



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

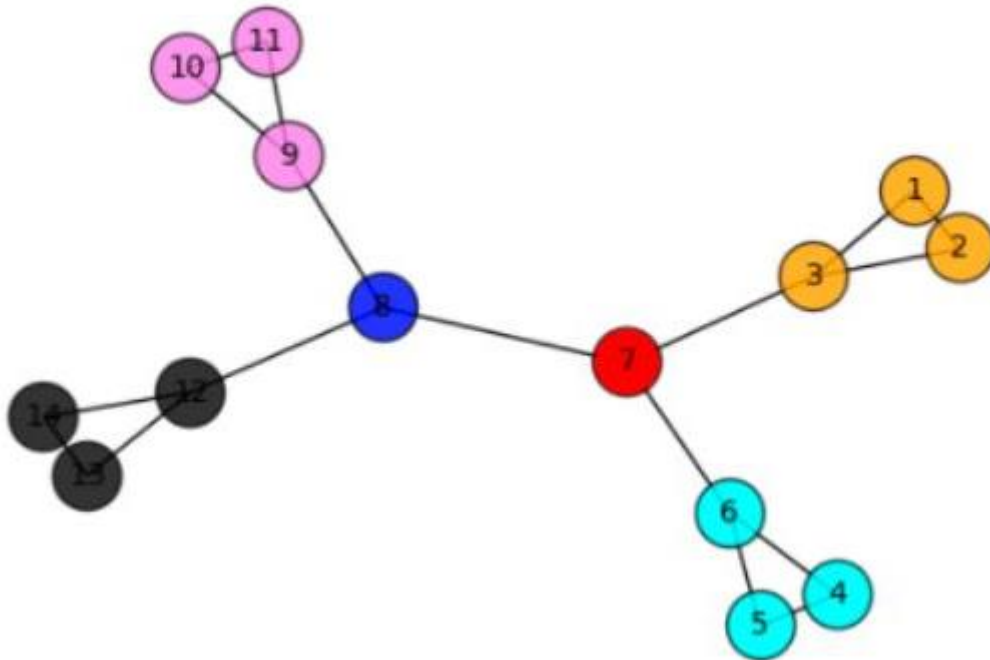
Programme	: B.Tech	Semester	: Win Sem 21-22
Course	: Web Mining Lab	Code	: CSE3024
Faculty	: Dr.Bhuvaneswari A	Slot	: L7+L8
Date	: 19-03-2022	Marks	: 10 Points

Name : Kartikey Gautam

Reg No – 19BCE1327

FILE NAME – 19BCE1327_WebMining_LAB10_PARTB.pdf

b. Facebook Friend Connection network dataset (refer folder B for custom implementation)



CODE:

```
import community.community_louvain as c
import networkx as nx
import matplotlib.pyplot as plt
import sys
import pylab
import copy

pylab.show()

def Remove_Edges(G):
    remove = []
    b = nx.edge_betweenness centrality(G)
    max_betweenness = b[max(b,key=b.get)]
    for k,v in b.items():
        if v==max_betweenness:
            remove.append(k)
            G.remove_edges_from(remove)
            graphs = list(nx.connected_components(G))

    d={}
    counter = 0
    for graph in graphs:
        counter+=1
    for node in graph:
        d[node]=counter

    if G.number_of_edges() == 0:
        return [list(nx.connected_components(G)),0,G]

    m = c.modularity(d,G)
    return [list(nx.connected_components(G)),m,G]

if __name__=="__main__":

    result_communities=[]
    G = nx.read_edgelist("input.txt")
    copyGraph = copy.deepcopy(G)
    d={}
    for node in G:
        d[node]=0
        initial_modularity = c.modularity(d, G)
        result_communities.append([d,initial_modularity,G])

    while G.number_of_edges()>0:
        subgraphs = Remove_Edges(G)
        result_communities.append(subgraphs)
        G=subgraphs[-1]
        for step in result_communities:
            if step[1]>initial_modularity:
                ng=step[0]
```

```

        result=[]
        modularity=step[1]
    for graph in step[0]:
        result.append(sorted([int(vertex) for vertex in graph]))

    for community in result:
        print(community)

d={ };
counter=0
for graph in ng:
    for node in graph:
        d[node] = counter
        counter+=1

pos=nx.spring_layout(copyGraph)
colors = ["violet", "blue", "green", "yellow", "turquoise"]
for i in range(len(ng)):
    graph=ng[i]
    nlist = [node for node in graph]

nx.draw_networkx_nodes(copyGraph,pos,nodelist=nlist,node_color=colors[i%10],node_size=500,alpha=0.8)

nx.draw_networkx_edges(copyGraph,pos)
nx.draw_networkx_labels(copyGraph,pos,font_size=10)
plt.axis('off')

```

OUTPUT:

```

[1, 2, 3]
[7]
[4, 5, 6]
[8]
[9, 10, 11]
[12, 13, 14]

```

