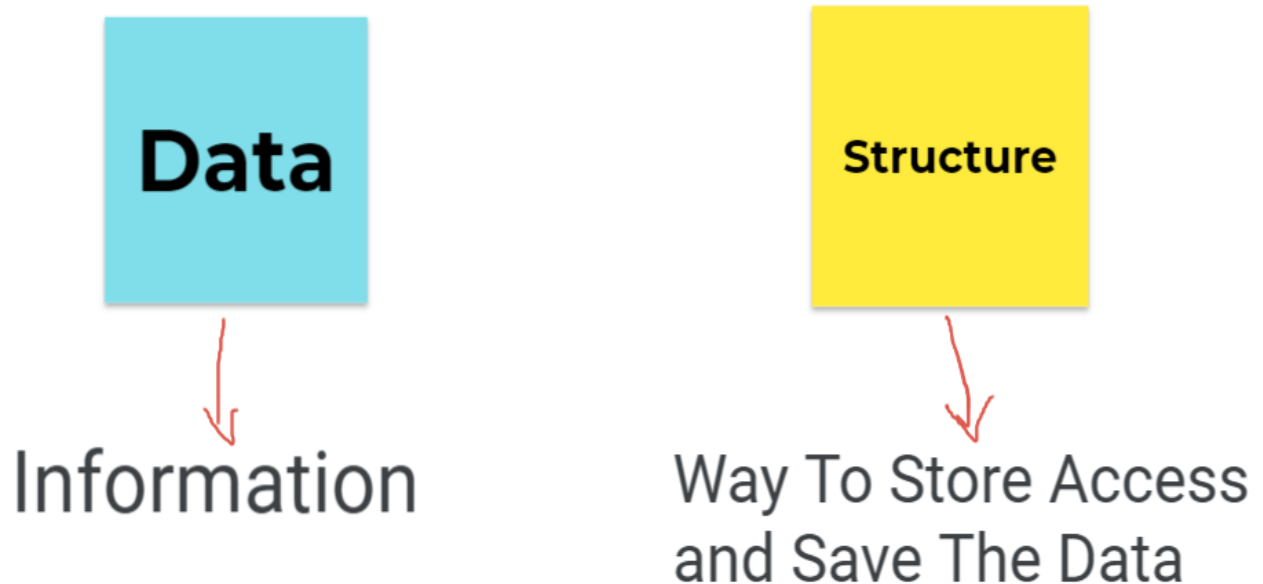


Data Structure Assignments _1 Feb 20

1. What are Data Structures?

Ans. Data Structure is a systematic way to organize data in order to use it efficiently. Data structures are the methods and techniques used to maintain data in an organized fashion. This is primarily done to ensure that data can be manipulated and accessed in an efficient manner. Data dependency and relationships between two or more entities of data also play a vital role in the concept of data structures. A data structure is a specialized format for organizing, processing, retrieving and storing data.



2. What is the difference between a File Structure and a Data Structure?

Ans-

File Structure	Data Structure
Representation of items in secondary memory.	Structural representation of data items in primary memory to do storage & retrieval operations efficiently.
Data stored on disk	Data stored on both RAM and disk
The file structures are more standardised	Data structures can change between app versions and there is no need for compatibility.
Standard file storage policies	Customized storage policies
Low compatibility with external apps	High compatibility with external apps.

3. What is a linked list?

Ans- A linked list is a sequence of data structures, which are connected together via links. Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. A linked list is a data structure that consists of individual entities called nodes. These nodes have the capability to connect to other nodes and create a chain in the process. This continuous chain structure forms a linked list.

4. Where are Data Structures primarily used?

Ans-In general, data structures are used to implement the physical forms of abstract datatypes. This can be translated into a variety of applications, such as displaying a relational database as a binary tree. Data structures are very much needed in almost all of the fields that you can think of. Algorithms are the primary requirement in every data handling situation. Following are some of the scenarios where data structures are widely used:

Data Organisation

Web pages

Linked Images Or Playlist

Multiplayer Games

Numerical computation

Operating system design

Artificial Intelligence

Compiler design

Database handling

Graphical processing

Lexical analysis

Statistics

5. What are the types of searching used in Data Structures?

Ans- There are 2 types of search linear and binary Search. Linear search algorithm is very simple and has $O(n)$ of complexity whereas Binary Search is a very fast searching algorithm having the complexity of $(\log n)$ but can only be used in case of the sorted list of elements.

Linear search involves iterating over a data unit in order to perform the required operation.

Binary search is more efficient in a way that it has the ability to split the data unit into chunks and then perform a search operation.

6. How does binary search work?

Ans- Binary search is used when there is primarily a criterion of efficiency. It involves working on the already ordered data, which is sorted either in ascending or in descending order. To begin with, the middle element of the array is found out, and the search begins from there. The array is searched in two parts based on the search value being higher or lower than the middle element. It is key to know the order of the arrangement to help search the value accordingly.

For a binary search to work, it is mandatory for the target array to be sorted. Binary search halves the searchable items and thus reduces the count of comparisons to be made to very less numbers.

7. How are individual elements accessed in an array?

Ans-

The elements of an array can be accessed by specifying the array name followed by subscript in brackets like say in array of 5 elements ie-int arr[5], we want to access the third element we will do so by saying arr[2] this will return element at third position in an array.

Each of the values in an array is given an index position starting from 0 to n-1, where 'n' is the number of elements in the array. Individual elements can be accessed by using the index element for operations. Multi-dimensional arrays will have more than one dimension to work with. So a 2-d array we have to two subscripts data_type_arrayname [rowsize] [columnsize]'

8. What is a queue in Data Structures?

Ans- A queue is a widely used data structure that is used to denote the ordered access and manipulation of an element. Queue is an abstract data structure, somewhat similar to Stacks. Unlike stacks, a queue is open at both its ends. 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.

The operation of this data structure is exactly the same as a literal queue in the real world like queue at bus stops or ticket window. Elements are added one after the other and are processed on the front end.

9. What is a binary tree?

Ans- Binary Tree is a special datastructure used for data storage purposes. A binary tree has a special condition that each node can have a maximum of two children. A binary tree has the benefits of both an ordered array and a linked list as search is as quick as in a sorted array and insertion or deletion operation are as fast as in linked list.

A binary tree, as the name suggests, is a tree data structure with two nodes, which are the nodes on the left and the right sides of the root node. In usage, binary trees are considered to be an extended linked list.

10. What is the meaning of stack?

Ans-A stack is another widely used data structure that provides users with the ability to work with data at one point only. A stack is an Abstract Data Type (ADT), commonly used in most programming languages. It is named stack as it behaves like a real-world stack, for example – a deck of cards or a pile of plates, etc.

11. What is the working of LIFO?

Ans-LIFO stands for Last-in-first-out access order. Here, the element which is placed (inserted or added) last, is accessed first. It is directly corresponding to how the data can be worked on and modified. The data entity that is stored or pushed in last is the first one to be worked

on at any point in time. If there is a requirement to access the very first element stored, then first you have to retrieve all of the data that came in after that element.

12. What are multi-dimensional arrays?

Ans-In C/C++, we can define multidimensional arrays in simple words as array of arrays. Data in multidimensional arrays are stored in tabular form (in row major order). Multi-dimensional arrays are arrays that span across more than one dimension. This means that they will have more than one index variable for every point of storage. This is primarily used in cases where data cannot be represented or stored using only one dimension.

13. Are linked lists Linear or Non-linear Data Structures?

Ans-Linked lists are considered to be the best of both worlds here. Based on usage, if it is a storage policy, then it can be considered as non-linear. Whereas, if a person is considering it based on retrieval strategies, then it can be considered linear.

According to Access strategies Linked list is a linear one. According to Storage Linked List is a Non-linear one.

14. What is a Binary Search Tree?

Ans-Binary Search tree exhibits a special behavior. A node's left child must have a value less than its parent's value and the node's right child must have a value greater than its parent value. A binary search tree is a data structure that stores data in a very efficient manner. It consists of two primary nodes from the root node. The main thing here is that the values of the nodes in the left sub-tree are less in number than the value of the root node, and the values of the nodes on the right of the root node are correspondingly higher than the root. Also, individually both of these left and right sub-trees are their own binary search trees at all points of time.

15. What is the meaning of FIFO?

Ans-FIFO, also known as First in, First out, is a way of representing a data operation on factors such as how data is accessed and in what order. Here, the data that is first put into the list will be the first entity to exit from the ordered data structure.

It is a method for handling data structures where the first element is processed first and the newest element is processed last.

16. What is the difference between void and null in Data Structures?

Ans-Void is a data type identifier in data structures, while null is considered to be a value with no physical presence. When void is used, it indicates that there is no size while initializing the data structure.

17. What is dynamic memory management?

Ans-Dynamic memory allocation is when an executing program requests that the operating system give it a block of main memory. The program then uses this memory for some purpose. Usually the purpose is to add a node to a data structure.

Dynamic memory management is a technique in which storage units are allocated based on the requirements continuously. Using dynamic memory allocations, individual data structures can be either stored separately or combined to form entities called composites. These composites can be worked on when required.

18. What are push and pop operations in Data Structures?

Ans-Both push and pop operations denote how data can be stored and used when required in a stack. The push operation denotes that users are adding data into the structure, and the pop operation denotes that the data is being pulled or removed from the structure. Usually, the

top-most element is considered when performing push and pop operations.

- `push()` – Pushing (storing) an element on the stack.
- `pop()` – Removing (accessing) an element from the stack.

19. How is a variable stored in memory when using Data Structures?

Ans-A variable is stored based on the amount of memory that is needed. First, the required quantity of memory is assigned, and later, it is stored based on the data structure being used. Using concepts such as dynamic allocation ensures high efficiency and that the storage units can be supplied based on the requirements in real time.

20. What is merge sort?

Ans-Merge sort is a sorting technique based on divide and conquer technique. With worst-case time complexity being $O(n \log n)$, it is one of the most respected algorithms. Merge sort first divides the array into equal halves and then combines them in a sorted manner. Here, data entities adjacent to each other are first merged and sorted in every iteration to create sorted lists. These smaller sorted lists are combined at the end to form the completely sorted list.

21. Why should heap be used over a stack?

Ans-The heap data structure is more efficient to work with when compared to stack in a way that memory allocation in a heap is dynamic and can be allocated and removed as per the requirement. The memory structure and access time of a stack are comparatively slow.

22. What is the meaning of Data Abstraction?

Ans-Data abstraction refers to providing only essential information to the outside world and hiding their background details, i.e., to represent the needed information in program without presenting the details. Data abstraction is one of the widely used tools in data structures. The goal is to break down complex entities into smaller problems and solve these by using the concepts of data structures. This provides users with the advantage of being focused on the operations and not worried about how the data is stored or represented in the memory.

23. What is the meaning of a post-fix expression in Data Structures?

Ans-Post-fix expressions are used in a scenario where every operator is preceded by its operands. Using this ensures to eliminate the need for parentheses or sub-expressions when it comes to the concept of operator precedence.

24. What is the working of a selection sort?

Ans-A selection sort is a widely used sorting algorithm in the world of Data Structures. The working is simple where the smallest entity is first found out and the index of that is set to zero, thereby permanently sorting this in the first step. The remaining steps involve iterating through other elements and adding the next index correspondingly. This is done until all of the elements are sorted. This sorting algorithm is an in-place comparison-based algorithm in which the list is divided into two parts, the sorted part at the left end and the unsorted part at the right end. Initially, the sorted part is empty and the unsorted part is the entire list. The smallest element is selected from the unsorted array and swapped with the leftmost element, and that element becomes a part of the sorted array. This

process continues moving unsorted array boundary by one element to the right.

25.What are signed numbers in Data Structures?

Ans-Signed numbers are the units that have a data bit at the beginning of the number that denotes if the number is positive or negative. Signed numbers have a range of -128 to +127.

26.What are the minimum nodes binary trees can have?

Ans-Binary trees can have zero nodes or a minimum of 1 or 2 as well. It can be zero in a case where all of the nodes have a NULL value. A Binary Tree can not have more than 2 nodes.

27.What Data Structures make use of pointers?

Ans-Pointers are used in a variety of data structures. Following are the data structures where pointers are used-

- Linked Lists
- Stacks
- Queues
- Binary trees

28.What is the use of dynamic Data Structures?

Ans-There are many situation where the number of items to be stored is not known beforehand. In this case we use dynamic data structure.

- Data Structure is allowed to grow and shrink as the demand for storage arises.
- Programmer should set a maximum size of data to help avoid memory collision.

Dynamic data structures provide users with a lot of flexibility in terms of the provision of data storage and manipulation techniques, which can change during the operation of the algorithm or the execution of the program.

29.What is a priority queue?

Ans-A priority queue is an abstract data type similar to regular queue or stack data structure in which each element additionally has a "priority" associated with it. In a priority queue, an element with high priority is served before an element with low priority.

A minimum of two queues are required in this case, one for the data and the other to store the priority.

30.Pointers allocate memory for data storage. True or False?

Ans-False.

Pointer operations such as declaration will not allocate any memory for the storage of data. But, memory is allocated for the variable that the pointer is pointing to. Memory processing begins only when the program begins its execution.

31.What is the meaning of deque?

Ans- Deque or Double Ended Queue is a generalized version of Queue data structure that allows insert and delete at both ends. A deque is a queue that is double-ended. This means that elements can be added or removed from any one of the two ends. It can be used both as a regular queue and as a stack. It performs better than both linked lists and stacks in general.

32.State the difference between Linear and Non-linear Data Structures.

Ans-

Linear DS	Non-Linear DS
Sequential Access	Direct/Guided Access
Data elements are stored next to each other	Data elements can be separated by other entities in memory
They are Simple In Nature	They need certain rule or method.
Theyre Slow.	Theyre fast.
Theyre easy to code	Theyre Complex To code.
E.g.: Arrays, linked lists, stacks, and queues	E.g.: Trees and graphs

33.What is the meaning of an AVL tree?

Ans-Named after their inventor Adelson, Velski & Landis, AVL trees are height balancing binary search tree. AVL tree checks the height of the left and the right sub-trees and assures that the difference is not more than 1. This difference is called the Balance Factor.

An AVL tree is a type of a binary search tree where the tree is only slightly balanced. Balance is the unit of comparison between the heights of the subtrees from the main (root) node.

Remaining (Some Not Taught Yet)

34.How does Huffman's algorithm work?

Huffman's algorithm uses a table, containing the frequency of the occurrence of every data entity on the list. This is used for creating extended binary trees, which are known to have minimum weights for the path lengths . This is considering each of the corresponding weights.

35. What are recursive algorithms?

Recursive algorithms are algorithms that solve a problem by breaking it down into simpler sub-problems and then solving them iteratively. The output of one recursion operation is usually the direct input for the next iteration operation, and this process goes on.

36. How does bubble sort work?

Bubble sort is one of the most used sorting techniques out there. It is applied to arrays where elements adjacent to each other are compared and values are exchanged based on the order of arrangement. It's called bubble sort because of the concept of exchanging elements like a bubble floating to the top of the water and larger entities sinking down to the bottom end.

37. Which is the fastest sorting algorithm available?

Among the many types of algorithms such as bubble sort, quick sort, merge sort, and more, it is not right to put one method on the podium for performance as this greatly varies based on data, the reaction after the algorithm processes the data, and how it's stored. The concept of time complexity is considered here.

38. What is the postfix form of: $(X + Y) * (Z - C)$

Ans-The postfix form of the given expression is $XY+ZC-*$

39. Where are Tree Data Structures used?

Ans-Tree data structures are used in a variety of applications. Following are some of them:

Arithmetic expression handling

Symbol table creation

Lexical analysis

Hierarchical data modeling

40. What are the Data Structures that are used in graphs?

To implement graphs, two data structures play a key role. They are:

Adjacency matrix: Used for sequential data representation

Adjacency list: Used to represent linked data

41. What are the Data Structures that are used in DFS and BFS algorithms?

In the depth-first search (DFS), the stack data structure is made use of.

In the case of the breadth-first search (BFS) technique, queues are used.

42. What are the time complexities of linear search and binary search?

Binary search is more effective as it takes lesser comparisons to search for an element in an array. The time complexity for linear search is $O(n)$, while it is $O(\log n)$ for binary search.

43. Where are Multi-linked Data Structures used?

Multi-linked data structures are used in many domains. Following are the two most important use cases of multi-linked data structures:

Generation of sparse matrices

Index creation for data entities

44. What is the method used for inorder traversal in trees?

Ans-Inorder traversal works in the following way:

- The tree is traversed through the left subtree.
- The root node is visited after the left visit.
- Then, the right subtree is traversed.

45. What is the working of post-order traversal in trees?

Ans-Postorder traversal works in the following way:

1. First, the left subtree is traversed through.
2. The right subtree is traversed next.
3. The root node is visited after the right subtree visit.

46. What are the disadvantages of implementing queues using arrays?

There are two main downsides when implementing queues using arrays. They are as follows:

Array sizing: The queue has to be constantly extended to make way for more elements that get implemented. Always extending the size of the array will not be feasible as there will be a discrepancy in the creation of the correct array size.

Memory dumps: The memory that is used to store the queue elements cannot be reused to actually store the queue. This is because of the working of queues where insertion happens at the head node only.

47. How can elements be inserted in the circular queue?

Ans-There are two cases in which items can be placed in a circular queue. They are as follows:

When $\text{front} \neq 0$ and $\text{rear} = \text{max} - 1$. This makes it possible as the queue will not be full, and new elements can be inserted here.

When $\text{rear} \neq \text{max} - 1$. This ensures that the rear is incremented to the maximum allocation size, and values can be inserted easily to the rear end of the queue.

Next on this Data Structures in Java interview questions is a question that is commonly asked, so pay attention.

48. What is the use of void pointers?

Void pointers are used because of their capability to store any pointer, which is pointing to a wide variety of data. It is used to implement heterogeneous linked lists in many programming languages.

49. What is the meaning of the stack overflow condition?

Stack overflow is the term given when the stack is full and an element cannot be inserted into the stack anymore.

Stack overflow happens when $\text{top} = \text{Maxsize} - 1$

50. Have you earned any sort of certification to boost your Data Structures learning?