**Breadth-First Search DFS**

**Iheb Gafsi**\*

INSAT Student

**Iheb.engineer@gmail.com**

**Definition:**

Breadth-First Search (BFS) is a graph traversal algorithm that explores neighboring nodes first before moving to their children. It ensures the shortest path to each node in an unweighted graph. BFS uses a queue to maintain the order of node exploration, avoiding revisiting nodes and ensuring each node is visited only once.

**Use cases:**

Breadth-First Search (BFS) is a versatile graph traversal algorithm widely used to find the shortest path, discover connections, and explore networks efficiently. It is crucial in route planning, social networking, web crawling, and puzzle-solving tasks, making it an essential tool for exploring and analyzing various structures and relationships in different fields.

**Algorithm:**

1. # Variables

2. graph = Adjacency List

3.

4. def bfs(graph, start):

5. visited = set()

6. queue = [start]

7.

8. while queue:

9. node = queue.pop(0)

10. if node not in visited:

11. print(node, end=' ')

12. visited.add(node)

13. queue.extend(graph[node])

14.

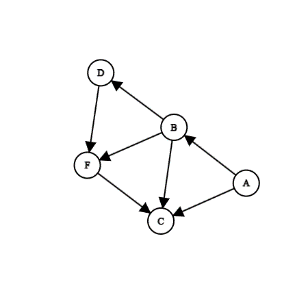
15. # Starting the BFS from node n

16. bfs(graph, n)

17.

**Example:**

Here’s a small example illustrating an example of input outputs for the BFS:



We will use the Python code down below to outline the output of the algorithm on this graph:

1. # Variables

2. graph = {

3. 'A': ['B', 'C'],

4. 'B': ['C', 'D', 'F'],

5. 'D': ['F'],

6. 'F': ['C']

7. }

8.

9. def bfs(graph, start):

10. visited = set()

11. queue = [start]

12.

13. while queue:

14. node = queue.pop(0)

15. if node not in visited:

16. print(node, end=' ')

17. visited.add(node)

18. if node in graph.keys():

19. queue.extend(graph[node])

20.

21. # Starting the BFS from node 'A'

22. bfs(graph, 'A')

23.

The corresponding output is:

Python >> A | B | C | D | F |

