Experiment 1

Student Name: Rahul Saxena

UID: 24MCI10204

Branch: MCA (AI-ML)

Section/Group: MAM - 3(B)

Semester: II

Subject Name: Artificial Intelligence Lab (24CAP-674)

Aim : Implement Breadth-First Search (BFS) Algorithm.

Definition: Breadth-First Search (BFS) is a graph traversal algorithm that explores all the nodes at the current depth level before moving to the next depth level, using a queue for tracking.

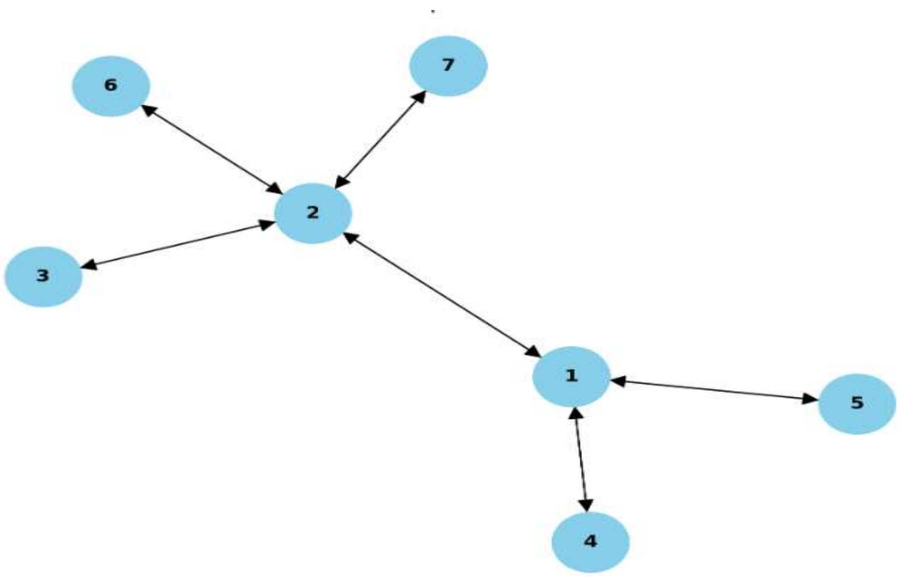
Steps:

1. **Start with the Source Node**: Choose a starting node and mark it as visited.
2. **Initialize a Queue**: Add the starting node to a queue to track nodes for exploration.
3. **Explore Nodes Level by Level**:
    a. Remove the first node from the queue (dequeue).
    b. Visit its unvisited neighbours, mark them as visited, and enqueue them.
4. **Repeat Until Queue is Empty**: Continue the process until there are no nodes left in the queue.

Algorithm:

1. **Initialize**:
    - Create an empty queue QQQ.
    - Mark all vertices as unvisited.
    - Mark the starting vertex sss as visited.
    - Enqueue sss into QQQ.
2. **While QQQ is not empty**:
    - Dequeue a vertex uuu from QQQ.
    - Process uuu (e.g., print it, store it in a list, etc.).
    - For each unvisited neighbor vvv of uuu:
        - Mark vvv as visited.
        - Enqueue vvv into QQQ.
3. **End**.

Graph:

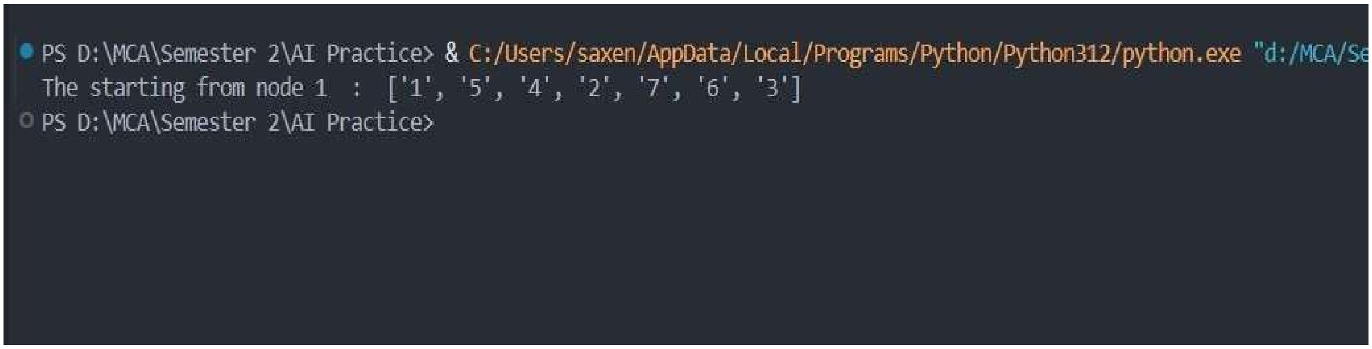


Code:

```
    from collections import deque
    graph = {
      "1": ["5", "4", "2"],
      "2": ["7", "6", "1", "3"],
      "3": ["2"],
      "4": ["1"],
      "5": ["1"],
      "6": ["2"],
      "7": ["2"]
    }

    StartNode = "1"
    def BFS(graph, startNode):
      visited = set()
      queue = deque([startNode])
bfs_order = []
while queue:
        node = queue.popleft()
        if node not in visited:
          visited.add(node)
          bfs_order.append(node)
          for neighbour in graph[node]:
            if neighbour not in visited:
              queue.append(neighbour)
      return bfs_order
    BFS_result = BFS(graph, StartNode)
    print("The starting from node", StartNode, " : ", BFS_result)
```

Output:



```
PS D:\MCA\Semester 2\AI Practice> & C:/Users/saxen/AppData/Local/Programs/Python/Python312/python.exe "d:/MCA/Se
The starting from node 1  :  ['1', '5', '4', '2', '7', '6', '3']
PS D:\MCA\Semester 2\AI Practice>
```

Learning Outcome:

1. Understand the systematic traversal of graphs using Breadth-First Search.
2. Learn the role of a queue data structure in implementing BFS.
3. Identify how BFS explores nodes level by level and ensures complete traversal.
4. Gain insight into applying BFS in real-world scenarios like shortest path problems, network analysis, and connectivity checks.
5. Develop a better understanding of graph theory concepts, such as vertices, edges, and adjacency.