**Labsheet 5**

**BFS**

from queue import Queue

adj\_list = {

    "0": ["1", "2"],

    "1": ["0", "3", "4"],

    "2": ["0"],

    "3": ["1"],

    "4": ["2","3"],

}

visited = {}    # dict

parent = {}

bfs\_traversal\_order =[]  # lst

queue = Queue()

for node in adj\_list.keys():

    visited[node] = False

    parent[node] = None

s = "0"

visited[s] = True

queue.put(s)

while not queue.empty():

    u = queue.get()

    bfs\_traversal\_order.append(u)

    for v in adj\_list[u]:

        if not visited[v]:

            visited[v] = True

            parent[v] = u

            queue.put(v)

print("BFS Traversal Order  : ")

print(bfs\_traversal\_order)

destination = "4"

path = []

while destination is not None:

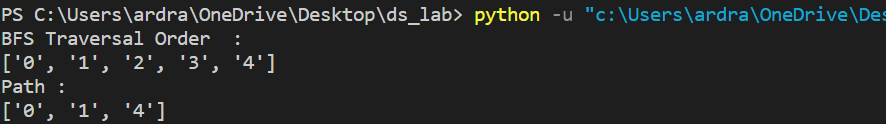
    path.append(destination)

    destination = parent[destination]

path.reverse()

print("Path : ")

print(path)

****

**DFS**

adj\_list = {

    "0": ["1", "2"],

    "1": ["0", "3", "4"],

    "2": ["0"],

    "3": ["1"],

    "4": ["2","3"],

}

color = {}

parent = {}

dfs\_traversal\_order = []

for node in adj\_list.keys():

    color[node] = "W"

    parent[node]= None

def dfs\_util(u):

    color[u] =  "G"

    dfs\_traversal\_order.append(u)

    for v in adj\_list[u]:

        if color[v] == "W":

            parent[v] = u

            dfs\_util(v)

    color[u] = "B"

dfs\_util("0")

print("dfs\_traversal\_order :")

print(dfs\_traversal\_order)

v = "2"

path = []

while v is not None:

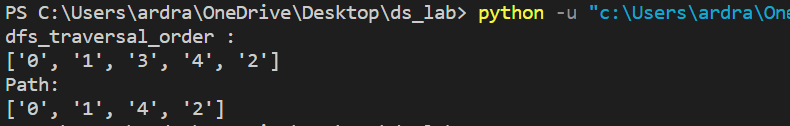
    path.append(v)

    v = parent[v]

path.reverse()

print("Path: ")

print(path)

****