

TASK 6

Name: Sami Imran

Roll No: 033

1. BFS Without Queue & Without Node

Breadth-First Search (BFS) is a tree/graph traversal algorithm that explores all nodes at the current level before moving to the next level. Normally, BFS uses a **queue** to keep track of nodes, but if we remove the queue, BFS will not function correctly as intended.

However, BFS **without a queue** can still be implemented recursively by using a list of nodes at each level. The idea is to process nodes **level by level** using a recursive function that keeps track of the current depth.

Since we are also removing the **Node** structure, we assume we have a data structure that allows us to access child nodes directly, such as an **adjacency list in graphs** or an array-based tree representation.

This method is inefficient because recursion depth increases with tree height, and tracking levels manually is complex. A queue is generally preferred for BFS.

2. BFS With Queue & Node

A **proper BFS implementation** requires a **queue** and a **node-based structure**. The process follows these steps:

1. **Start with the root node** and place it in a queue.
2. **Dequeue the front node**, process it, and enqueue its children.
3. **Repeat until the queue is empty**, ensuring nodes are visited level by level.

This approach ensures efficient traversal of trees and graphs while keeping track of visited nodes to avoid infinite loops in cyclic graphs. Using a **queue** guarantees that nodes are explored in the correct order without excessive recursion depth. BFS is commonly used in **shortest path algorithms**, **network broadcasting**, and **AI pathfinding** (e.g., in games or navigation systems).

