Depth First Search:

DFS1:

```
#DFS non recursive
def dfs_non_recursive(graph, source):
    if source is None or source not in graph:
        return 'Invalid Input'
    path = []
    stack = [source]
    while (len(stack) != 0):
        s = stack.pop()
        if s not in path:
            path.append(s)
        if s not in graph:
            continue
        for neighbor in graph[s]:
            stack.append(neighbor)
    return ' '.join(path)

def main():
    graph = {
        'A': ['D', 'C', 'B'],
        'B': ['E'],
        'C': ['F', 'G'], #The example has G, F but that seems w
rong.
        'D': ['H'],
        'E': ['I'],
        'F': ['J']
    }

    path = dfs_non_recursive(graph, 'A')
    print(path)
main()
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

```
PS C:\Users\fagge\Documents\CS591> python -u "c:\Users\fagge\Documents\CS591\DFS1.py"
A B E I C G F J D H
PS C:\Users\fagge\Documents\CS591>
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

DFS2:

```
#DFS Recursive
def dfs_recursive(graph, source, path = []):
    if source not in path:
        path.append(source)

        if source not in graph:
            return path
        for neighbor in graph[source]:
            path = dfs_recursive(graph, neighbor, path)
    return path

def main():
    graph = {
        'A': ['D', 'C', 'B'],
        'B': ['E'],
        'C': ['F', 'G'], #The example has G, F but that seems w
rong.
        'D': ['H'],
        'E': ['I'],
        'F': ['J']
    }

    path = dfs_recursive(graph, 'A')
    print(*path)
main()
```

```
PS C:\Users\fagge\Documents\CS591> python -u "c:\Users\fagge\Documents\CS591\DFS2.py"
A D H C F J G B E I
PS C:\Users\fagge\Documents\CS591>
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

DFS3:

```
#Binary Tree DFS
class Node:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

class Tree:
    def __init__(self, value):
        self.root = Node(value)

    def insert(self, value):
        current = self.root
        while True:
            if value > current.value:
                if current.right is None:
                    current.right = Node(value)
                    break
                else:
                    current = current.right
            elif value < current.value:
                if current.left is None:
                    current.left = Node(value)
                    break
                else:
                    current = current.left
            else:
                current.value = value
                break

    def DFSTree(self, node: Node):
        if node is None:
            return
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

```
        else:
            print(node.value, end = ' ')
        self.DFSTree(node.left)
        self.DFSTree(node.right)

    def PrintDFSTree(self):
        self.DFSTree(self.root)

def main():
    root = Tree(7)
    root.insert(2)
    root.insert(25)
    root.insert(9)
    root.insert(80)
    root.insert(0)
    root.insert(5)
    root.insert(15)
    root.insert(8)

    root.PrintDFSTree()
main()
```

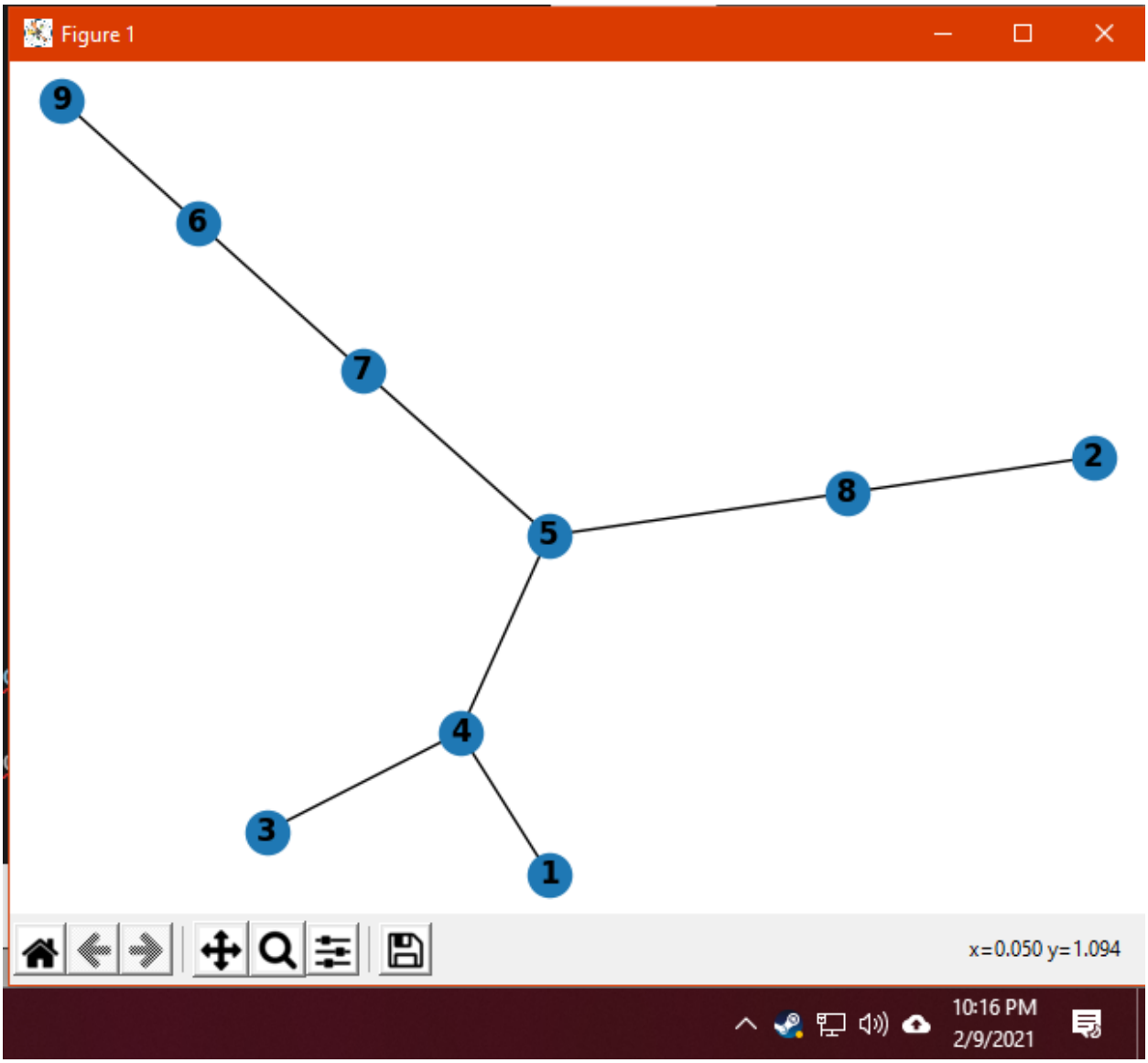
```
PS C:\Users\fagge\Documents\CS591> python -u "c:\Users\fagge\Documents\CS591\tempCodeRunnerFile.py"
7 2 0 5 25 9 8 15 80
PS C:\Users\fagge\Documents\CS591>
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

DFS4:

```
#DFS networkx and matplot lib
import networkx as nx
import matplotlib.pyplot as plt

G = nx.Graph()
G.add_node(1)
G.add_node(2)
G.add_node(3)
G.add_node(4)
G.add_node(5)
G.add_nodes_from([6, 7, 8, 9])

G.add_edge(5, 8)
G.add_edge(5, 4)
G.add_edge(5, 7)
G.add_edge(8, 2)
G.add_edge(4, 3)
G.add_edge(4, 1)
G.add_edge(7, 6)
G.add_edge(6, 9)

nx.draw(G, with_labels = True, font_weight = 'bold')

output = list(nx.dfs_preorder_nodes(G, source = 5))

plt.show()
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)



Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

DFS5:

```
#networkx digraph
import networkx as nx
import matplotlib.pyplot as plt

def dfs(dag, start, visited, stack):

       if start in visited:

           # node and all its branches have been visited
           return stack, visited


       if dag.out_degree(start) == 0:

           # if leaf node, push and backtrack
           stack.append(start)

           visited.append(start)

           return stack, visited

       #traverse all the branches
       for node in dag.neighbors(start):

           if node in visited:

               continue

           stack, visited = dfs(dag, node, visited, stack)

       #now, push the node if not already visited
       if start not in visited:
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

```
            print("pushing %s"%start)

            stack.append(start)

            visited.append(start)

        return stack, visited

def topological_sort_using_dfs(dag):

        visited = []

        stack=[]

        start_nodes = [i for i in dag.nodes if dag.in_degree(i)==0]

    #      print(start_nodes)

        for s in start_nodes:

            stack, visited = dfs(dag, s, visited, stack)

        print("Topological sorted:")

        while(len(stack)!=0):

            print(stack.pop(), end=" ")

dag = nx.digraph.DiGraph()
dag.add_nodes_from(['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'])
dag.add_edges_from([('A', 'B'), ('A', 'E'), ('B', 'D'), ('E', 'C'),
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

```
                    ('D', 'G'), ('C', 'G'), ('C', 'I'), ('F', '
I')])
topological_sort_using_dfs(dag)
topological_sorting = nx.topological_sort(dag)
print()
for n in topological_sorting:
    print(n, end=' ')
```

```
PS C:\Users\fagge\Documents\CS591> python -u "c:\Users\fagge\Documents\CS591\DFS5.py"
pushing D
pushing B
pushing C
pushing E
pushing A
pushing F
Topological sorted:
H F A E C I B D G
H F A E C I B D G
PS C:\Users\fagge\Documents\CS591>
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

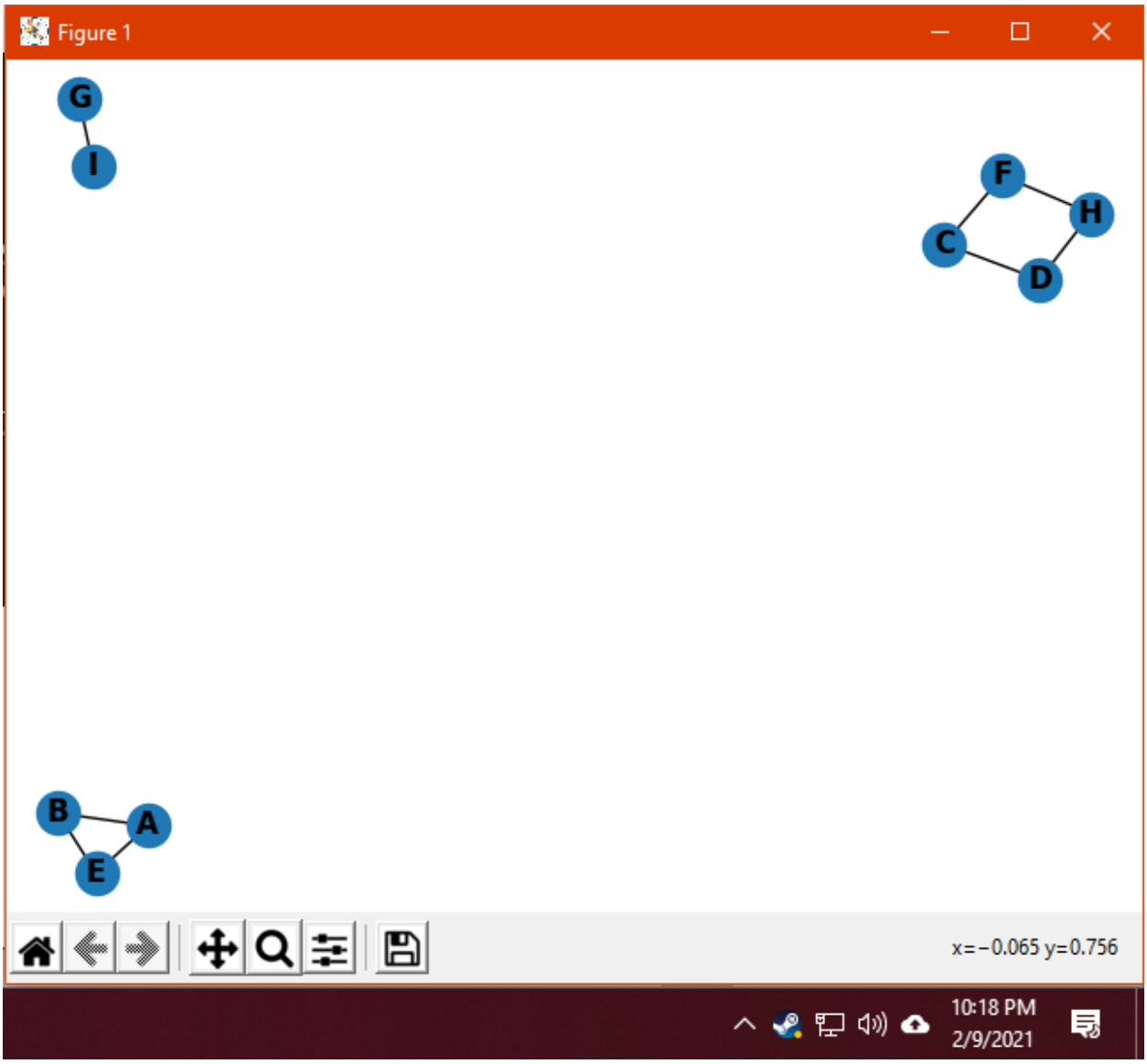
DFS6:

```
#Connected components DFS
import networkx as nx
import matplotlib.pyplot as plt

def dfs_traversal(graph, start, visited, path):
        if start in visited:
            return visited, path
        visited.append(start)
        path.append(start)
        for node in graph.neighbors(start):
            visited, path = dfs_traversal(graph, node, visited,
 path)
        return visited, path

def find_connected_components(graph):
    visited = []
    connected_components = []
    for node in graph.nodes:
        if node not in visited:
            cc = []
            visited, cc = dfs_traversal(graph, node, visited, c
c)
            connected_components.append(cc)
    return connected_components

def main():
    graph = nx.Graph()
    graph.add_nodes_from(['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H
', 'I'])
    graph.add_edges_from([('A', 'B'), ('B', 'E'), ('A', 'E')])
    graph.add_edges_from([('C', 'D'), ('D', 'H'), ('H', 'F'), (
'F', 'C')])
    graph.add_edge('G', 'I')
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

```
    nx.draw(graph, with_labels =  True, font_weight='bold')
    plt.show()

    connected_components = find_connected_components(graph)
    print(f'Total number of connected components = {len(connect
ed_components)}')

    for cc in connected_components:
        print(cc)

main()
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)




```
PS C:\Users\fagge\Documents\CS591> python -u "c:\Users\fagge\Documents\CS591\DFS6.py"
Total number of connected components = 3
['A', 'B', 'E']
['C', 'D', 'H', 'F']
['G', 'I']
PS C:\Users\fagge\Documents\CS591>
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

Breadth First Search:

Iterative BFS:

```
#Iterative BFS
from collections import deque

class Graph:
    def __init__(self, edges, N):

        #A list of lists that represents an adjacency list
        self.adjList = [[] for _ in range(N)]

        #This adds edges to the undirected graph
        for (src, dest) in edges:
            self.adjList[src].append(dest)
            self.adjList[dest].append(src)

#Perform BFS on the graph with starting point 'v'
def BFS(graph, v, discovered):

    q = deque() #queue for BFS
    discovered[v] = True #source vertex marked discovered

    #enqueue source vertex
    q.append(v)

    #loop till queue is empty
    while q:
        #dequeue front node and print it
        v = q.popleft()
        print(v, end=' ')

        #do for every edge `v -> `u
        for u in graph.adjList[v]:
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

```
            if not discovered[u]:
                #mark it as discovered and enqueue it
                discovered[u] = True
                q.append(u)

if __name__ == '__main__':
    edges = [
        (1, 2), (1, 3), (1, 4), (2, 5), (2, 6), (5, 9),
        (5, 10), (4, 7), (4, 8), (7, 11), (7, 12)
        #vertexes 0, 13, and 14 are single nodes
    ]
    N = 15

    graph = Graph(edges, N)

    discovered = [False] * N

    for i in range(N):
        if not discovered[i]:
            BFS(graph, i, discovered)
```

```
PS C:\Users\fagge\Documents\CS591> python -u "c:\Users\fagge\Documents\CS591\Assignment2\tempCodeRunnerFile.py"
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
PS C:\Users\fagge\Documents\CS591>
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

Recursive BFS:

```
#Recursive BFS Implementation
from collections import deque

class Graph:
    def __init__(self, edges, N):
        self.adjList = [[] for _ in range(N)]

        #add edges to the undirected graph
        for (src, dest) in edges:
            self.adjList[src].append(dest)
            self.adjList[dest].append(src)

#Perform BFS recursively on the graph
def recursiveBFS(graph, q, discovered):
    if not q: return

    #dequeue front node and print it
    v = q.popleft()

    print(v, end=' ')

    #do for every edge `v -> `u
    for u in graph.adjList[v]:
        if not discovered[u]:
            #mark it as discovered and enqueue it
            discovered[u] = True
            q.append(u)
    recursiveBFS(graph, q, discovered)

if __name__ == '__main__':
    #List of Graph edges:
    edges = [
        (1, 2), (1, 3), (1, 4), (2, 5), (2, 6), (5, 9),
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)

```
        (5, 10), (4, 7), (4, 8), (7, 11), (7, 12)
    ]

    N = 15
    #build a graph from edges
    graph = Graph(edges, N)

    discovered = [False] * N

    q = deque()

    for i in range(N):
        if not discovered[i]:
            discovered[i] = True

            q.append(i)

            recursiveBFS(graph, q, discovered)
```

```
PS C:\Users\fagge\Documents\CS591> python -u "c:\Users\fagge\Documents\CS591\Assignment2\BFSRecursive.py"
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
PS C:\Users\fagge\Documents\CS591>
```


Violations of academic honesty represent a serious breach of discipline and may be considered grounds for disciplinary action, including dismissal from the University. The University requires that all assignments submitted to faculty members by students be the work of the individual student submitting the work. An exception would be group projects assigned by the instructor. (Source: SEMO website)