

LAB # 04

Task# 01: Graph Implementation

Code:

```
In [3]: import random

class Node:
    def __init__(self, data):
        self.data = data
        self.next = None

class Graph:
    def __init__(self, ver):
        self.vertex = ver
        self.graph = [None] * self.vertex

    def add_edge(self, ver1, ver2):
        node = Node(ver2)
        node.next = self.graph[ver1]
        self.graph[ver1] = node

    def display(self):
        for i in range(self.vertex):
            print("Vertex " + str(i) + ":", end="")
            temp = self.graph[i]
            while temp:
                print(" -> {}".format(temp.data), end="")
                temp = temp.next
            print(" \n")

if __name__ == "__main__":
    V = 4
    graph = Graph(V)
    graph.add_edge(0, 1)
    graph.add_edge(0, 2)
    graph.add_edge(0, 3)
    graph.add_edge(1, 5)

    graph.display()
```

Vertex 0: -> 3 -> 2 -> 1

Vertex 1: -> 5

Vertex 2:

Vertex 3:

CODE OF DYNAMIC GRAPH:

```
import networkx as nx
import matplotlib.pyplot as plt
import seaborn as sns
#%%matplotlib notebook
sns.set()

G = nx.Graph()
po = nx.spring_layout(G)
while True:
    print("Enter 1 for input edges and weight: ")
    print("Enter 2 for input vertices: ")
    a = int(input())
    if(a==1):
        edg1 = int(input("Enter edge: "))
        edg2 = int(input("Enter edge: "))
        wei = int(input("Enter weight: "))
        G.add_edge(edg1,edg2,weight=wei)
    elif(a==2):
        nod = int(input("Enter node: "))
        px = int(input("Enter position at x: "))
        py = int(input("Enter position at y: "))
        G.add_node(nod,pos=(px,py))
    else:
        break
weight = nx.get_edge_attributes(G,'weight')
pos = nx.get_node_attributes(G,'pos')

#plt.figure()
nx.draw_networkx(G,pos)
nx.draw_networkx_edge_labels(G,pos,edge_labels=weight)
plt.show()
```

Input & Output:

```
PS F:\UBIT ( BSCS )\3 Year\Semester V\AI\Labs> python -u "f:\UBIT ( BSCS )\3 Year\Semester V\AI\Labs\graph.py"
Enter 1 for input edges and weight:
Enter 2 for input vertices:
1
Enter edge: 1
Enter edge: 2
Enter weight: 3
Enter 1 for input edges and weight:
Enter 2 for input vertices:
2
Enter node: 2
Enter position at x: 1
Enter position at y: 1
Enter 1 for input edges and weight:
Enter 2 for input vertices:
2
Enter node: 1
Enter position at x: 0
Enter position at y: 0
Enter 1 for input edges and weight:
Enter 2 for input vertices:
1
Enter edge: 2
Enter edge: 3
Enter weight: 6
Enter 1 for input edges and weight:
Enter 2 for input vertices:
2
Enter node: 3
Enter position at x: 2
Enter position at y: 2
Enter 1 for input edges and weight:
Enter 2 for input vertices:
3
█
```

Figure 1

