7 : Q31-Q35

Q. Differentiate between iteration and recursion?

A.



RECURSION	ITERATION
Recursion achieves repetition through repeated function calls.	Iteration explicitly uses repeated structure.
Recursion terminates when base case is recognized.	Iteration terminates when loop condition fails.
Recursion returns a value to the calling function.	Iteration does not returns any value.
Recursion makes a code smaller.	Iteration makes a code larger.
Recursion is a slower process than Iteration.	Iteration is faster.

Q. What is Queue?

A. Queue is a non-primitive non-linear data structure. It is a homogenous collection of elements. It follows FIFO order – First In First Out. Insertion of any new value takes place at ‘rear’ while deletion takes place at ‘front’.

Q. What is the basic condition to check whether a circular queue is full or not?

A. (rear+1)%size=front

Q. Explain me the concept of Priority Queue?

A. Priority queue is the collection of elements such that each element has been assigned a priority i.e order in which elements are deleted or processed. An element of high priority is processed before any element of lower priority.

Q. What are the minimum number of queues required to implement priority queue?

A. 2, one for data & one for priority.

SET 8 : Q36-Q40

Q. What is dequeue operation?

A. It is doubly-ended queue. In doubly-ended queue, insertion and deletion takes place at both ends.

Q. What do you know about Tree data structure?

A. Tree is a non-linear data structure which is defined as the finite set of one or more nodes.

Q. Explain me the property of Tree data structure?

A. * Top most node of a tree is known as ‘root’.

* Tree comprises of one or more subtrees.

Q. What is degree of a node?

A. It is the number of subtree(s) of a node.

Q. What is the degree of a tree?

A. It is the maximum degree of node in tree. A node of tree with degree 0 i.e no child is called leaf node.

SET 9 : Q41-Q45

Q. What is the depth of a tree?

A. It is also known as the height of a tree. It represents the maximum level of tree.

Q. What is Forest of a tree?

A. It is the set of disjoint tree. In a tree, if you remove its root node then it becomes a forest.

Q. How many children a binary tree may have?

A. 0| 1| 2.

Q. Differentiate between binary tree and tree.

A. In an ordinary tree, parent may have many children, but a binary tree can have atmost 2 children.

Q. Explain different types of binary tree?

A. 1. Full Binary tree : Every node has 0 or 2 children.

2. Complete Binary tree : All internal nodes have 2 children.

3. Skewed tree : Tree that goes in single direction.

Left Skewed tree : Node have only left child not right.

Right Skewed tree : Node have only right child not left.

SET 10 : Q46-Q50

Q. What is the maximum number of nodes in a binary tree of height 'h'?

A. 2^(h+1)

Q. What is the depth of binary tree with 'n' nodes?

A. D = log(base 2)(n)+1

Q. Explain the game of Tower of Hanoi.

A. The Tower of Hanoi is a mathematical game or puzzle. It consists of three rods and a number of disks of different sizes, which can slide onto any rod. The puzzle starts with the disks in a neat stack in ascending order of size on one rod, the smallest at the top, thus making a conical shape.

Q. In Tower of Hanoi game, what could be the possible number of moves for 5 disks?

A. 31.

General formula is given as : 2^(n)-1

Q. How Binary Search Tree works?

A. It is a binary tree which may be either empty or satisfies following properties :

1. Value of key in the left child is less than the value of the root.

2. Value of key in the right child is more than or equal to the value of the root.

SET 11 : Q51-Q55

Q. What are the different cases for deleting an element?

A. 1. When node (to be deleted) is leaf node – delete the node directly.

2. When node has only ONE child – Make child node as parent (swapping the position of both nodes) and then we can delete it directly.

3. When node has TWO children – Find inorder successor of that given node and swap their position and delete the node.

Q. Where BST is used?

A. 1. To implement multilevel indexing in database.

2. To implement various efficient algorithms.

3. To implement applications that require a sorted list as input like various e-commerce websites.

4. To manage Virtual Memory Areas.

5. To index various IP addresses.

Q. Give the worst case complexity for BST operation-insertion.

A. Insertion : O(h), h is height of BST.

Q. Give me the properties of Threaded Binary tree?

A. It is a binary tree with ‘n’ nodes out of which n+1 are always null, and this space is used to contain some useful information.

A Left null pointer stores the address of inorder predecessor of the node and a right null pointer stores the address of inorder successor of the node. These pointers are referred to as Threads.

It is more efficient than a normal binary tree.

Q. Explain the concept of Huffman coding?

A. It is a particular type of optimal prefix code that is used for lossless data compression. It is an example of Greedy algorithm. It assigns variable length code to all the characters. The code length of a character depends on how frequently it occurs in the given text. The character which occurs most frequently gets the smallest code. The character which occurs least frequently gets the largest code.

SET 12 : Q56-Q60

Q. Differentiate between B-Tree & B+ Tree?

A. B-tree is a self-balancing tree data structure that maintains sorted data and allows searches, sequential access, insertions, and deletions in logarithmic time. The B-tree is a generalization of a binary search tree in that a node can have more than two children.

In B+ Tree, each node contains key only(not pairs), and all pointers to data-records exists at leaf level only.

Q. How can we overcome drawbacks of BST?

A. Using AVL Trees.

Q. What is AVL Tree?

A. Concept of AVL Tree was given by Adelson, Velski, Landis. AVL trees are height balancing BST. It checks the height of left&right subtree and assures that difference is not more than 1, this difference is termed as balanced factor.

Q. What are the different operations applied on AVL tree?

A. 1. Left-Left Rotation (LL) : right rotate node p.

2. Left-Right Rotation (LR) : left rotate “parent of p” and right rotate “parent of parent of p (p-parent-parent)”.

3. Right-Right Rotation (RR) : left rotate node p.

4. Right-Left Rotation (RL) : right rotate “parent of p” and left rotate “parent of parent of p (p-parent-parent)”.

where p is the node with violating balance-factor.

Q. Define Graph Data Structure?

A. Graph is an abstract datatype, containing the set of vertices and edges.

SET 13 : Q61-Q65

Q. Define Path & cycle?

A. Path represents the sequence of adjacent vertices whereas a cycle represents a closed path.

Q. What are the different ways to represent a graph?

A. There are to ways to represent a graph, using Adjacency Matrix and Adjacency List.

Q. What are the application areas of Graph data structure?

A. 1. Circuit Designing.

2. Computer Networks.

3. In study of DNA structure of organism.

Q. What is a vertex in graph?

A. Vertex is a point where lines meet.

Q. What is eccentricity?

A. It is the maximum distance between a vertex to all other vertices.

SET 14 : Q66-Q70

Q. What is Spanning Tree?

A. It is a subset of a graph, which has all the vertices covered with minimum possible number of edges. It does not have cycles and can’t be disconnected.

Q. What is MST?

A. MST is minimum spanning tree. It is a spanning tree having minimum weight than any other spanning tree in the same graph.

Example : Prims, Kruskal algorithm.

Q. Tell something about BFS & DFS?

A. BFS is Breadth First Search. It is used to traverse a graph in horizontal manner.

DFS is Depth First Search. It is used to traverse a graph in vertical or depthward manner.

Q. Name the data structure used in BFS?

A. Queue.

Q. Name the data structure used in DFS?

A. Stack.

SET 15 : Q71-Q76

Q. Tell any one Single Source Shortest Path algorithm?

A. Dijikstra Algorithm.

Questions on Searching and Sorting techniques

Q. What is linear searching?

A. Linear Search is a linear time searching technique in which a complete list/array is traversed in order to search a element.

Time complexity of linear search : O(n), where n denotes the number of elements in an item.

Q. What is the best case for Binary Search?

A. When element is present at middle position.

Q. What are inplace sorting techniques, tell me one inplace and one non-inplace sorting technique.

A. Inplace sorting techniques does not require any extra space to sort a given list/array.

Inplace Sorting – Bubble Sort

Non-inplace Sorting – Merge Sort

Q. Is bubble sort stable?

A. Yes.

SET 16 : Q77-Q81

Q. Explain the working of selection sort?

A. In Selection sort, we select a minimum element from the array and store it in the appropriate position.

Time complexity for selection sort – O(n^2).

Q. Explain the working of Insertion Sort?

A. In this, left most element is considered to be already sorted. From remaining elements, left most is taken out and is compared to already sorted elements to its left.

Best Case Time Complexity : O(n)

Q. Is insertion sort stable?

A. Yes.

Q. Is insertion sort inplace?

A. Yes.

Q. Explain me the working of Quick sort?

A. Quick sort works on divide and conquer technique. In quick sort, we partition the array into subarrays, this partition takes place around a special element called pivot.

Time Complexity of Quick sort is O(nlogn).

SET 17 : Q82-Q86

Q. What is the worst time complexity of quick sort?

A. O(n^2), it happens when :

Array is already sorted in same order.

Array is already sorted in reverse order.

All elements are same.

Q. Is Quick Sort stable?

A. No.

Q. What is the role of minheap operation in heap sort?

A. It extracts the minimum element from heap.

Q. What is randomized quick sort?

A. In randomized quick sort, initial position of pivot is determined randomly.

Q. What are the disadvantages of radix sort?

A. It needs more memory.

This sorting algo. works for integers only.

So, these are the most asked interview questions on DSA.

We, at CODE OF GEEKS, wish you all the best for your upcoming future.

For any assistance, feel free to mail us at support@codeofgeeks.com.