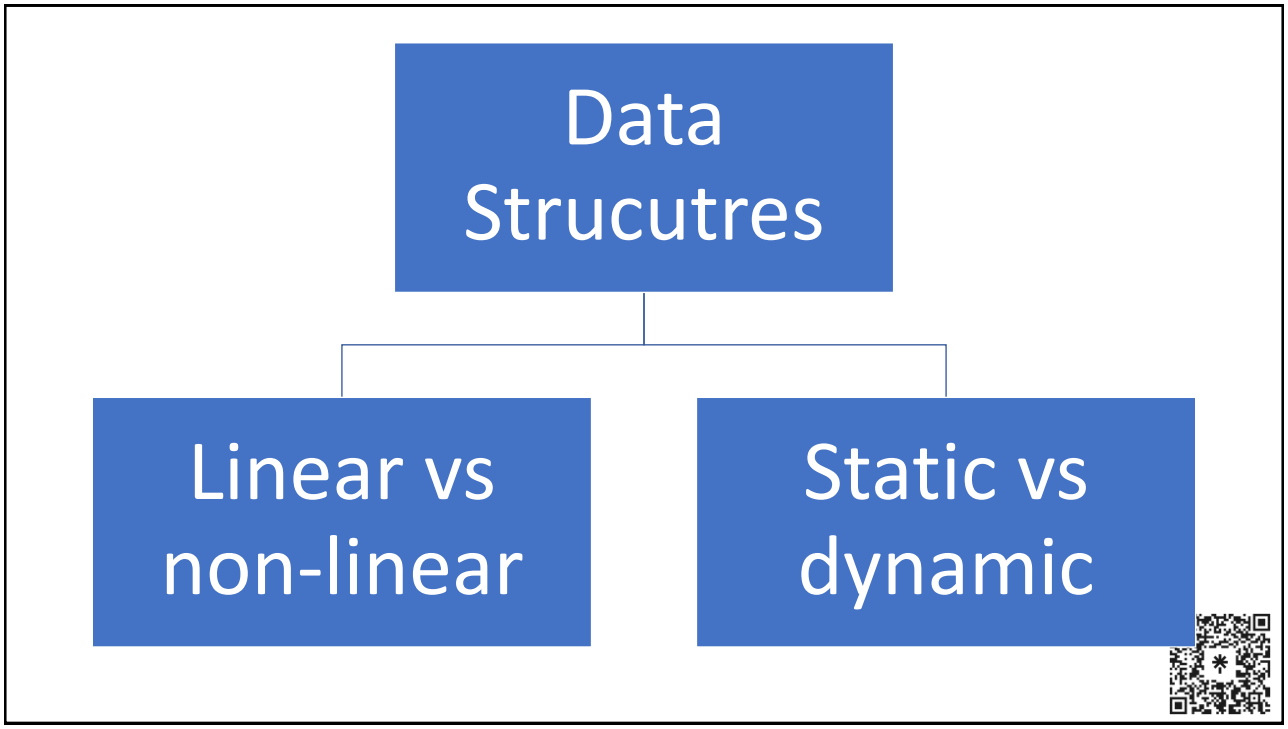
Introduction to DS

What is it?

- **A data structure is a storage that is used to store and organize data. It is a way of arranging data on a computer so that it can be accessed and updated efficiently.**



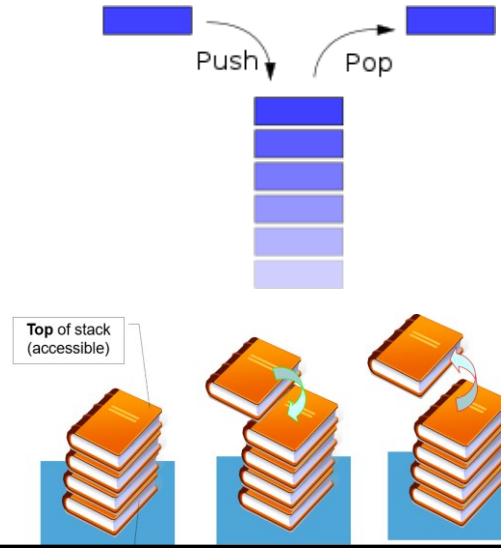
Linear Data Structure	Non-linear Data Structure
In a linear data structure, data elements are arranged in a linear order where each and every element is attached to its previous and next adjacent.	In a non-linear data structure, data elements are attached in hierarchically manner.
In linear data structure, single level is involved.	Whereas in non-linear data structure, multiple levels are involved.
Its implementation is easy in comparison to non-linear data structure.	While its implementation is complex in comparison to linear data structure.
In linear data structure, data elements can be traversed in a single run only.	While in non-linear data structure, data elements can't be traversed in a single run only.
In a linear data structure, memory is not utilized in an efficient way.	While in a non-linear data structure, memory is utilized in an efficient way.
Its examples are: array, stack, queue, linked list, etc.	While its examples are: trees and graphs.
Applications of linear data structures are mainly in application software development.	Applications of non-linear data structures are in Artificial Intelligence and image processing.

Static data structure	Dynamic data structure
1. A static data structure has a fixed and predefined size because of which it can store only a particular amount of data in the memory.	1. Dynamic data structures can store any amount of data in the memory since it does not have a fixed size.
2. The size of the structure cannot grow or shrink.	2. It can easily grow or shrink in size as per the need.
3. A static data structure can only modify the data in the memory.	3. A dynamic data structure can modify both the data items as well as the size of the structure during runtime.
4. They are not very efficient as compared to dynamic data structures as they have a fixed size.	4. The size of the dynamic data structures is randomly updated during runtime which is why it is considered more efficient.
5. Static memory allocation can only be done on the stack.	5. Dynamic memory allocation can be done on both stack and heap.
6. They are not as flexible as the dynamic data structures because of their fixed size.	6. The dynamic data structure is more flexible than the static data structures because of their easily growable and shrinkable size.
7. An example of a static data structure is an array.	7. An example of a dynamic data structure can be a linked list.

LIST OF DATA STUCTURES

- **STACK**
- **QUEUE**
- **LINKED LIST**
- **TREE**
- **GRAPH**

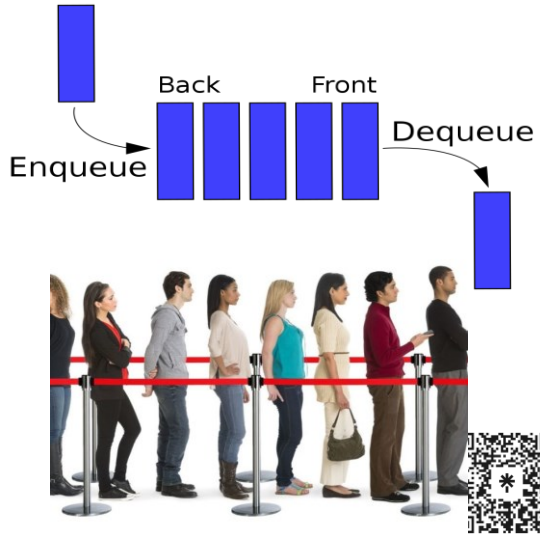
STACK



STACK

- **Common Application**
  - **Recursion**
  - **Checking balance of (),{},[]**
  - **Reversing string**
  - **Dec to binary conversion**
  - **Expression evaluation and conversion**

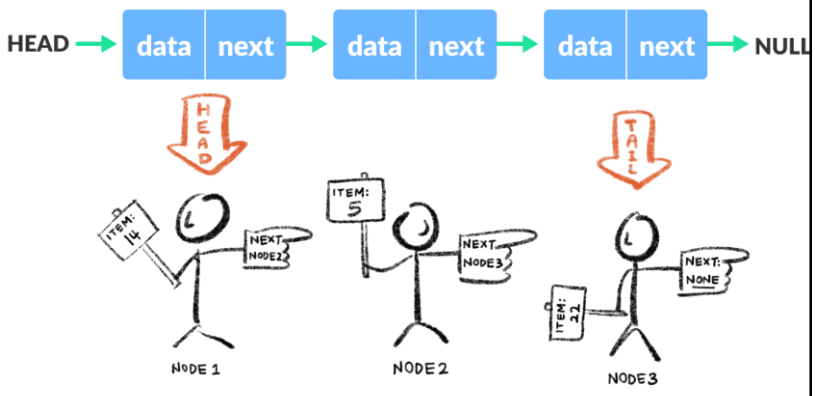
QUEUE



QUEUE

- **Common Application**
  - **Maintaining the playlist**
  - **Managing requests in scheduling and disk scheduling**
  - **Handling hardware or real-time systems interrupts**
  - **Handling website traffic**
  - **Routers and switches in networking**

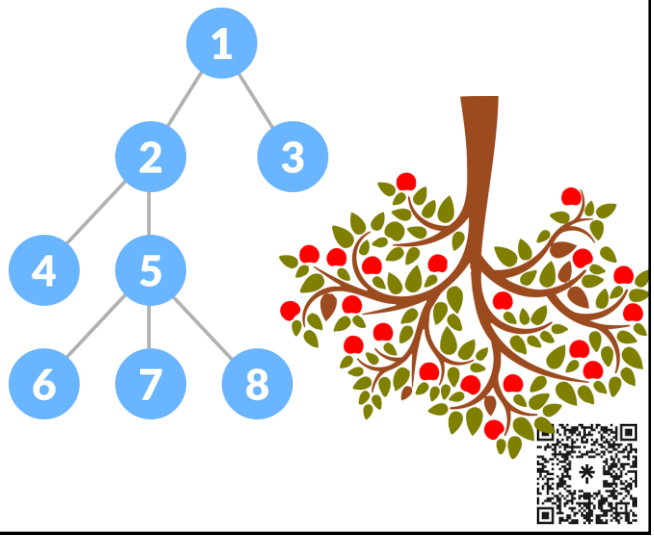
Linked List



Linked List

- **Common Application**
- **Implementation of stacks and queues**
- **Implementation of graphs : Adjacency list representation of graphs is most popular which is uses linked list to store adjacent vertices.**
- **Dynamic memory allocation :**
- **Maintaining directory of names**

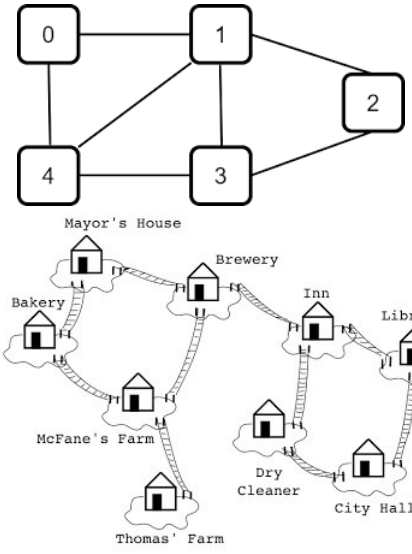
TREE



Tree

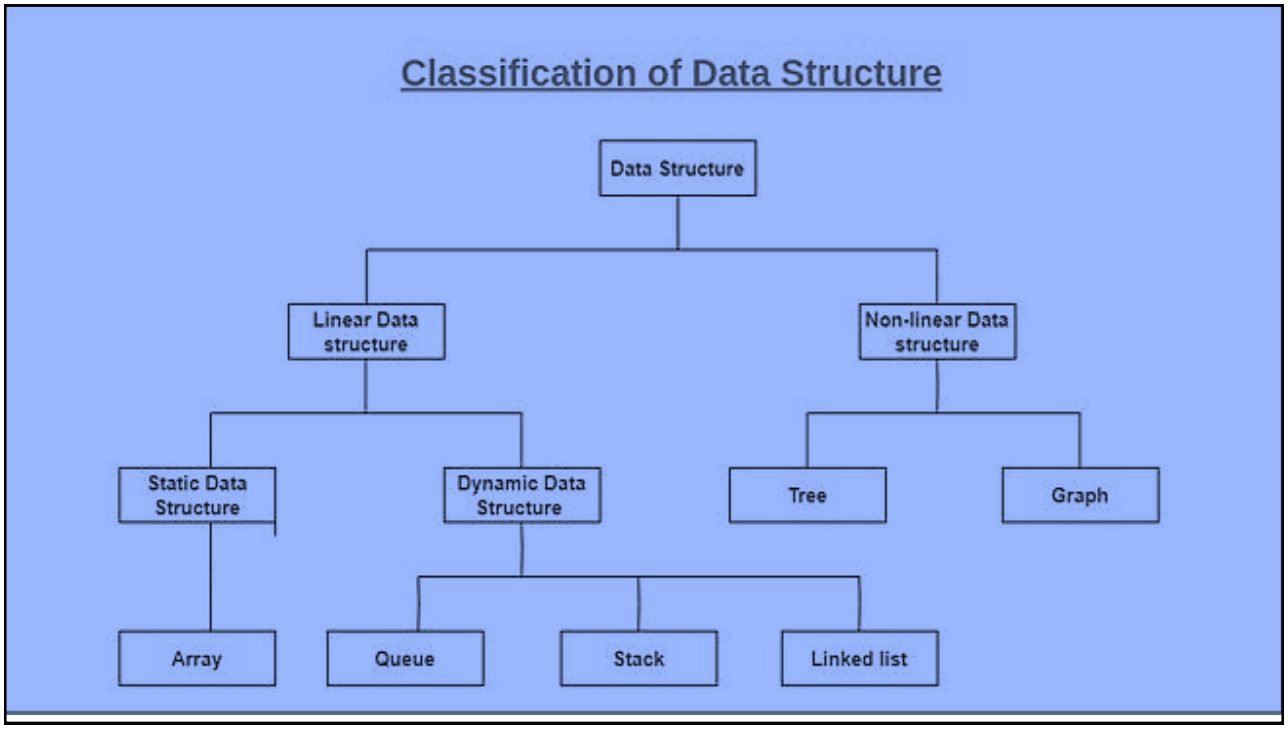
- **Common Application**
  - **Storing naturally hierarchical data**
  - **Organize data**
  - **Trie**
  - **Heap**
  - **B-Tree and B+Tree**

GRAPH



Graph

- **Common Application**
  - **Routing**
  - **Navigation**
  - **Biometric**
  - **Minimum Spanning tree**



ADT ?

- Abstract Data Types:
- Set of methods one must implement to realize/implement data structure
- Concentrates: on encapsulation and abstraction.
- Allow

Data Type	Data Structure
The data type is the form of a variable to which a value can be assigned. It defines that the particular variable will assign the values of the given data type only.	Data structure is a collection of different kinds of data. That entire data can be represented using an object and can be used throughout the program.
It can hold value but not data. Therefore, it is dataless.	It can hold multiple types of data within a single object.
The implementation of a data type is known as abstract implementation.	Data structure implementation is known as concrete implementation.
There is no time complexity in the case of data types.	In data structure objects, time complexity plays an important role.
In the case of data types, the value of data is not stored because it only represents the type of data that can be stored.	While in the case of data structures, the data and its value acquire the space in the computer's main memory. Also, a data structure can hold different kinds and types of data within one single object.
Data type examples are int, float, double, etc.	Data structure examples are stack, queue, tree, etc.