In [4]:

```
#KETHARNATH R 111723102089 CSE C 2(A)

from collections import defaultdict  # Ensure defaultdict is imported

class Graph:
    def __init__(self):
        """Initialize a graph with an adjacency list."""
        self.graph = defaultdict(list)  # Correctly initializes the adjacency li

    def addEdge(self, u, v):
        """Adds a directed edge from node u to node v."""
        self.graph[u].append(v)

    def BFS(self, start):
        """Performs Breadth-First Search (BFS) from the given start node."""
        visited = set()  # Using a set to track visited nodes
        queue = [start]  # BFS queue

        visited.add(start)  # Mark start node as visited

        while queue:
            node = queue.pop(0)  # Dequeue
            print(node, end=" ")

            # Enqueue all unvisited adjacent nodes
            for neighbor in self.graph[node]:
                if neighbor not in visited:
                    queue.append(neighbor)
                    visited.add(neighbor)

# Driver code
g = Graph()
g.addEdge(0, 1)
g.addEdge(0, 2)
g.addEdge(1, 2)
g.addEdge(2, 0)
g.addEdge(2, 3)
g.addEdge(3, 3)

print("Following is Breadth First Traversal (starting from vertex 2):")
g.BFS(2)
```

```
Following is Breadth First Traversal (starting from vertex 2):
2 0 3 1
```

In [ ]: