

# DSA Interview

## Questions & Answers

### What are Data Structures?

A data structure is a mechanical or logical way that data is organized within a program. The organization of data is what determines how a program performs. There are many types of data structures, each with its own uses.

### What are some applications of Data structures?

- i) Decision Making
- ii) Genetics
- iii) Image Processing
- iv) Blockchain
- v) Numerical and Statistical Analysis
- vi) Compiler Design
- vii) Database Design and many more.

### Explain the process behind storing a variable in memory.

- A variable is stored in memory based on the amount of memory that is needed. Following are the steps followed to store a variable:
  - The required amount of memory is assigned first.
  - Then, it is stored based on the data structure being used.
- Using concepts like dynamic allocation ensures high efficiency and that the storage units can be accessed based on requirements in real time.

## Describe the types of Data Structures?

**i) Linear Data Structure:** A data structure that includes data elements arranged sequentially or linearly, where each element is connected to its previous and next nearest elements, is referred to as a linear data structure. Arrays and linked lists are two examples of linear data structures.

**ii) Non-Linear Data Structure:** Non-linear data structures are data structures in which data elements are not arranged linearly or sequentially. We cannot walk through all elements in one pass in a non-linear data structure, as in a linear data structure. Trees and graphs are two examples of non-linear data structures.

## What are the applications for stack data structure?

- i) It acts as temporary storage during recursive operations
- ii) Redo and Undo operations in doc editors
- iii) Reversing a string
- iv) Parenthesis matching
- v) Postfix to Infix Expressions
- vi) Function calls order

## What are different operations available in stack data structure?

- i) push:** This adds an item to the top of the stack. The overflow condition occurs if the stack is full.
- ii) pop:** This removes the top item of the stack. Underflow condition occurs if the stack is empty.
- iii) top:** This returns the top item from the stack.
- iv) isEmpty:** This returns true if the stack is empty else false.
- v) size:** This returns the size of the stack

## What is a Dequeue?

It is a double-ended queue, or a data structure, where the elements can be inserted or deleted at both ends (FRONT and REAR).



## What are the applications of queue Data Structure

To prioritize jobs as in the following scenarios:

- As waiting lists for a single shared resource in a printer, CPU, call center systems, or image uploads; where the first one entered is the first to be processed
- In the asynchronous transfer of data; or example pipes, file IO, and sockets
- As buffers in applications like MP3 media players and CD players
- To maintain the playlist in media players (to add or remove the songs)

## What is hashmap in data structure?

Hashmap is a data structure that uses an implementation of a hash table data structure which allows access to data in constant time ( $O(1)$ ) complexity if you have the key.

## What operations can be performed on queues?

- enqueue() adds an element to the end of the queue
- dequeue() removes an element from the front of the queue
- init() is used for initializing the queue
- isEmpty tests for whether or not the queue is empty
- The front is used to get the value of the first data item but does not remove it
- The rear is used to get the last item from a queue

## Where can stack Data Structure be used?

- Expression evaluation
- Backtracking
- Memory management
- Function calling and return

## What is the time complexity of basic operations get() and put() in HashMap class?

The time complexity is  $O(1)$  assuming that the hash function used in the hash map distributes elements uniformly among the buckets.

## **Explain the difference between file structure and storage structure?**

- **File Structure:** Representation of data into secondary or auxiliary memory say any device such as a hard disk or pen drives that stores data which remains intact until manually deleted is known as a file structure representation.
- **Storage Structure:** In this type, data is stored in the main memory i.e RAM, and is deleted once the function that uses this data gets completely executed.

The difference is that the storage structure has data stored in the memory of the computer system, whereas the file structure has the data stored in the auxiliary memory.

## **What is array data structure?**

An array data structure is a data structure that is used to store data in a way that is efficient and easy to access. It is similar to a list in that it stores data in a sequence. However, an array data structure differs from a list in that it can hold much more data than a list can.

An array data structure is created by combining several arrays together. Each array is then given a unique identifier, and each array's data is stored in the order in which they are created.

## **What is an asymptotic analysis of an algorithm?**

Asymptotic analysis of an algorithm defines the run-time performance as per its mathematical boundations. Asymptotic analysis helps us articulate the best case( $\Omega$  Notation,  $\Omega$ ), average case( $\Theta$  Notation,  $\theta$ ), and worst case( $O$  Notation,  $O$ ) performance of an algorithm.

## **Can we apply Binary search algorithm to a sorted Linked list?**

No, we cannot apply the binary search algorithm to a sorted linked list because finding the index of the middle element is difficult.



## **What is selection sort?**

In the selection sort technique, the list is divided into two parts. In one part all elements are sorted and in another part the items are unsorted. At first we take the maximum or minimum data from the array. After getting the data (say minimum) we place it at the beginning of the list by replacing the data of first place with the minimum data.

## **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 the merge sort? How does it work?**

Merge sort is a divide-and-conquer algorithm for sorting the data. It works by merging and sorting adjacent data to create bigger sorted lists, which are then merged recursively to form even bigger sorted lists until you have one single sorted list.

## **Define the graph Data Structure?**

It is a type of non-linear data structure that consists of vertices or nodes connected by edges or arcs to enable storage or retrieval of data. Edges may be directed or undirected.

## **What are the applications of graph Data Structure?**

- Transport grids where stations are represented as vertices and routes as the edges of the graph
- Utility graphs of power or water, where vertices are connection points and edge the wires or pipes connecting them
- Social network graphs to determine the flow of information and hotspots (edges and vertices)
- Neural networks where vertices represent neurons and edge the synapses between them.

## **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 the operations that can be performed on a data-structures?**

Following are the operations can be performed –

- Insertion: Adding a new data item.
- Deletion: Deleting the existing data item.
- Traversal: Accessing each data item.
- Searching: Finding a particular data item.
- Sorting: Arranging the data item in a particular sequence.

## **What do you understand by a binary search?**

A binary search is an algorithm that starts with searching in the middle element. If the middle element is not the target element then it further checks whether to continue searching the lower half of the higher half. The process continues until the target element is found.

## **What are some examples of divide-and-conquer algorithms?**

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

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