***BFS***

Simple Python implementation of Breadth-First Search (BFS) algorithm for graph traversal.

* BFS is an efficient algorithm for finding the shortest path between two nodes in an unweighted graph. Breakdown of the code

**1) Initialization:**

* **graph:** This dictionary represents the graph. Keys are nodes, and their corresponding values are lists of their adjacent nodes.
* **start:** This variable stores the starting node for the BFS traversal.
* **queue:** A deque (double-ended queue) is used to efficiently add and remove nodes during the traversal. Initially, it contains only the start node.
* **visited:** A set is used to keep track of nodes that have already been visited, preventing cycles and ensuring each node is explored only once.

**2) Breadth-First Search (BFS) Loop:**

* The **while queue loop** continues as long as there are nodes to explore in the queue.
* **node = queue.popleft() :** In each iteration, the node at the front of the queue is dequeued (removed from the front) and processed.
* **print(node, end=" ") :** The current node is printed. (This is for visualization and may not be part of the core BFS algorithm)
* **for neighbor in graph[node]:** The code iterates through each neighbor of the current node.
  + **if neighbor not in visited:** If the neighbor has not been visited before:
  + **visited.add(neighbor):** The neighbor is added to the visited set.
  + **queue.append(neighbor):** The neighbor is added to the end of the queue for exploration later.

**OUTPUT :** A B C D E F

MADE BY – SWAGNIK SHIT

REG NO. - 2301292134