BFS with Queue

This code implements a Breadth-First Search (BFS) algorithm using a queue and a node structure.

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 and Queue**: Two lists are used:

* `visited` to keep track of the nodes that have already been visited.
* `queue` to manage the nodes to be explored.

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

* It takes four parameters: `visited`, `graph`, `node`, and `goal`.
* The starting node is added to both `visited` and `queue`.
* The function enters a loop that continues until the queue is empty.
* In each iteration, the first node (`m`) is removed from the queue and printed.
* If `m` is the goal node, it prints a message and returns.
* Otherwise, it adds all unvisited neighbors of `m` to both `visited` and `queue`.

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.