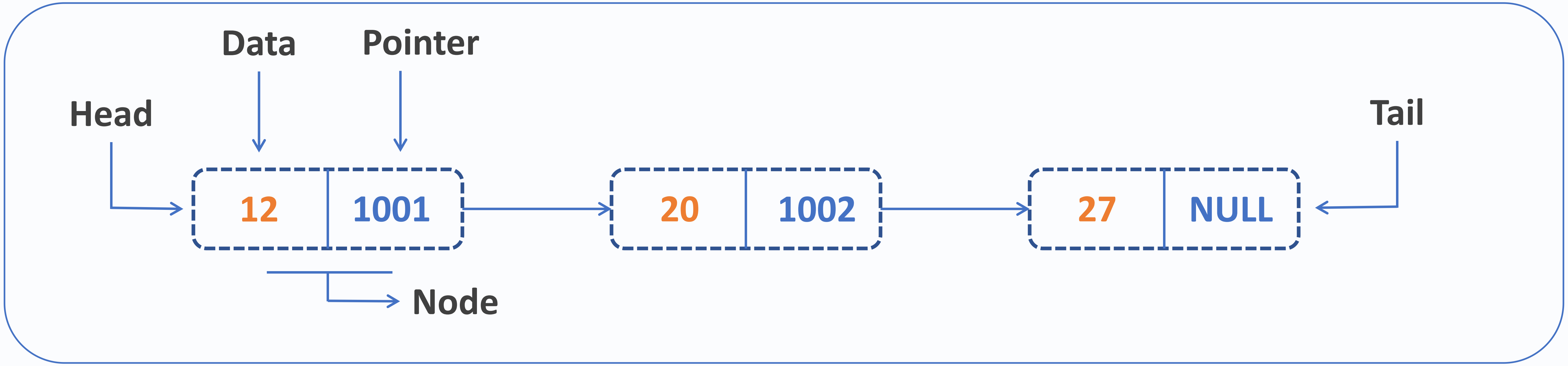
Introduction to Linked Lists

Linked lists are fundamental data structures used to store and organize data in a linear fashion.

They consist of nodes, each containing data and a pointer to the next node, forming a chain-like structure.

Linked lists offer flexibility and efficiency compared to arrays, enabling dynamic memory allocation and insertion/deletion operations in constant time.



Representing Linked Lists

1

Head

The head pointer references the first node.

2

Node

Each node stores data and a pointer.

3

Pointer

The pointer points to the next node in the list.

4

Tail

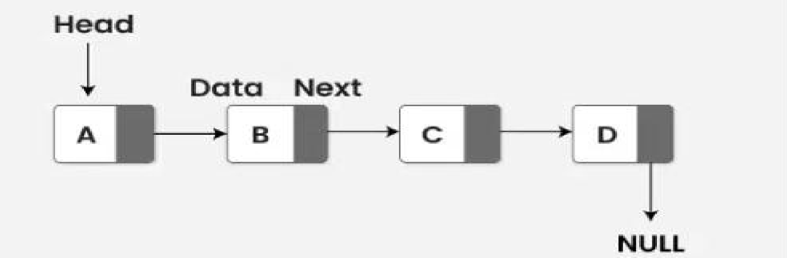
The tail pointer references the last node.

Types and Applications of Linked Lists

Single Linked List

A single linked list connects nodes in a linear fashion, with each node pointing to the next.

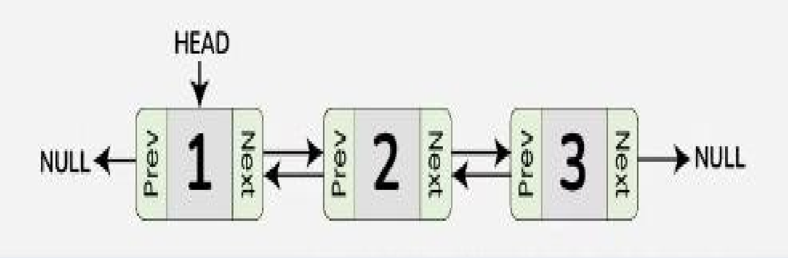
This type is simple and efficient for sequential access but requires traversal from the head for reaching specific nodes.



Doubly Linked List

Each node in a doubly linked list has pointers to both the previous and next nodes.

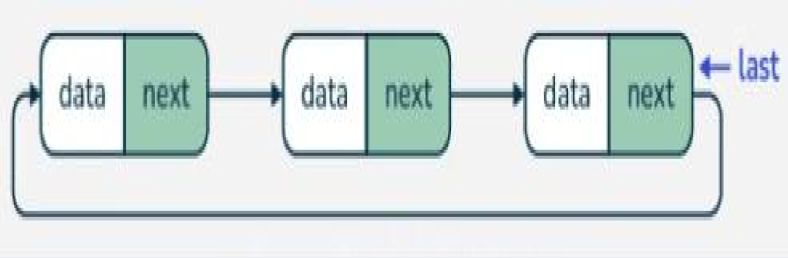
This allows for bidirectional traversal and faster insertion/deletion operations compared to single linked lists.



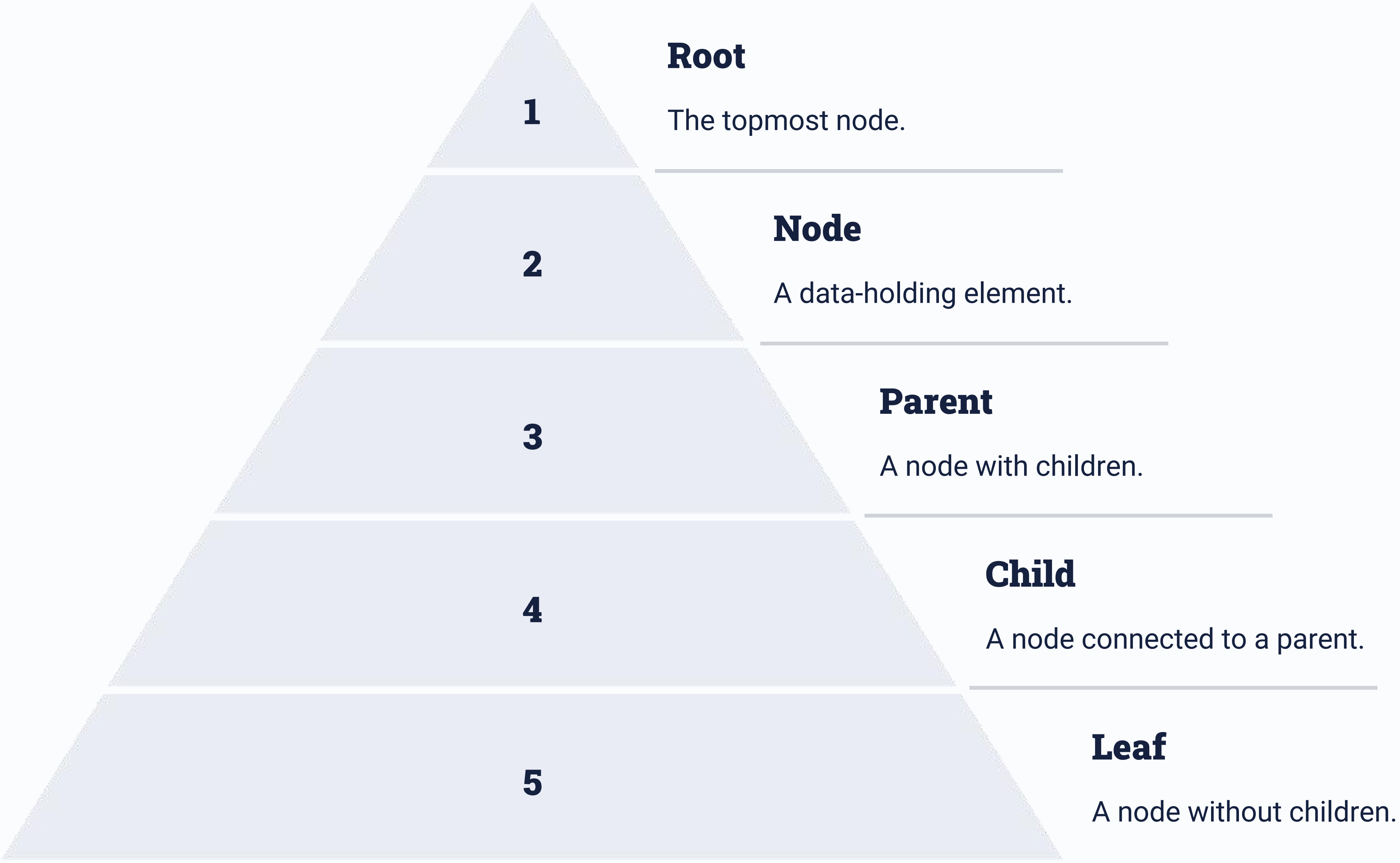
Circular Linked List

In a circular linked list, the last node points back to the first node, creating a circular structure.

This allows for traversal in a loop, making it useful for managing queues and other circular data structures.



Introduction to Trees



1 **Root**

The topmost node.

2 **Node**

A data-holding element.

3 **Parent**

A node with children.

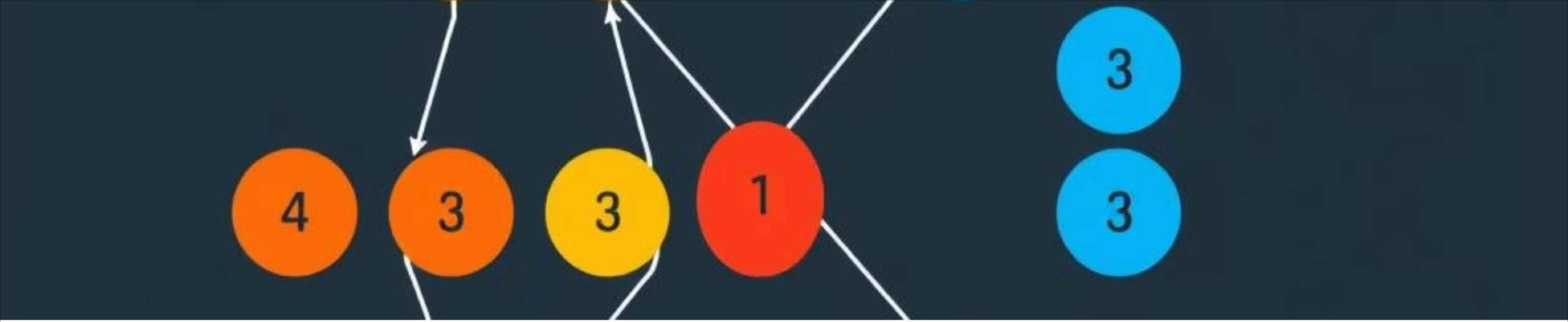
4 **Child**

A node connected to a parent.

5 **Leaf**

A node without children.

Trees are hierarchical data structures that organize information in a branching fashion.
They consist of nodes connected by edges, forming a tree-like structure with a single root node.



Representing and Traversing Trees

1

Pre-order Traversal

Visit the root node first, then the left subtree, then the right subtree.

2

In-order Traversal

Visit the left subtree first, then the root node, then the right subtree.

3

Post-order Traversal

Visit the left subtree first, then the right subtree, then the root node.

Applications of Trees

Binary Search Trees

Binary search trees are used for efficient searching and sorting. They quickly find data based on key values, which is useful for databases and search engines.

Decision Trees

Decision trees help predict outcomes. They show how variables relate to each other, useful for classifying data in machine learning and predictions.

File Systems

File systems use trees to organize files and folders in a hierarchy. This makes it easy to manage and access data.

Network Structures

Computer networks can be represented as trees. Devices are nodes and connections are branches. This helps with routing and communication.