

AI Lab Exam

Shashank
Kumar

~~Algo.~~

~~Algo :~~

1BM18CS098

if source == target
return True

def dfs(source, limit, layer=0, target)
if layer == limit
return
visited [source] = True
for node in graph [source]
dfs(node, limit, layer+1)
return False

def idfs (source, limit, target)
for i in range (1, 100)
dfs (source, i, 0, target)
add nodes to a list as u traverse
if target is found
print it
or delete content of list
if visited [source] == True
return

Prog:-

from collections import defaultdict

class Graph:

def __init__(self, vertices):

self.V = vertices

self.graph = defaultdict(list)

def addEdge(self, u, v):

self.graph[u].append(v)

def DFS(self, src, target, maxDepth):

if src == target: return True

if maxDepth <= 0: return False

for i in self.graph[src]:

(self.DFS(i, target, maxDepth - 1)):

return True

return False

def IDDFS(self, src, target, maxDepth):

for i in range(maxDepth):

(self.DLS(src, target, i)).

return True

return False

~~✓~~

```
g = Graph();  
g.addEdge(0, 1)  
g.addEdge(0, 2)  
g.addEdge(1, 3)  
g.addEdge(1, 4)  
g.addEdge(2, 5)  
g.addEdge(2, 6)
```

target = 6;

maxDepth = 3;

src = 0

if g.IDDFS(src, target, maxDepth)

== True:

print("Target is reachable from
source") + " within maxdepth")

else print("Target is NOT

reachable from source") + " within
max depth")

print(g.nodeCount)

✓