BFS without Queue

This code implements a Breadth-First Search (BFS) algorithm without using a queue or node structure. Instead, it uses recursion to traverse the graph.

1. **Graph Representation**: The graph is represented as a dictionary where each key is a node, and the value is a list of its neighboring nodes

2. **Visited List**: A list named `visited` is used to keep track of the nodes that have already been visited during the traversal.

3. **Recursive BFS Function**: The `bfs\_recursive` function performs the BFS traversal.

* It takes three parameters: `graph`, `node`, and `goal`.
* The current node is added to the `visited` list and printed.
* If the current node is the goal node, it prints a message and returns `True`.
* It recursively visits each unvisited neighbor of the current node.

4. **Execution**: The BFS traversal starts from node 'A' and searches for node 'E'.

The code prints the nodes as they are visited and stops when the goal node is found.