Assignment 5

Adjacency List Graph Structure

Adjacency List

```
class Node_Vertex():
  def __init__(self,Name):
    self.incident=[]
    self.Vname=Name
class Node_degree():
  def __init__(self,Name,E):
    self.V=Name
    self.E=E


class Graph():
  def __init__(self):
    self.vertices=[]
    self.edge={}
    self.numofvert=0
    self.numofedge=0
    self.listofdegree={}
  def insert_vertex(self,V):
    A=Node_Vertex(V)
    self.vertices.append(A)
    self.numofvert+=1
    self.listofdegree[V]=0
```

```
def insert_Edge(self,E,*vertice):
  #N=Node_Edge(Name,*vertice)
  K=vertice
  A=Node_degree(K[0],E)
  B=Node_degree(K[1],E)
  self.edge[E]=vertice
  self.numofedge+=1

  self.listofdegree[A.V]+=1
  self.listofdegree[B.V]+=1
```

- สร้าง class ของ Node_Vertex และ Node_degree
- สร้างโครงสร้าง Graph โดยใน Graph จะประกอบไปด้วย methods
    1. คุณสมบัติพื้นฐาน ประกอบด้วย:

       vertices, edge, listofdegree, numofedge, numofvert
    2. insert_vertex(V) : เพิ่มจุด vertex
    3. insert_Edge(E,*vertice) : เพิ่ม Edge

Driver Code

```
A=Graph()
A.insert_vertex("SFO")
A.insert_vertex("ORD")
A.insert_vertex("LAX")
A.insert_vertex("DFW")
A.insert_Edge("1843","SFO","ORD")
A.insert_Edge("337","SFO","LAX")
A.insert_Edge("1743","ORD","LAX")
A.insert_Edge("802","ORD","DFW")
A.insert_Edge("1233","DFW","LAX")
print(f"set of degree {A.listofdegree}")
for i in range(A.numofvert):
  print(f"vertice{i+1} : ({A.vertices[i].Vname})",end=" ")
print(f"\n set of edges: {A.edge}")
```

```
set of degree {'SFO': 2, 'ORD': 3, 'LAX': 3, 'DFW': 2}
vertice1 : (SFO) vertice2 : (ORD) vertice3 : (LAX) vertice4 : (DFW)
 set of edges: {'1843': ('SFO', 'ORD'), '337': ('SFO', 'LAX'), '1743': ('ORD', 'LAX'), '802': ('ORD', 'DFW'), '1233': ('DFW', 'LAX')}
```

Edge List Graph Structure

```
class Node_Vertex():
  def __init__(self,Name):
    self.incident=[]
    self.Vname=Name
class Graph():
  def __init__(self):
    self.vertices=[]
    self.edge={}
    self.numofvert=0
    self.numofedge=0
  def insert_vertex(self,V):
    A=Node_Vertex(V)
    self.vertices.append(A)
    self.numofvert+=1
  def insert_Edge(self,Name,*vertice):
    #N=Node_Edge(Name,*vertice)
    self.edge[Name]=vertice
    self.numofedge+=1
```

```
def degree(self,Vert):
  deg=0
  for x in self.edge:

    if self.edge[x][0]==Vert:

      deg+=1
    if self.edge[x][1]==Vert:
      deg+=1
  return deg
```

- สร้าง class ของ Node_Vertex
- สร้างโครงสร้าง Graph โดยใน Graph จะประกอบไปด้วย methods
    1. คุณสมบัติพื้นฐาน ประกอบด้วย:

       vertices, edge, listofdegree, numofedge, numofvert
    2. insert_vertex(V) : เพิ่มจุด vertex
    3. insert_Edge(E,*vertice) : เพิ่ม Edge
    4. degree(Vert): Return จำนวน degree ของ vertex นั้น

Driver Code

```
A=Graph()
A.insert_vertex("SFO")
A.insert_vertex("ORD")
A.insert_vertex("LAX")
A.insert_vertex("DFW")
A.insert_Edge("1843","SFO","ORD")
A.insert_Edge("337","SFO","LAX")
A.insert_Edge("1743","ORD","LAX")
A.insert_Edge("802","ORD","DFW")
A.insert_Edge("1233","DFW","LAX")
for i in range(A.numofvert):
  print(f"vertex{i+1}:{A.vertices[i].Vname}",end=" ")
print(f"\n set of edge{A.edge}")
print("\n Number of Degree of each vertex")
print(A.degree("SFO"))
print(A.degree("ORD"))
print(A.degree("LAX"))
print(A.degree("DFW"))
```

```
vertex1:SFO vertex2:ORD vertex3:LAX vertex4:DFW
 set of edge{'1843': ('SFO', 'ORD'), '337': ('SFO', 'LAX'), '1743': ('ORD', 'LAX'), '802': ('ORD', 'DFW'), '1233': ('DFW', 'LAX')}

 Number of Degree of each vertex
2
3
3
2
```