

Practical no:01

DFS and BFS

Program:

```
from collections import deque
```

```
class graph:
```

```
    def __init__(self,edge,n):
```

```
        self.adjList={}
```

```
        for i in range(n):
```

```
            self.adjList[i]=[]
```

```
        for (src,dest) in edge:
```

```
            self.adjList[src].append(dest)
```

```
            self.adjList[dest].append(src)
```

```
def recBFS(graph,q,discovered):
```

```
    if not q:
```

```
        return
```

```
    v=q.popleft()
```

```
    print(v,end=" ")
```

```
    for u in graph.adjList[v] :
```

```
        if not discovered[u]:
```

```
            discovered[u]= True
```

```
            q.append(u)
```

```
    recBFS(graph,q,discovered)
```

```
def DFSrec(graph,v,discovered) :
```

```
    print(v,end=" ")
```

```
    for u in graph.adjList[v]:
```

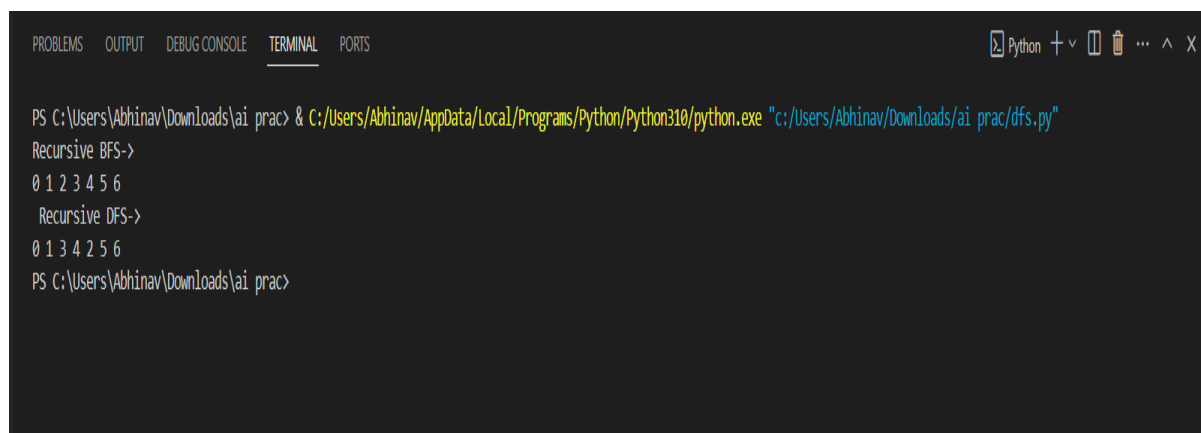
```
        if not discovered[u]:
```

```

        discovered[u]=True
        DFSrec(graph,u,discovered)
if __name__ == '__main__':
    edge=[(0,1),(0,2),(1,3),(1,4),(2,5),(2,6)]
    n=7
    graph=graph(edge,n)
    discovered=[False] * n
    q=deque()
    print ("Recursive BFS->")
    for i in range(n):
        if not discovered[i]:
            discovered[i]=True
            q.append(i)
            recBFS(graph,q,discovered)
    print ("\n Recursive DFS->")
    discovered=[False] * n
    discovered[0]=True
    DFSrec(graph,0,discovered)

```

Output:



The screenshot shows a terminal window with the following output:

```

PS C:\Users\Abhinav\Downloads\ai prac> & C:/Users/Abhinav/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/Abhinav/Downloads/ai prac/dfs.py"
Recursive BFS->
0 1 2 3 4 5 6
Recursive DFS->
0 1 3 4 2 5 6
PS C:\Users\Abhinav\Downloads\ai prac>

```

