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 Ο(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

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 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 Ο(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