**Data Structures** is about how data can be stored in different structures.

**Algorithms** is about how to solve different problems, often by searching through and manipulating data structures.

Theory about Data Structures and Algorithms (DSA) helps us to use large amounts of data to solve problems efficiently.



* [Classification of Data Structure](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#classification)
  + [Linear Data Structure](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#linear)
    - [1) Static data structure](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#static)
    - [2) Dynamic data structure](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#dynamic)
  + [Non-linear Data Structure](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#nonlinear)
* [Arrays](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#array)
  + [Characteristics of an Array](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#arraychar)
  + [Applications of Array](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#arrayapp)
  + [Real-Life Applications of Array](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#arrayapprl)
* [Linked List](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#linkedlist)
  + [Characteristics of a Linked List](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#llchar)
  + [Applications of Linked List](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#llapp)
  + [Real-Life Applications of Linked List](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#llapprl)
* [Stack](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#stack)
  + [Characteristics of a Stack](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#stackchar)
  + [Applications of Stack](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#stackapp)
  + [Real-Life Applications of Stack](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#stackapprl)
* [Queue](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#queue)
  + [Characteristics of a Queue](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#queuechar)
  + [Applications of Queue](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#queueapp)
  + [Real-Life Applications of Queue](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#queueapprl)
* [Tree](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#tree)
  + [Characteristics of a Tree](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#treechar)
  + [Applications of Tree](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#treeapp)
  + [Real-Life Applications of Tree](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#treeapprl)
* [Graph](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#graph)
  + [Characteristics of a Graph](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#graphchar)
  + [Applications of Graph](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#graphapp)
  + [Real-Life Applications of Graph](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#graphapprl)
* [Conclusion](https://www.geeksforgeeks.org/what-is-data-structure-types-classifications-and-applications/#conclusion)

**Linear data structure:** Data structure in which data elements are arranged sequentially or linearly, where each element is attached to its previous and next adjacent elements, is called a linear data structure.

* **Static data structure:**Static data structure has a fixed memory size. It is easier to access the elements in a static data structure.   
  *An example of this data structure is an* ***array****.*
* **Dynamic data structure:**In the dynamic data structure, the size is not fixed. It can be randomly updated during the runtime which may be considered efficient concerning the memory (space) complexity of the code. *Examples of this data structure are* ***queue, stack****, etc.*

**Non-linear data structure:**Data structures where data elements are not placed sequentially or linearly are called non-linear data structures. In a non-linear data structure, we can’t traverse all the elements in a single run only.   
*Examples of non-linear data structures are* ***trees and graphs****.*

Need of DSA

Data Structures and Algorithms (DSA) is a fundamental part of computer science that's essential for many reasons, including:

DSA is about finding efficient ways to store and retrieve data, to perform operations on data, and to solve specific problems.

* Problem-solving: DSA teaches you how to think and solve complex problems systematically.
* Software development: DSA is the building block of software development, and is critical for creating scalable and high-performance software solutions.
* Technical interviews: DSA is a critical component of technical interviews, and mastering it can give you a significant advantage.
* Real-world applications: DSA is integral to real-world applications across various domains, from web development to data science.
* Program performance: Using the right data structure and algorithm makes your program run faster, especially when working with lots of data.
* Career success: DSA plays a pivotal role in shaping successful careers.

DATA STRUCTURES

1. **Arrays**: Simple, fixed-size collection of elements of the same type.
2. **Linked Lists**: A linear collection of nodes where each node contains data and a reference to the next node.
3. **Stacks**: LIFO (Last In First Out) data structure used for processing tasks in reverse order.
4. **Queues**: FIFO (First In First Out) structure, useful for task scheduling.
5. **Trees**: Hierarchical structures like Binary Trees, Binary Search Trees, AVL trees, used for searching and sorting.
6. **Graphs**: Nodes connected by edges, used to represent networks, with algorithms like DFS, BFS, Dijkstra's, etc.
7. **Hash Tables**: Used for efficient searching by converting keys into array indices.

ALGORITHMS

1. **Sorting**: Techniques like Bubble Sort, Merge Sort, Quick Sort to arrange elements in a specific order.
2. **Searching**: Linear Search, Binary Search, etc.
3. **Greedy Algorithms**: Optimizing a solution by choosing the best option at every step (e.g., Huffman Coding).
4. **Dynamic Programming**: Breaking problems into subproblems and storing results (e.g., Fibonacci sequence, Knapsack problem).
5. **Graph Algorithms**: Techniques like Kruskal's, Prim's, and Maximum Flow algorithms to solve network-related problems.