

Bahria University, Islamabad

Department of Software Engineering

Artificial Intelligence Lab

(Fall-2021)

Teacher: Engr. M Waleed Khan

Student : M Iqrar Ijaz Malik

Enrollment: 01-131182-021

Lab Journal: 6 Date: 10-11-2021

Task No:	Task Wis	se Marks	Docume Ma	Total Marks	
	Assigned	Obtained	Assigned	Obtained	(20)
1	15		5		

Comments:			

Signature

Lab No: 6 IMPLEMENTING GRAPH ALGORITHS

Introduction

- Graphs are very useful data structures in solving many important mathematical challenges.
- For example computer network topology or analysing molecular structures of chemical compounds. They are also used in city traffic or route planning and even in human languages and their grammar

OBJECTIVE:

- Introduce the basics of python.
- Code in python.
- Implement Graph Algorithms in Python

Tools Used

Tool used to perform this task is PyCharm Community Addition

Task 1: Implement DFS and BFS

Code For BSF

```
class Graph:
            def __init__(self, name=""):
 2
                self.name = name
 3
                self.neighborList = {}
 5
                self.nodeList = {}
 6
 7
            def add_node(self, node):
                self.nodeList[node] = True
 8
 9
            def add_edge(self, node, nodebis):
                try:
                    self.neighborList[node].append(nodebis)
                except:
                    self.neighborList[node] = []
14
                    self.neighborList[node].append(nodebis)
17
                    self.neighborList[nodebis].append(node)
18
                except:
                    self.neighborList[nodebis] = []
19
                    self.neighborList[nodebis].append(node)
            def neighbors(self, node):
23
                try:
                    return self.neighborList[node]
24
25
                except:
                    return []
                    return []
            def nodes(self):
28
                return self.nodeList.keys()
            def delete_edge(self, node, nodebis):
31
                self.neighborList[node].remove(nodebis)
32
                self.neighborList[nodebis].remove(node)
34
            def delete_node(self, node):
35
                del self.nodeList[node]
                try:
37
                    for nodebis in self.neighborList[node]:
                        self.neighborList[nodebis].remove(node)
39
                    del self.neighborList[node]
40
                except:
41
                    return "error"
42
43
```

```
def BFS(graph, start, end):
46
           step_count = 0
47
           explored = set()
48
           frontiers = [start]
           new_path = []
49
           if start == end:
               print("Goal is at start :" + str(start))
51
           while frontiers:
52
53
               step_count += 1
54
               node = frontiers.pop(0)
               explored.add(node)
57
               new_path.append(node)
               if node == end:
                   print("Breath First Search\n")
60
                   print("Sequence Is : ", new_path)
                   print("The Number of steps are : ", step_count)
                   return
64
               neighbours = graph.neighbors(node)
67
               for neighbour in neighbours:
                   if neighbour not in frontiers and neighbour not in explored:
68
```

```
69
                       frontiers.append(neighbour)
70
           return "No such Node Exist "
      dif __name__ == "__main__":
73
           G = Graph("BFS")
74
75
           # Creating Graph
76
           G.add_node(1)
77
78
           G.add_node(2)
79
           G.add_node(3)
80
           G.add_node(4)
81
           G.add_node(5)
82
           G.add_node(6)
           G.add_node(7)
83
           G.add_node(8)
84
85
           G.add_node(9)
           G.add_node(10)
86
           G.add_node(11)
87
           G.add_node(12)
88
89
           G.add_edge(1, 2)
90
           G.add_edge(1, 7)
           G.add_edge(1, 8)
91
           G.add_edge(2, 3)
92
```

```
M Iqrar Ijaz Malik Artificial Intelligence Engr. M Waleed Khan Enrollment Number Lab # 06 Dept of SE, BUIC
```

Screenshot

```
C:\Users\01-131182-021\AppData\Local\Microsoft\WindowsApps\python3.9.exe "F:/Study/Lectures Semister 7/AI/AI Assignment Breath First Search

Sequence Is: [1, 2, 7, 8, 3, 6, 9, 12, 4, 5]
The Number of steps are: 10

Process finished with exit code 0
```

Code For DFS

```
class Graph:
1
2
           def __init__(self, name=""):
3
               self.name = name
               self.neighborList = {}
6
               self.nodeList = {}
5
6
           def add_node(self, node):
7
               self.nodeList[node] = True
8
9
           def add_edge(self, node, nodebis):
               try:
11
                    self.neighborList[node].append(nodebis)
12
13
               except:
                    self.neighborList[node] = []
14
                    self.neighborList[node].append(nodebis)
               try:
                    self.neighborList[nodebis].append(node)
17
               except:
                    self.neighborList[nodebis] = []
                    self.neighborList[nodebis].append(node)
           def neighbors(self, node):
               try:
                    return self.neighborList[node]
24
25
               except:
                   return []
            def nodes(self):
                return self.nodeList.keys()
            def delete_edge(self, node, nodebis):
                self.neighborList[node].remove(nodebis)
32
                self.neighborList[nodebis].remove(node)
33
34
            def delete_node(self, node):
                del self.nodeList[node]
                try:
                    for nodebis in self.neighborList[node]:
38
                         self.neighborList[nodebis].remove(node)
                    del self.neighborList[node]
41
                except:
42
                    return "error"
43
```

```
def DFS(graph, start, end):
            step_count = 0
47
            explored = set()
            frontiers = [start]
            new_path = []
            if start == end:
                return "The starting node is the Goal :" + start
            while frontiers:
                step_count += 1
                node = frontiers.pop()
                explored.add(node)
                new_path.append(node)
                if node == end:
                    print("Depth First Search\n")
                    print("Sequence Is : ", new_path)
                    print("The Number of steps are : ", step_count)
                    return
                neighbours = graph.neighbors(node)
                for neighbour in neighbours:
                    if neighbour not in frontiers and neighbour not in explored:
                    if neighbour not in frontiers and neighbour not in explored:
68
                        frontiers.append(neighbour)
            return "No such Node Exist"
70
71
72
73

if __name__ == "__main__":
            G = Graph("DFS")
74
            # Creating Graph
            G.add_node(1)
            G.add_node(2)
78
79
            G.add_node(3)
            G.add_node(4)
80
            G.add_node(5)
81
            G.add_node(6)
82
83
            G.add_node(7)
            G.add_node(8)
84
85
            G.add_node(9)
            G.add_node(10)
86
            G.add_node(11)
87
88
            G.add_node(12)
            G.add_edge(1, 2)
89
90
            G.add_edge(1, 7)
91
            G.add_edge(1, 8)
            G.add_edge(2, 3)
92
```

```
M Iqrar Ijaz Malik Artificial Intelligence Engr. M Waleed Khan Enrollment Number Lab # 06 Dept of SE, BUIC
```

```
G.add_edge(1, 8)
91
             G.add_edge(2, 3)
92
             G.add_edge(2, 6)
93
             G.add_edge(3, 4)
94
             G.add_edge(3, 5)
95
             G.add_edge(8, 9)
96
             G.add_edge(8, 12)
97
             G.add_edge(9, 10)
98
             G.add_edge(9, 11)
99
100
             # Finding shortest Path
101
             DFS(G, 1, 9)
102
103
```

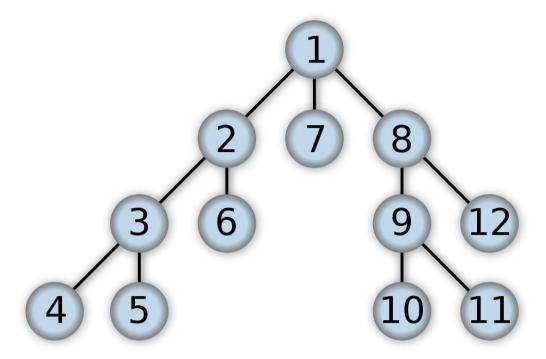
Output

```
C:\Users\01-131182-021\AppData\Local\Microsoft\WindowsApps\python3.9.exe "F:/Study/Lectures Semister 7/AI/AI Assignment
Depth First Search

Sequence Is: [1, 8, 12, 9]
The Number of steps are: 4

Process finished with exit code 0
```

• Graph Used



Conclusion

I completed the tasks given to us and pasted the output above.