Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

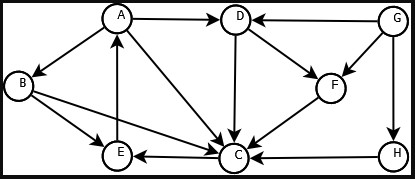
5

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | Apply DFS search on the graph given below. |
| 2 | Implement DFS search algorithm while using recursion. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Submitted On:

**Task no 1:** **Apply DFS search on the graph given below.**



**CODE:**

graph1 = {

  'A' : ['B','D'],

  'B' : ['C', 'E'],

  'C' : ['E'],

  'D' : ['C','F'],

  'E' : ['A'],

  'F' : ['C'],

  'G' : ['D','F','H'],

  'H' : ['C']

  }

g1=nx.Graph(graph1)

dfs=nx.dfs\_predecessors(g1,source='A')

dfs=dict(dfs)

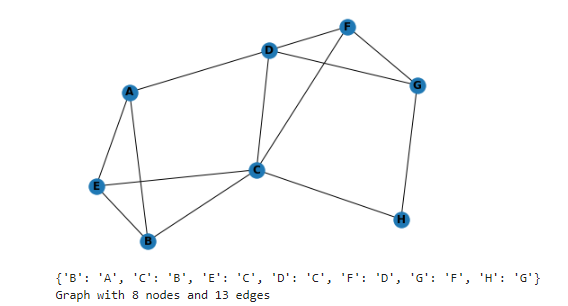
nx.draw(g1,font\_weight='bold',with\_labels=True)

plt.show()

print(dfs)

print(g1)

**OUTPUT:­**



**Task no 2:** **Implement DFS search algorithm while using recursion.**

**CODE:**

graph5 = {

  'A' : ['B','D'],

  'B' : ['C', 'E'],

  'C' : ['E'],

  'D' : ['C','F'],

  'E' : ['A'],

  'F' : ['C'],

  'G' : ['D','F','H'],

  'H' : ['C']

}

visited = set()

def dfs(visited, graph5, node):

    if node not in visited:

        print (node)

        visited.add(node)

        for neighbour in graph5[node]:

            dfs(visited, graph5, neighbour)

dfs(visited, graph5, 'A')

**OUTPUT:­**

Graphical user interface

Description automatically generated with low confidence