



CodeHype ✓
@codehype_

Data Structures

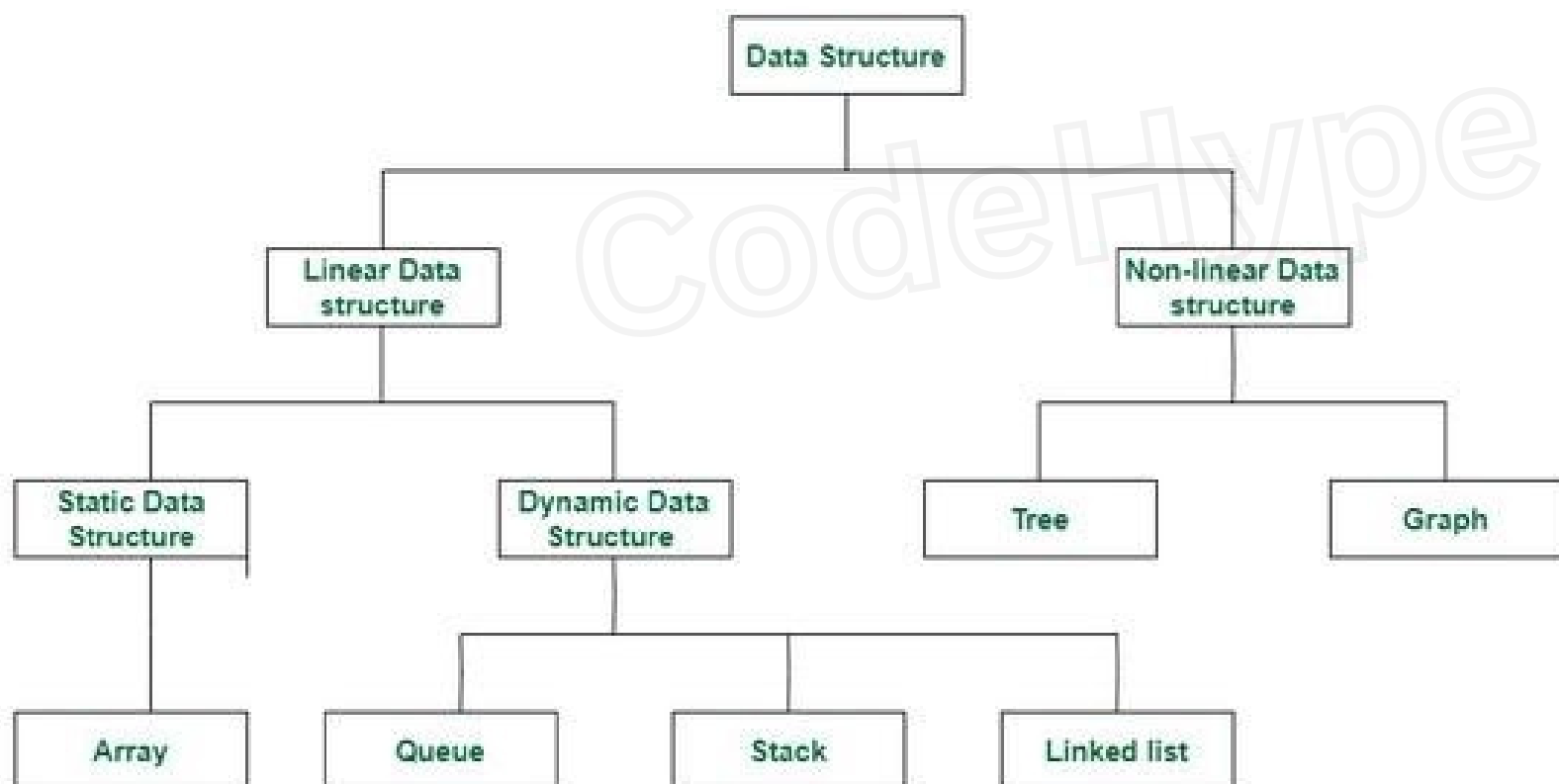
Explained!



What is Data Structure ?

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.

Classification of Data Structure



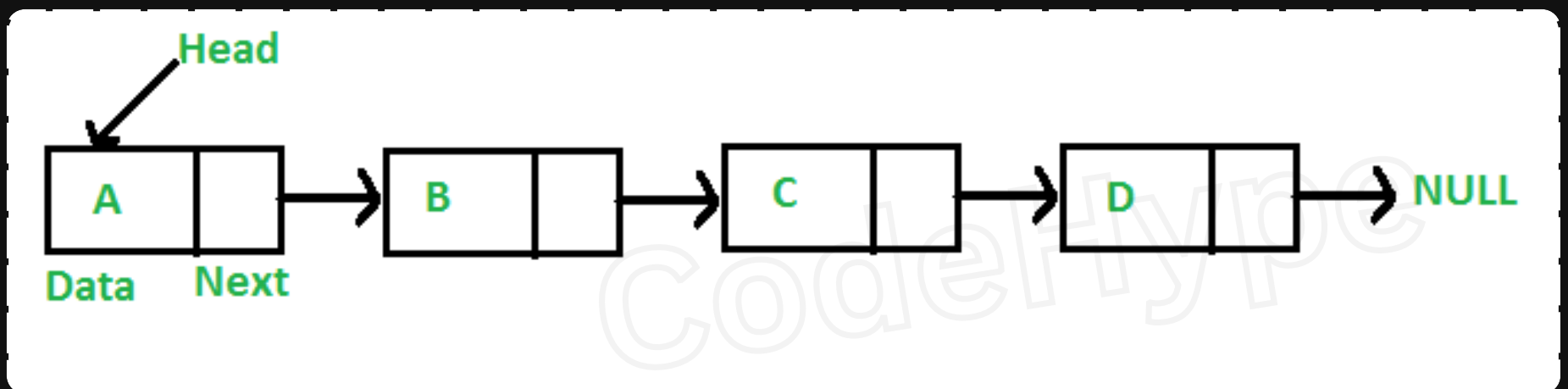
Arrays

A fundamental data structure consisting of a **collection of elements** stored in **contiguous** memory locations. Elements are accessed using an index, allowing for efficient random access but limited flexibility in size.

Memory Location									
200	201	202	203	204	205	206	■	■	■
U	B	F	D	A	E	C	■	■	■
0	1	2	3	4	5	6	■	■	■
Index									

Linked Lists

A linked list is a linear data structure in which elements are **not stored at contiguous** memory locations. The elements in a linked list are **linked using pointers** as shown in the below image:

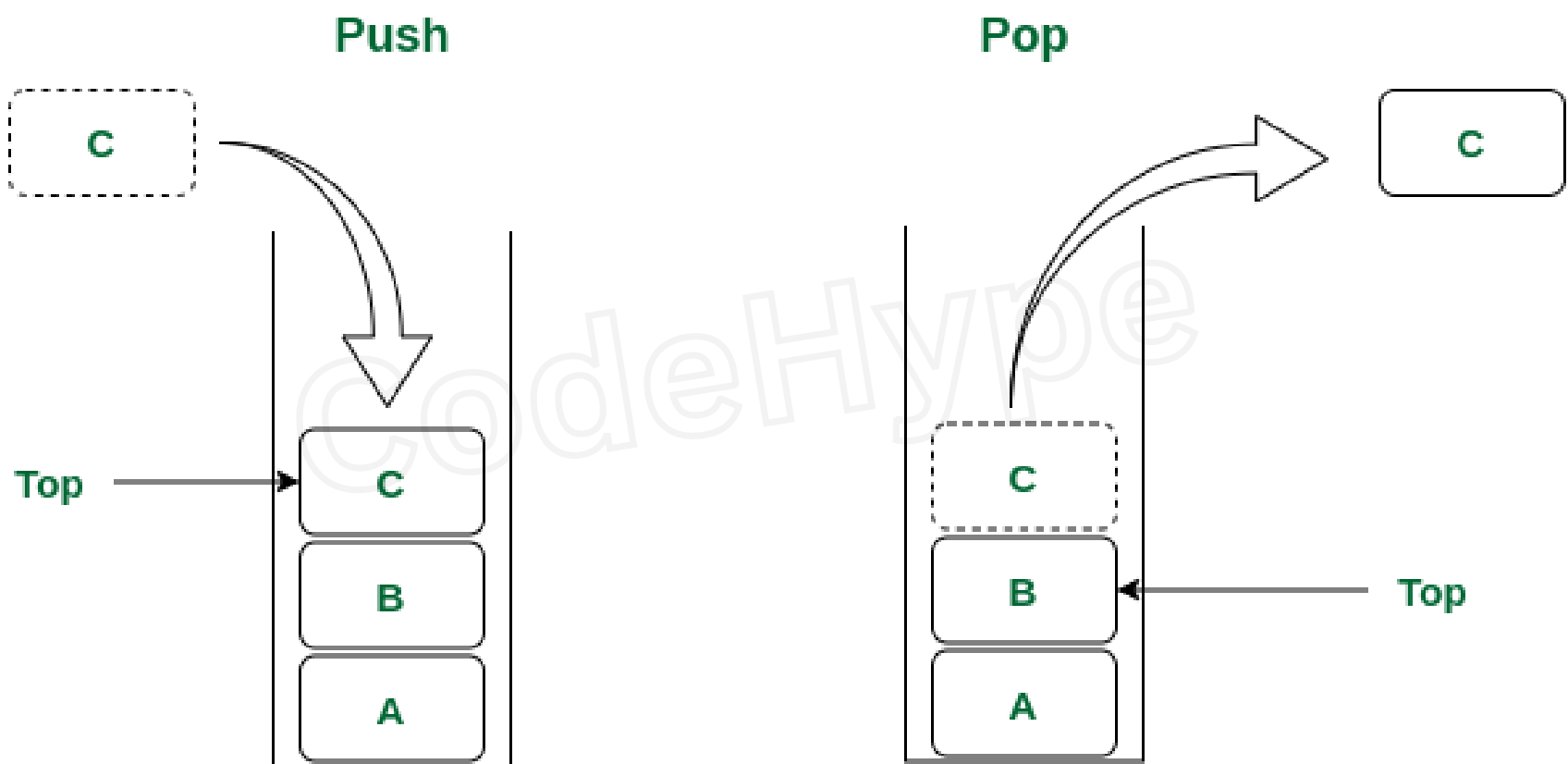


Types of linked lists:

- Singly-linked list
- Circular linked list
- Doubly linked list
- Doubly circular linked list

Stack

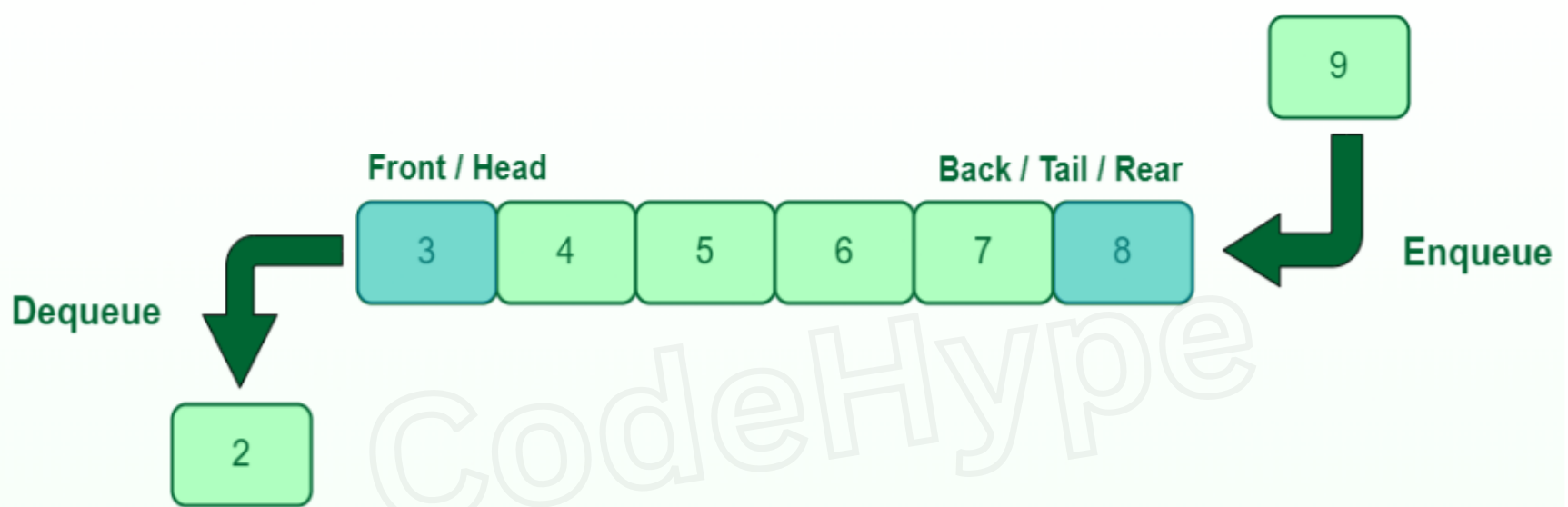
Stack is a linear data structure that follows a particular order in which the operations are performed. **The order is LIFO (Last in first out)**. Entering (Push) and retrieving (Pop) data is possible from only one end.



Stack Data Structure

Queue

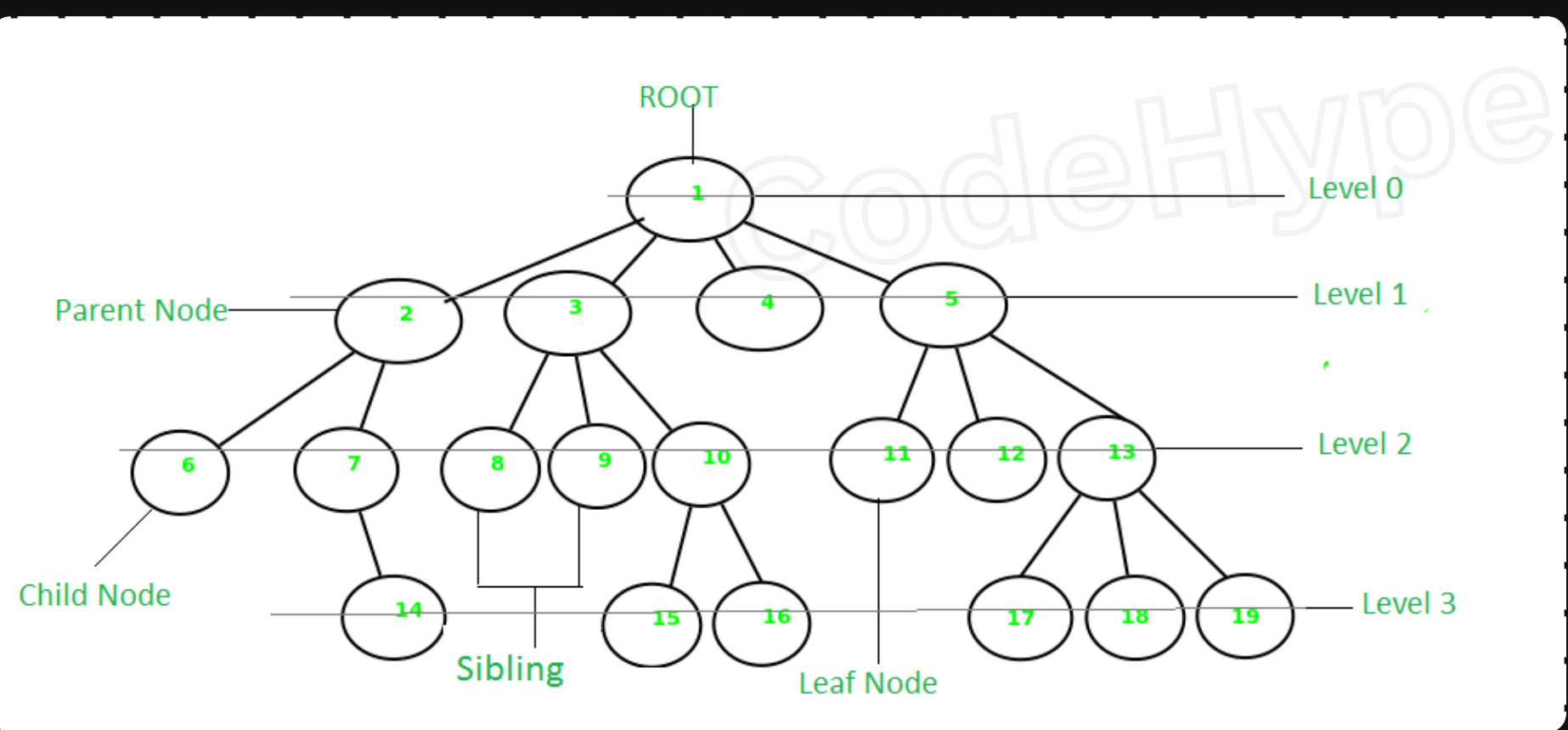
Queue is a linear data structure that follows a particular order in which the operations are performed. **The order is First In First Out(FIFO)** i.e. the data item stored first will be accessed first. In this, entering and retrieving data is not done from **only one end**.



Queue Data Structure

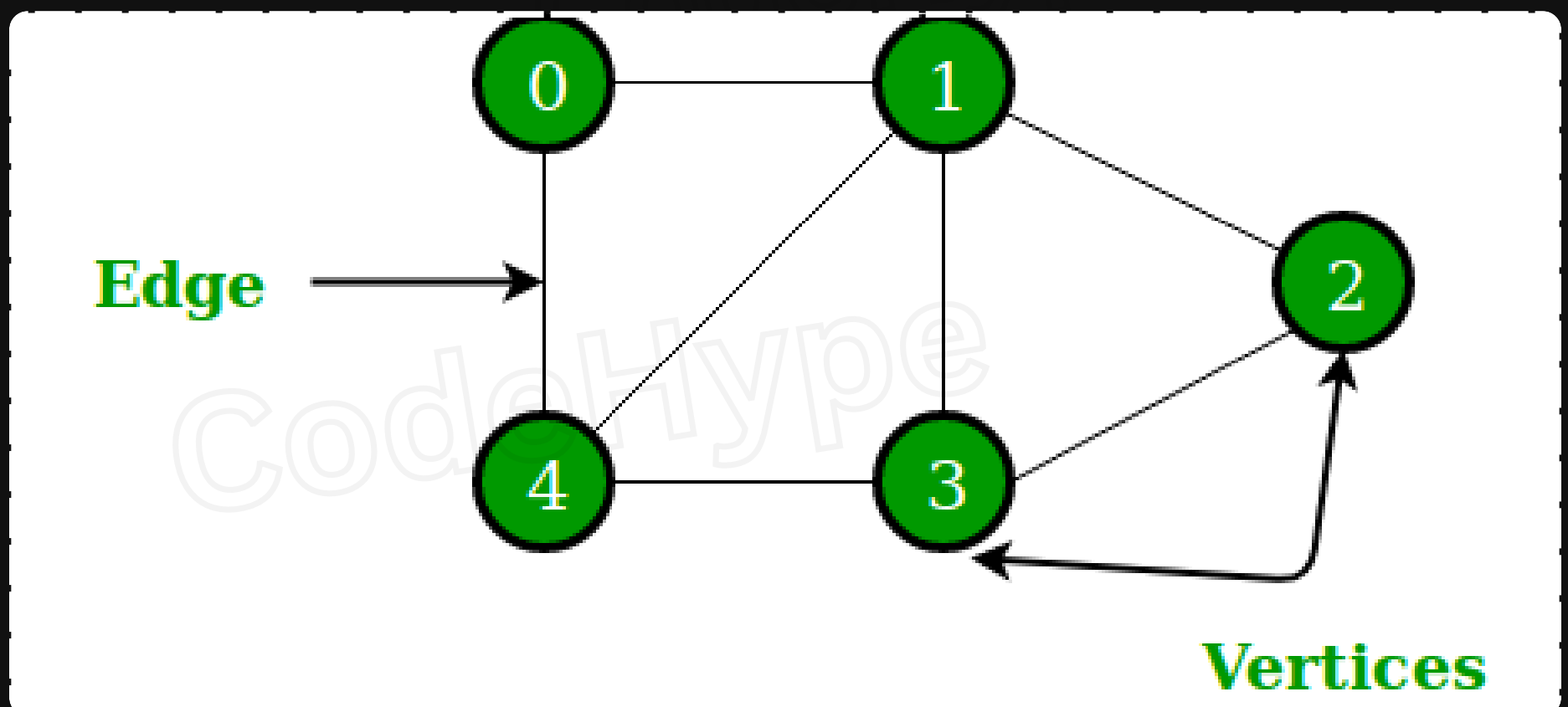
Tree

A tree is a **non-linear** and **hierarchical data** structure where the elements are arranged in a **tree-like structure**. In a tree, the topmost node is called the root node. Each node contains some data, and data can be of any type. It consists of a central node, **structural nodes**, and sub-nodes which are connected via edges.



Graph

A graph is a **non-linear data structure** that consists of vertices (or nodes) and edges. It consists of a finite set of vertices and set of edges that connect a pair of nodes. The graph is used **to solve the most challenging and complex programming problems**. It has different terminologies which are Path, Degree, Adjacent vertices, Connected components, etc.



Conclusion

While these are the commonly utilized data structures, there are other types like **policy-based data structures** in Computer Science. Regardless of the choice, each structure has its benefits and drawbacks.

Selecting the appropriate one is crucial to avoid costly mistakes. Understanding the situation's requirements helps in making the **best-suited data structure** decision.



Follow for more useful content



Telegram @codehype

Join Telegram for
**FREE Coding
Resources**

 [CodeHype](https://www.instagram.com/codehype)

