Lab Assignment – 11

Here are the 2 functions that you are supposed to write with the following signatures. Please follow them accordingly. No class is required. You can use as many helper functions as you want for recursion

Function 1: def dfs(graph, start)

Parameters:

graph = undirected graph represented as adjacency list (List of Lists)
start = Starting node for traversal (integer)

Returns:

result = A list containing the nodes of depth first traversal order.

Function 2: def bfs(graph, start)

Parameters:

graph = undirected graph represented as adjacency list (List of Lists)
start = Starting node for traversal (integer)

Returns:

result = A list containing the nodes of breadth first traversal order.

Following are the sample test cases which you can try out locally for testing.

graph = [[1], [0, 2, 3], [1], [1]]
start_node = 0
expected_dfs_result = [0, 1, 2, 3]
expected_bfs_result = [0, 1, 2, 3]

---

graph = [[1, 2], [0, 3], [0], [1, 4, 5], [3], [3]]
start_node = 0
expected_dfs_result = [0, 1, 3, 4, 5, 2]
expected_bfs_result = [0, 1, 2, 3, 4, 5]

graph = [[1], [0, 2], [1], [], [5], [4]]
start_node = 0
expected_dfs_result = [0, 1, 2]
expected_bfs_result = [0, 1, 2]

---

graph = [[1, 2, 3], [0], [0], [0]]
start_node = 0
expected_dfs_result = [0, 1, 2, 3]
expected_bfs_result = [0, 1, 2, 3]

---

graph = [[1, 2], [0, 3], [0, 4], [1], [2]]
start_node = 0
expected_dfs_result = [0, 1, 3, 2, 4]
expected_bfs_result = [0, 1, 2, 3, 4]

---

graph = [[1, 2], [0, 3, 4], [0], [1], [1, 5],[4]]
start_node = 0
expected_dfs_result = [0, 1, 3, 4, 5, 2]
expected_bfs_result = [0, 1, 2, 3, 4, 5]