

Assignment – 8

UI18EC39 (Abhishek Kumar)

Write a program to implement DFS and BFS

DFS

```
from collections import defaultdict

class Graph:

    def __init__(self):
        self.graph=defaultdict(list)

    def addEdge(self,u,v):
        self.graph[u].append(v)

    def DFSUtil(self,v,visited):
        visited.add(v)
        print(v)
        for neighbour in self.graph[v]:
            if neighbour not in visited:
                self.DFSUtil(neighbour,visited)

    def DFS(self):
        visited=set()
        for vertex in self.graph:
            if vertex not in visited:
                self.DFSUtil(vertex,visited)

g=Graph()
g.addEdge(0,1)
g.addEdge(0,2)
g.addEdge(1,2)
g.addEdge(2,0)
g.addEdge(2,3)
g.addEdge(3,3)
g.DFS()
```

BFS

```
from collections import defaultdict
```

```
class Graph:
```

```
    def _init_(self):  
        self.graph = defaultdict(list)
```

```
    def addEdge(self,u,v):  
        self.graph[u].append(v)
```

```
    def BFS(self, s):  
        visited = [False] * (max(self.graph) + 1)  
        queue = []  
        queue.append(s)  
        visited[s] = True
```

```
        while queue:  
            s = queue.pop(0)  
            print (s, end = " ")  
            for i in self.graph[s]:  
                if visited[i] == False:  
                    queue.append(i)  
                    visited[i] = True
```

```
g = Graph()  
g.addEdge(0, 1)  
g.addEdge(0, 2)  
g.addEdge(1, 2)  
g.addEdge(2, 0)  
g.addEdge(2, 3)  
g.addEdge(3, 3)
```

```
g.BFS(2)
```

-----**-----