

## lab-6-22mcb1002

April 12, 2023

```
[31]: import networkx as nx
import matplotlib.pyplot as plt
from networkx.algorithms.community centrality import girvan_newman
from networkx.algorithms.community import greedy_modularity_communities
```

```
[2]: G = nx.karate_club_graph()
communities = girvan_newman(G)
```

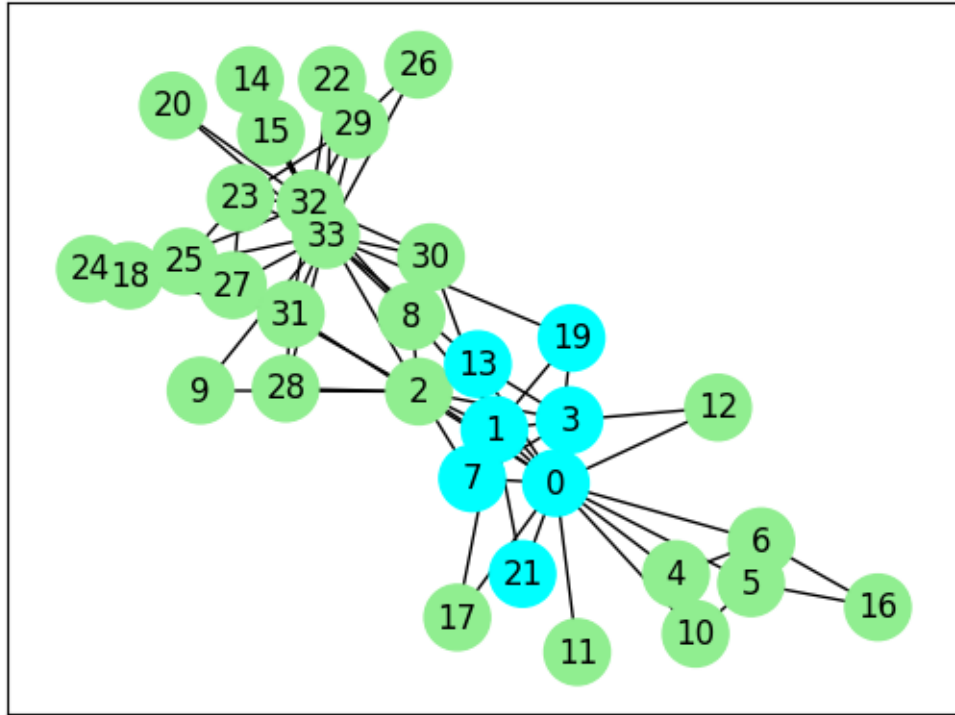
```
[15]: node_groups = []
for com in next(communities):
    node_groups.append(list(com))

print(node_groups)
```

```
[[0, 1, 3, 7, 13, 19, 21], [2, 27, 28], [4, 5, 6, 10, 16], [32, 33, 8, 22, 23,
29, 30], [9], [11], [12], [14], [15], [17], [18], [20], [24, 25, 31], [26]]
```

### A. Zachary karate club network dataset

```
[18]: color_map = []
for node in G:
    if node in node_groups[0]:
        color_map.append('#00FFFF')
    else:
        color_map.append('#90EE90')
graph = nx.draw_networkx(G, node_size = 600, node_color = color_map)
plt.show()
```



## B. Albert-László Barabási community algorithm

```
[20]: communities = list(greedy_modularity_communities(G))

for i, community in enumerate(communities):
    print(f"Community {i+1}: {community}")
```

Community 1: frozenset({8, 14, 15, 18, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33})

Community 2: frozenset({1, 2, 3, 7, 9, 12, 13, 17, 21})

Community 3: frozenset({0, 16, 19, 4, 5, 6, 10, 11})

## C. Internal and External Community Densities

```
[21]: for i, community in enumerate(communities):
        subgraph = G.subgraph(community)
        internal_density = nx.density(subgraph)
        external_density = nx.density(G) - internal_density
        print(f"Community {i+1}: Internal density={internal_density:.3f}, External_
        ↪density={external_density:.3f}")
```

Community 1: Internal density=0.250, External density=-0.111

Community 2: Internal density=0.361, External density=-0.222

Community 3: Internal density=0.429, External density=-0.290

## D. Community detection of facebook

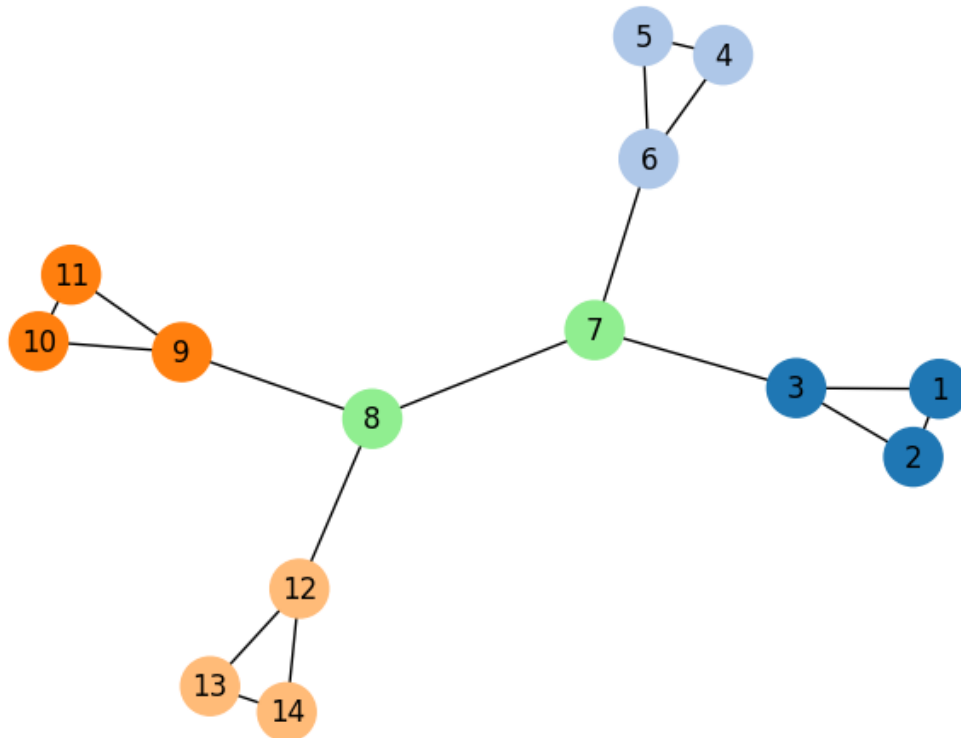
```
[32]: fb = nx.read_edgelist("input.txt", nodetype = int)

[43]: triangles = [clique for clique in nx.find_cliques(fb) if len(clique) == 3]

colors = {}
for i, tri in enumerate(triangles):
    color = plt.cm.tab20(i)
    for node in tri:
        colors[node] = color

node_colors = [colors[node] if node in colors else '#90EE90' for node in fb.
↪nodes()]
nx.draw(fb, with_labels = True, node_color = node_colors, node_size = 600)
plt.show()
```

```
/usr/local/lib/python3.9/dist-packages/matplotlib/cbook/__init__.py:1062:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences
(which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray.
  x = np.asanyarray(x)
```



```
[40]: communities = list(greedy_modularity_communities(fb))

for i, community in enumerate(communities):
    print(f"Community {i+1}: {community}")
```

```
Community 1: frozenset({1, 2, 3})
Community 2: frozenset({4, 5, 6})
Community 3: frozenset({9, 10, 11})
Community 4: frozenset({12, 13, 14})
Community 5: frozenset({8, 7})
```