Breadth-First Search (BFS)

1. A class named BFS is created to encapsulate the graph and the traversal process.

2. Inside the class, the \_\_init\_\_ method takes the graph as input and initializes an empty 'visited' list.

3. The 'search' method performs BFS starting from a given node and looks for a target node (goal).

4. A list called 'level' keeps track of the current nodes being explored.

5. For each node in the current level: if it has not been visited, it is marked as visited.

6. If the node is the target, the search stops and 'Found' is printed.

7. If not, all of its children are added to 'next\_level', which represents the nodes to be explored in the next round.

8. After finishing the current level, the search continues with the next level until either the goal is found or all nodes are explored.

9. The 'get\_visited' method is used to return the order in which nodes were visited during the search.

Bfs (with queue)

1. A class named 'BreadthFirstSearch' is defined. It stores the graph and a list of visited nodes (called 'traversed').

2. The '\_\_init\_\_' method takes the graph as input and prepares an empty list for traversal history.

3. The 'run' method begins the search from a given start node towards a goal node.

4. A queue is used to keep track of nodes to explore. Initially, the start node is added to the queue.

5. While the queue is not empty, the first node is removed and checked.

6. If this node has not been visited, it is added to the traversed list.

7. If the node is the goal, the search prints 'Target Found' and ends.

8. If not the goal, all children of the current node are added to the queue for later exploration.

9. If the queue becomes empty without finding the goal, the program prints 'Target Not Found'.

10. The 'get\_traversed' method returns the full order of nodes that were visited during the search.