

MINING GRAPH DATA

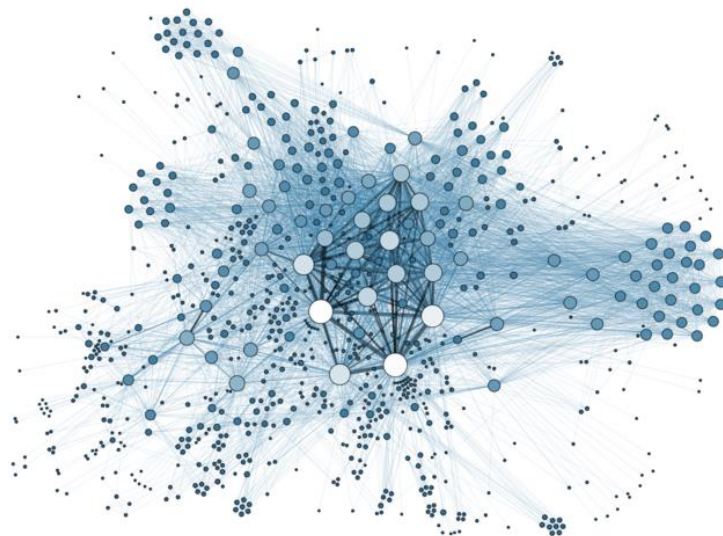
STQD6414 PERLOMBONGAN DATA



Assoc. Prof. Dr. Nurulkamal Masseran
Department of Mathematical Sciences
Universiti Kebangsaan Malaysia

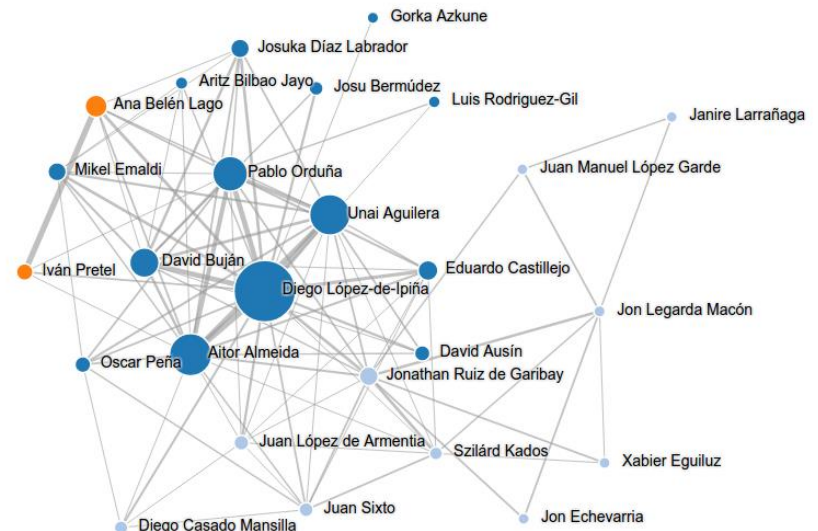
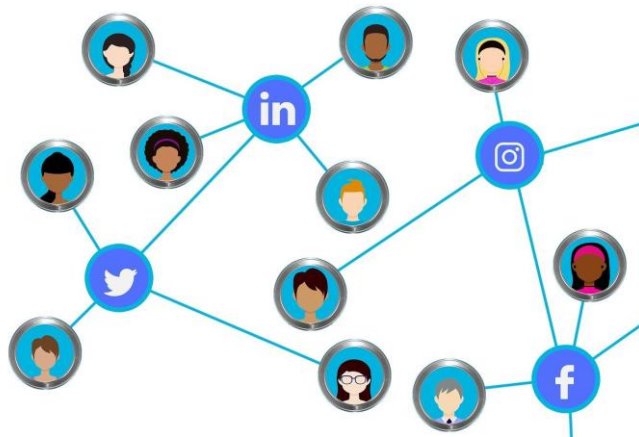
INTRODUCTION:

- A graph is a non-linear data structure that consist of nodes and edges.
- The objective of graph mining is to extract insightful knowledge from a data that is represented as a graph.
- Nowadays, graph-type data is everywhere, and available in many different fields.
- **Examples:** social network graphs, web graphs, cybersecurity networks, power grid networks, supply chain management, protein-protein interaction networks, and etc.



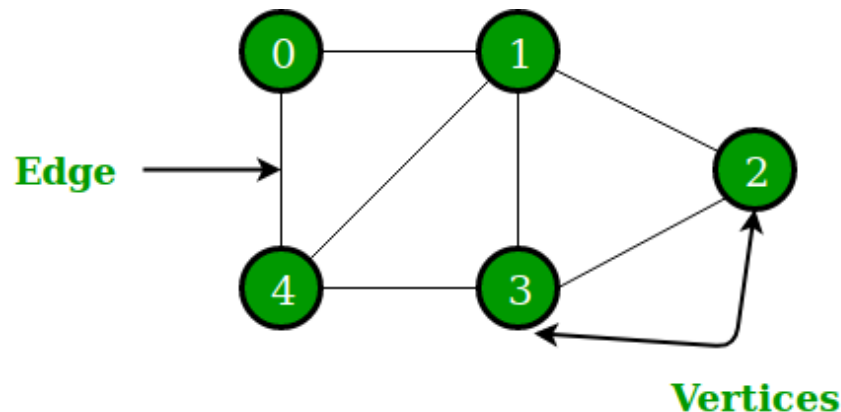
INTRODUCTION:

- Graphs often used to describe about links, relationships, or interconnections among some entities.
- **Example:** In the social science domain, the nodes in a graph are people and the links between them are friendship or professional collaboration, as can be seen from the platform of Facebook, LinkedIn, Instagram, twitter and etc.
- Through the analysis of graph data, we can gain a better understanding about the characteristics, behaviors or interaction trends among some particular entities.



INTRODUCTION:

- To analyze the structure of a graph data, the knowledge of graph theory is required.
- Graph theory is a branch of mathematics that concerned with networks of points connected by lines.
- It provides a mathematical foundation used to model pairwise relations between objects.
- In general, graph represent structured data that contain vertices/nodes and edges:
 - i) Graph vertices represent information related to some entities.
 - ii) The edges of the graph represent the relationship between the information and the entities.



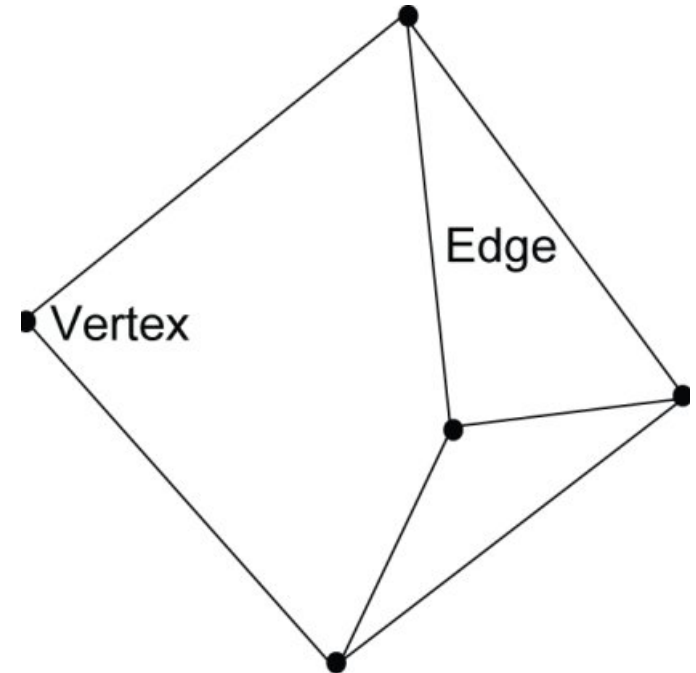
BASIC OF GRAPH THEORY:

- Some important definitions:

i) **Graph:** A graph G is composed of two sets: a set of vertices, denoted $V(G)$, and a set of edges, denoted $E(G)$.

ii) **Edge:** An edge in a graph G is an unordered pair of two vertices (v_1, v_2) such that $v_1 \in V(G)$ and $v_2 \in V(G)$.

iii) **Degree:** $\text{degree}(v)$, is the number of times vertex v occurs as an endpoint for the edges $E(G)$.

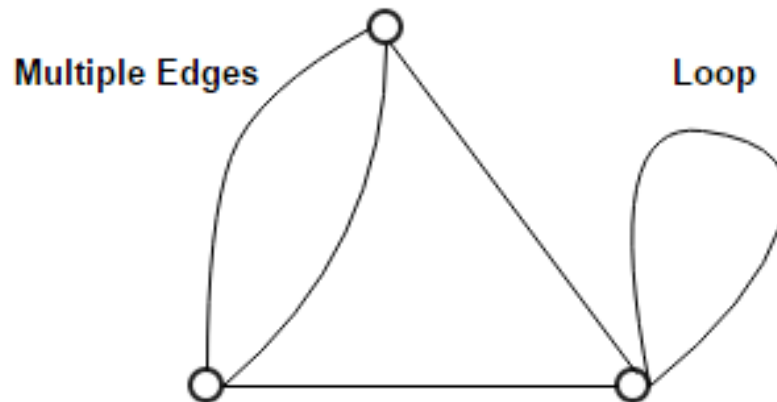


BASIC OF GRAPH THEORY:

iv) **Loop:** A loop is an edge that joins a vertex to itself.

v) **Multiple Edge:** An edge is a multiple edge if there is another edge in $E(G)$ which joins the same pair of vertices.

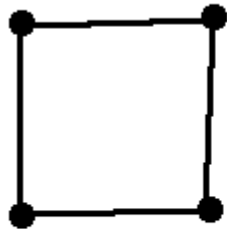
vi) **Simple Graph:** A graph with no loops or multiple edges.



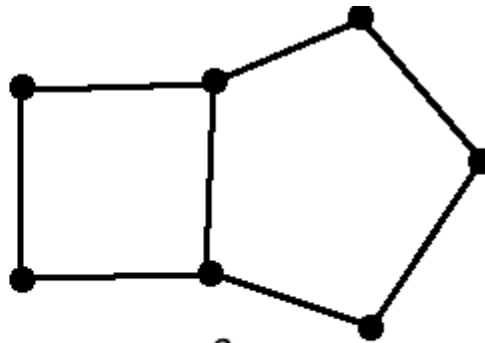
BASIC OF GRAPH THEORY:

vii) Subgraph:

- A subgraph S of a graph G is a graph whose vertex set $V(S)$ is a subset of the vertex set $V(G)$, that is $(V(S) \subseteq V(G))$.
- While, an edge set $E(S)$ is a subset of the edge set $E(G)$, that is $(E(S) \subseteq E(G))$.



S
(Subgraph of G)



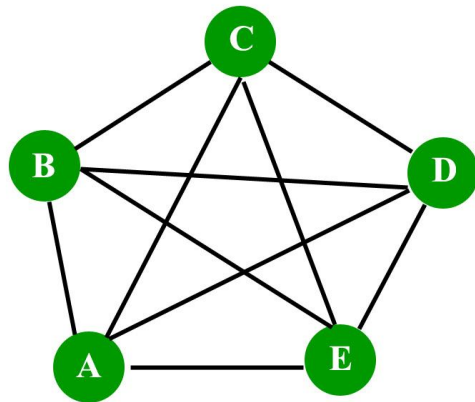
G



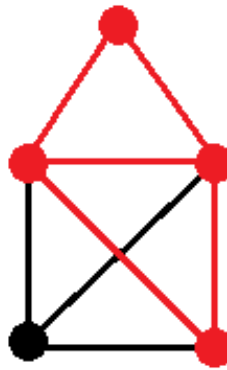
BASIC OF GRAPH THEORY:

viii) Clique:

- A subset $A \subseteq V$ is a complete graph if all vertex pairs in A are connected by an edge.
- A graph $G = (V, E)$ is complete if the vertex set V is complete.
- A clique is a maximal complete subset, if a complete subset is not contained in a larger complete subset.
- The set of cliques of a graph G is denoted by $C(G)$.



Complete Graph



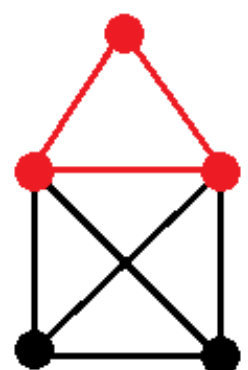
not a clique



non-maximal clique



maximal clique

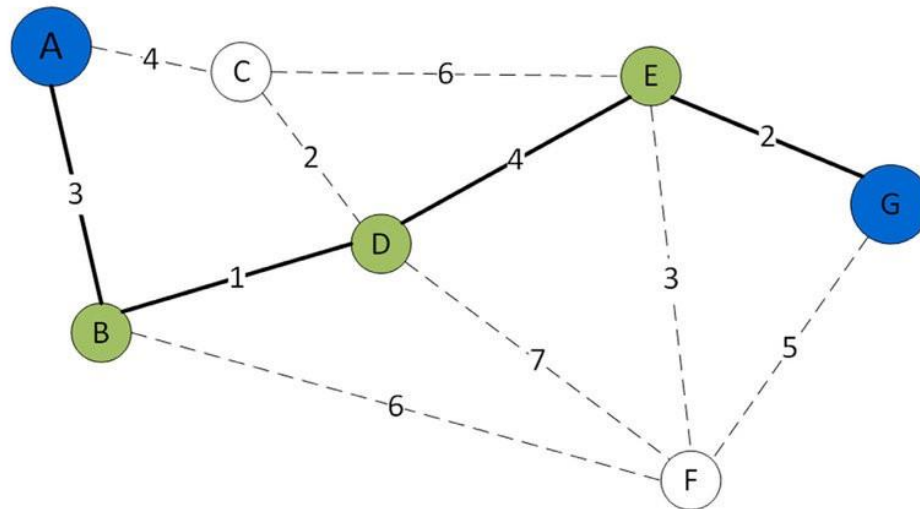


maximal clique

BASIC OF GRAPH THEORY:

ix) Path and circle:

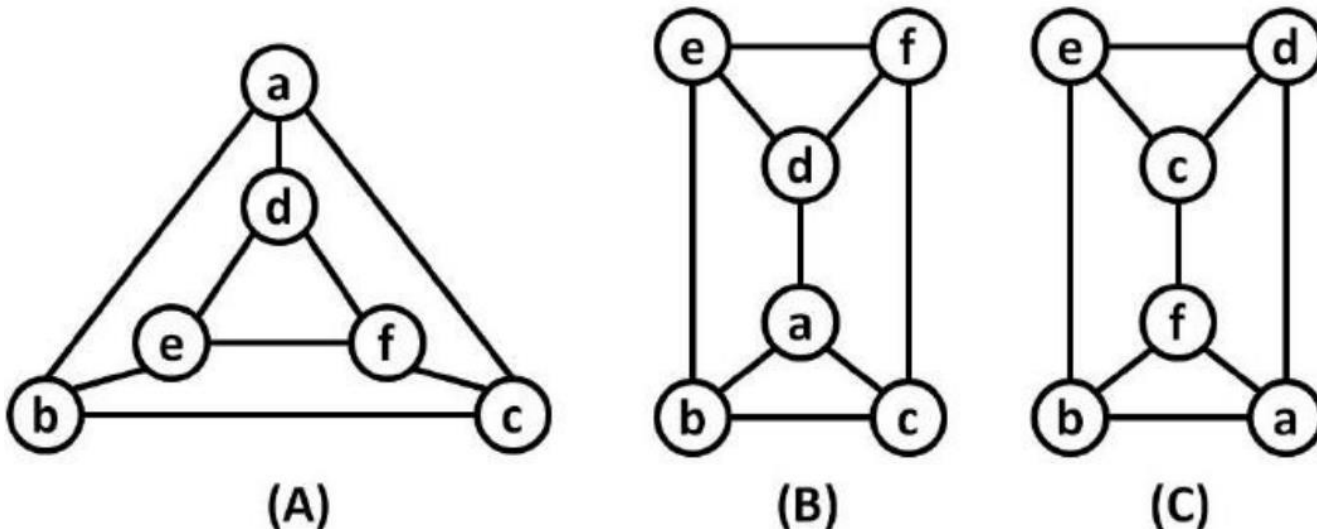
- A path (of length n) between α and β in an undirected graph is a set of vertices, such that $\alpha = \alpha_0, \alpha_1, \dots, \alpha_n = \beta$.
- If a path $\alpha = \alpha_0, \alpha_1, \dots, \alpha_n = \beta$ has $\alpha = \beta$ then the path is said to be a cycle of length n .



BASIC OF GRAPH THEORY:

x) **Isomorphic graphs:** A graph that can exist in different forms but having the same number of vertices, edges, and also having same edge connectivity.

xi) **Automorphic Graphs:** A graphs that have the same structure, but they have different relationship behavior. Thus, they are not exactly the same.

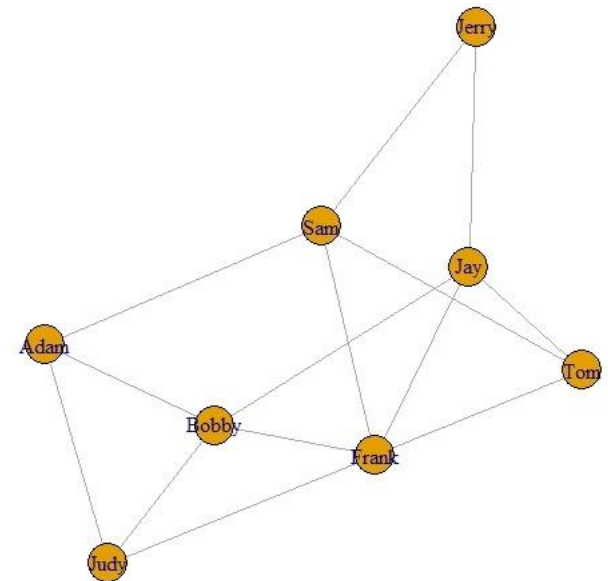
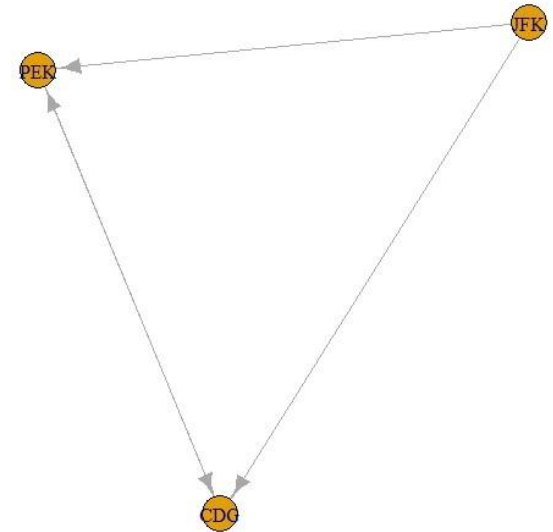


TYPES OF GRAPHS:

- There are various types of graph, among them are:

i) Undirected and Directed graphs:

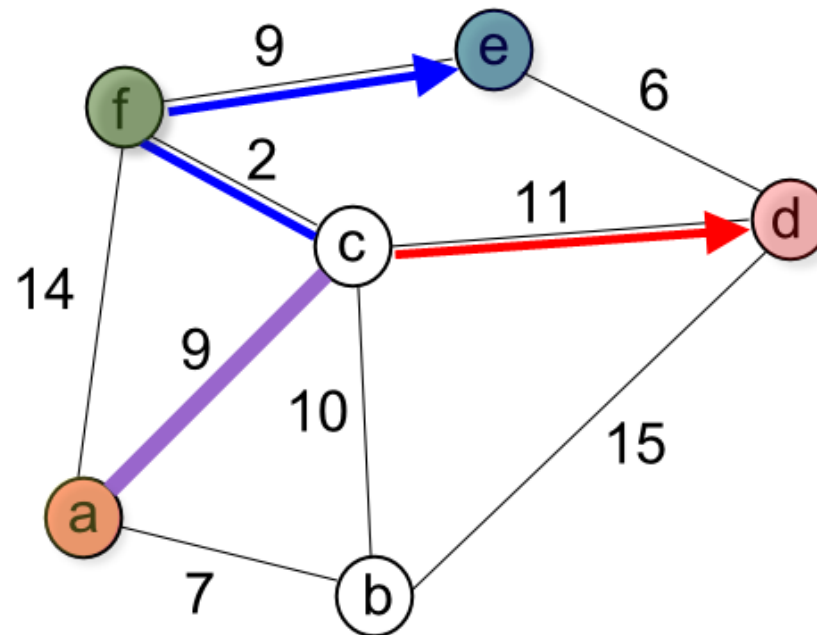
- A directed graph contains an ordered pair of vertices.
- Thus, directed graphs have edges with some specific directions.
- Undirected graph having unordered pair of vertices.
- Which implies that the edges for undirected graphs do not have a specific direction.



TYPES OF GRAPHS:

ii) Weighted and Unweighted graphs:

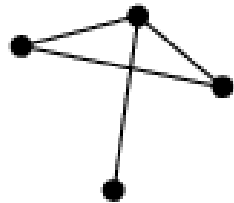
- A weight in graph represent a magnitude of relationships among the nodes and edges.
- A graphs that have weights is said to be a weighted graph, and vise versa.



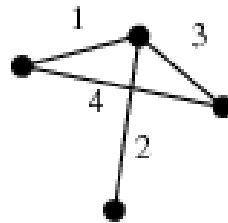
TYPES OF GRAPHS:

iii) Labeled and Unlabeled graphs:

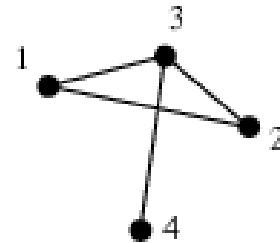
- An unlabeled graph is a graph whose nodes or edges do not have any indicator except through their relations.
- Whereas a labeled graph has several indicators in its nodes or edges.



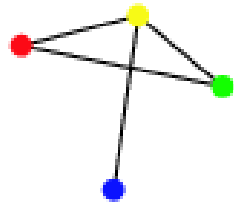
unlabeled graph



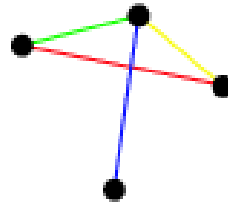
edge-labeled graph



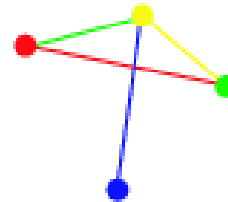
vertex-labeled graph



vertex-colored graph



edge-colored graph



vertex- and edge-colored graph

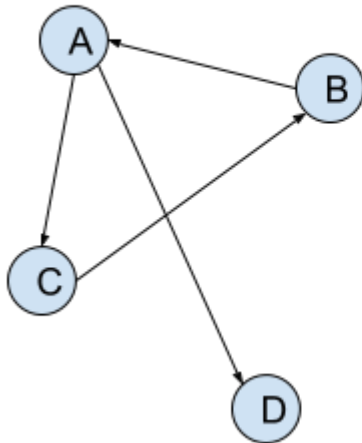


TYPES OF GRAPHS:

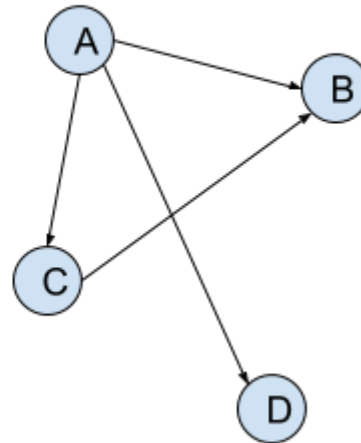
iv) Cyclic and acyclic graphs:

- A graph with at least one cycle is called a cyclic graph.
- A graph with no cycles is called an acyclic graph.

Cyclic Graph



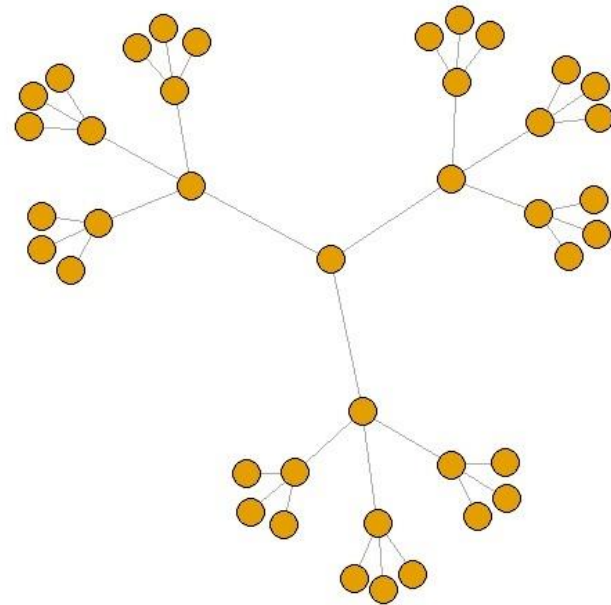
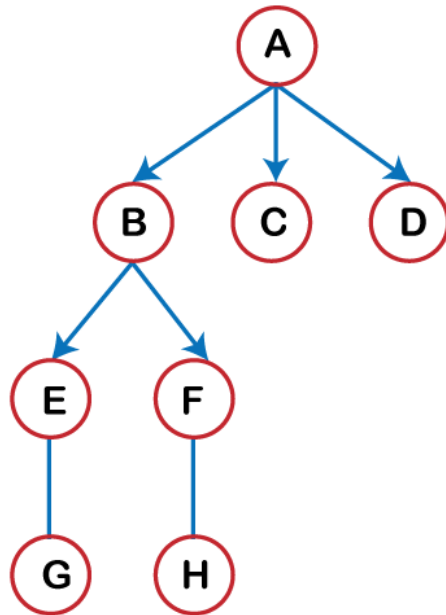
Acyclic Graph



TYPES OF GRAPHS:

v) Trees Graph:

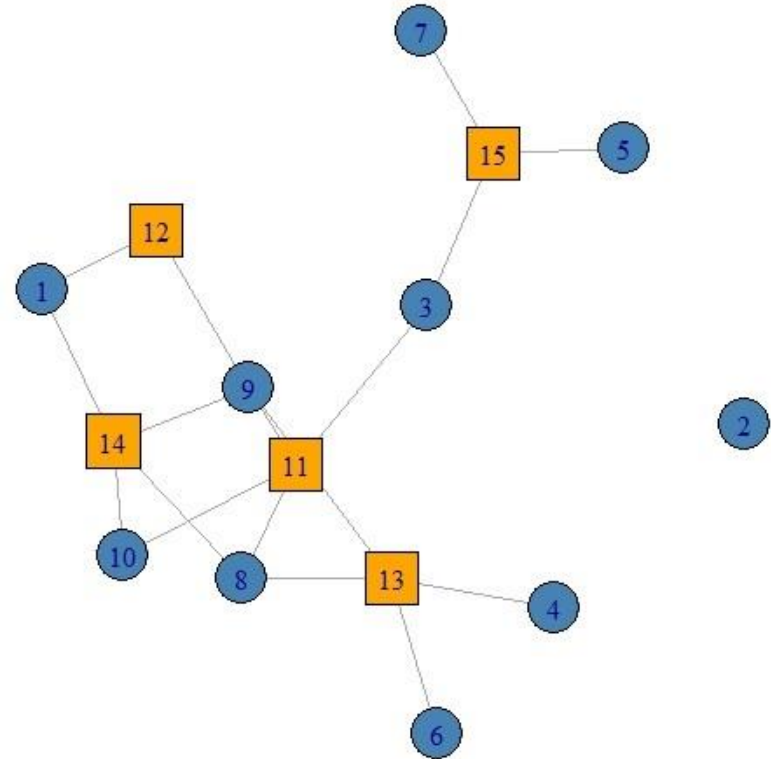
- This is undirected graph with any two vertices are connected by exactly one path.
- There is no cycles in this graph.
- Its also known as a connected acyclic undirected graph



TYPES OF GRAPHS:

vi) Bipartite graph:

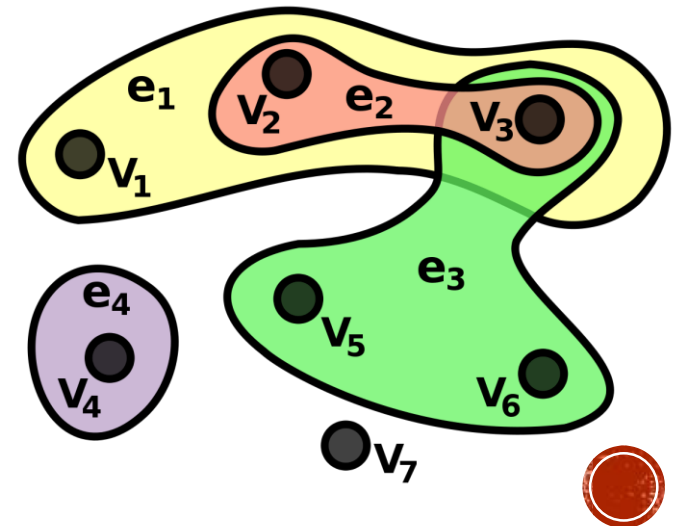
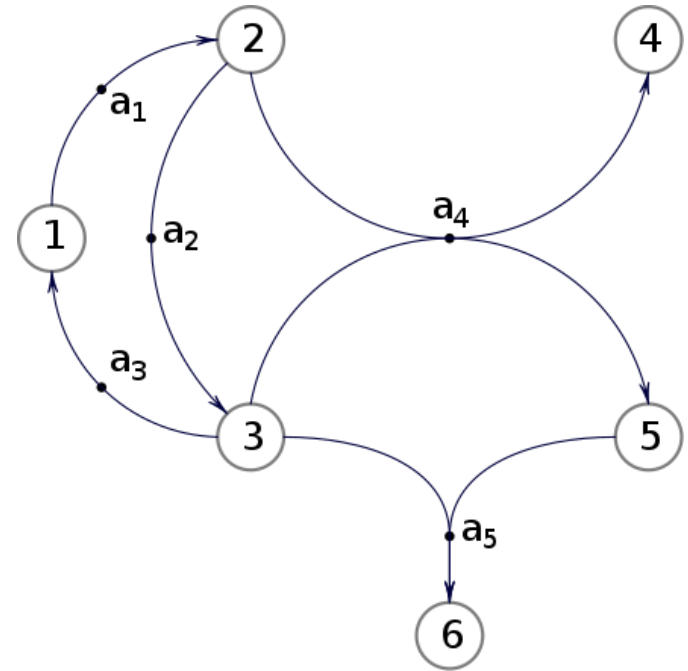
- A Bipartite Graph is a graph whose vertices can be divided into two independent sets (U and V)
- Every edge (u,v) either connects a vertex from U to V or a vertex from V to U.
- There is no edge that connects vertices of same set.
- The concept of bipartite graph can be generalized into multipartite graph.



TYPES OF GRAPHS:

vii) Hypergraph:

- A common hyper graph is a 2-hypergraph (one edge connects 2 nodes).
- A hypergraph is a generalization of a graph where an edge can join any number of nodes.
- Hyper-edges (generalized edges) can connect to a subset of nodes compared to a non-hyper graph that only connects to 2 nodes on one edge.
- A k-hypergraph has all such hyper-edges connecting exactly k nodes.



REPRESENTATIONS FOR GRAPHS:

- Generally, the graph will be stored in four basic formats:

i) Adjacency lists:

- An adjacency list is a collection of unordered lists.
- Each unordered list describes the set of neighbors of a specific vertex in the graph within an adjacency list.

ii) Edge lists:

- An edge list is a two-column table to list all the node pairs in the graph.

```
$Adam  
+ 3/8 vertices, named, from d339868:  
[1] Judy Bobby Sam
```

```
$Judy  
+ 3/8 vertices, named, from d339868:  
[1] Adam Bobby Frank
```

```
$Bobby  
+ 4/8 vertices, named, from d339868:  
[1] Adam Judy Frank Jay
```

```
$Sam  
+ 4/8 vertices, named, from d339868:  
[1] Adam Frank Tom Jerry
```

```
$Frank  
+ 5/8 vertices, named, from d339868:  
[1] Judy Bobby Sam Jay Tom
```

	V1	V2
1	Adam	Judy
2	Adam	Bobby
3	Adam	Sam
4	Judy	Bobby
5	Judy	Frank
6	Bobby	Frank
7	Bobby	Jay
8	Sam	Frank
9	Sam	Tom



REPRESENTATIONS FOR GRAPHS:

iii) Adjacency matrix:

- This matrix shows whether two vertices in the graph are connected or not.
- If there is a link between two nodes “i and j,” the row-column indices (i, j) will be marked as 1, otherwise 0.

```
8 x 8 sparse Matrix of class "dgCMatrix"
      Adam Judy Bobby Sam Frank Jay Tom Jerry
Adam    .    1     1    1     .    .    .    .
Judy    1     .     1    .     1    .    .    .
Bobby   1     1     .    .     1    1    .    .
Sam     1     .     .    .     1    .    1    1
Frank   .     1     1    1     .    1    1    .
Jay     .     .     1    .     1    .    1    1
Tom     .     .     .    1     1    1    .    .
Jerry   .     .     .    1     .    1    .    .
```



GRAPH MANIPULATION:

- Among the important techniques of graph manipulation are:
 - i) remove specific nodes/vertices.
 - ii) generate subgraph.
 - iii) join graphs.
 - iv) modify the nodes data.
 - v) modify the edge data.



LINK AND NETWORK ANALYSIS:

- Link refer to a relationship between two entities.
- Network refer to a collection of entities and links between them.
- Graph mining provide a basis for link and network analysis.
- Example:
 - i) Graph mining can be used to interpret networks by determining clustering of nodes.
 - ii) Graph mining useful in determining how densely nodes connected in network data.
 - iii) Graph mining is useful in identifying the layout structure on network data.



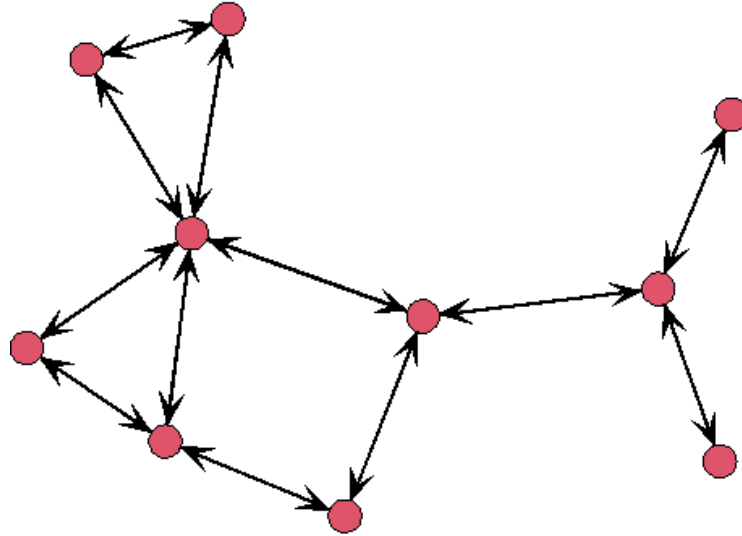
NODE PROMINENCE ANALYSIS:

- Networks data are interesting because of their specific structural patterns.
- The structures will affect the characteristics of nodes/members in the network.
- **Example:** A person who is connected to many other members of a network is likely to view entire network in a different context than somebody who is relatively isolated from the other members.
- Thus, by examining the location of individual network members, we can assess the prominence of those nodes.
- A node is prominent if their ties make that node visible to the other members in the network.



NODE PROMINENCE ANALYSIS:

- Among the measures that can be used to measure prominence node :



i) Degree Centrality:

- Based on this measure, nodes that have more direct ties are more prominent.

ii) Closeness Centrality:

- Based on this measure, nodes are more prominent if they are more closer to all other nodes in the network.



NODE PROMINENCE ANALYSIS:

iii) Betweenness Centrality:

- Nodes are more prominent if their location sits 'between' pairs of other nodes in the network.
- A paths between the other nodes has to go through prominent node

iv) Eigenvector Centrality Scores:

- Measures the transitive influence of nodes.
- A high eigenvector score means that a node is connected to many nodes who themselves have high scores.

v) Information Centrality Scores:

- Nodes with higher information centrality have a greater control over the flow of information within a network.
- Its implies the existence of a large number of short paths within the network structure.

vi) Flow Betweenness Scores:

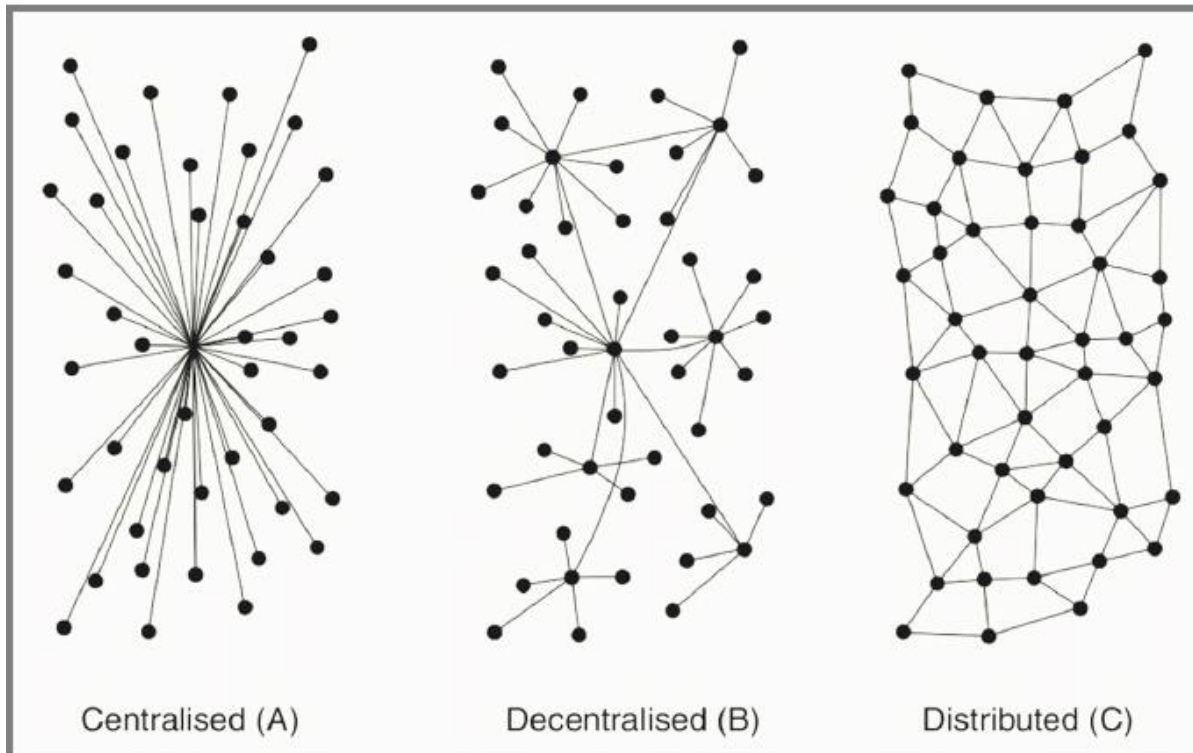
- Measures the total maximum flow of a particular nodes.



NODE PROMINENCE ANALYSIS:

vi) Centralization:

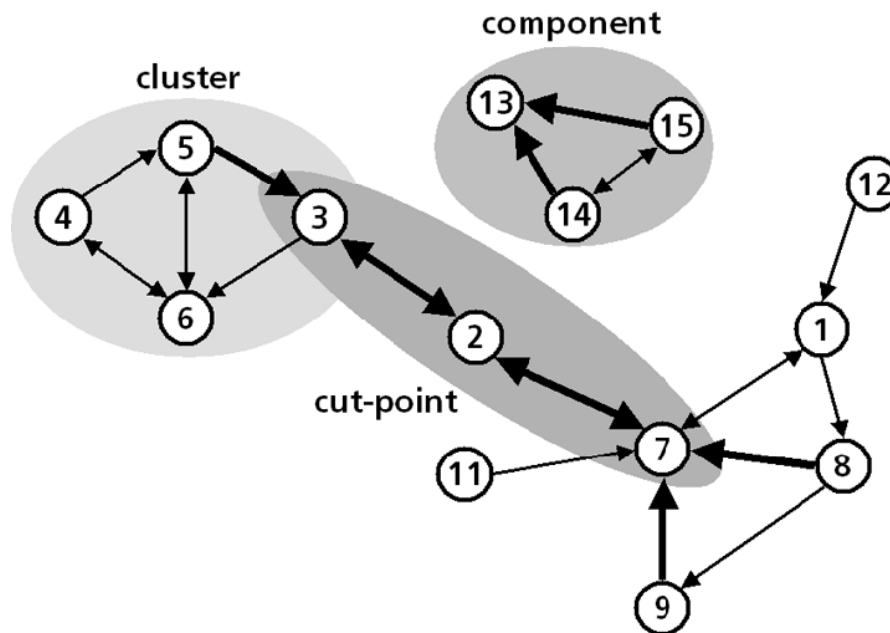
- Based on the given nodes measure, we can analyze the centralization behaviors of a network.
- Centralization provides a measure of the variability of the network centrality.



NODE PROMINENCE ANALYSIS:

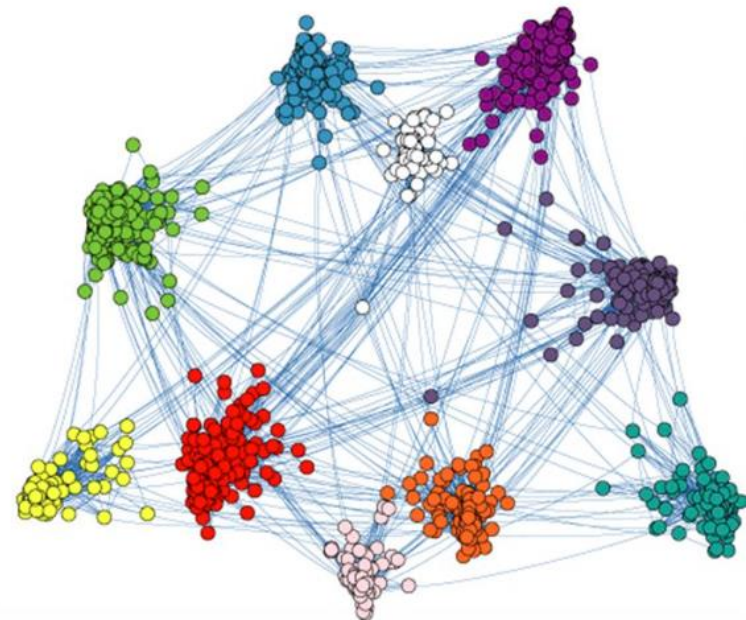
vii) Cutpoints:

- A cutpoint refer to a node that, if we delete it, the number of components in the network will be increase.
- Cutpoints is a node with an important position that connects different parts of the network.
- If a cutpoint node is removed, it will result in two subsets of nodes that will not be able to communicate with each other.



SUBGROUPS ANALYSIS:

- Network data can be form by several densely connected subgroups that are themselves only connected via less common ties.
- **Example:** Friendship subnetworks can be found between acquaintances.
- These subgroups contain different information between each other.
- Thus, for a large network data, it is important to be able to define and identify such subgroups for further analysis.



SUBGROUPS ANALYSIS:

- In a real-application, for a large networks, the existence of subgroup structure usually difficult to be detected clearly.
- Thus, a systematic analysis need to done to determine the existence of subgroup structure.
- Subgroup structure can be detected based on the concept of social cohesion.
- Cohesive subgroups refer to a sets of nodes that are tied together through frequent, strong, and direct ties.

- Two most important types of cohesive subgroups are:

i) Cliques:

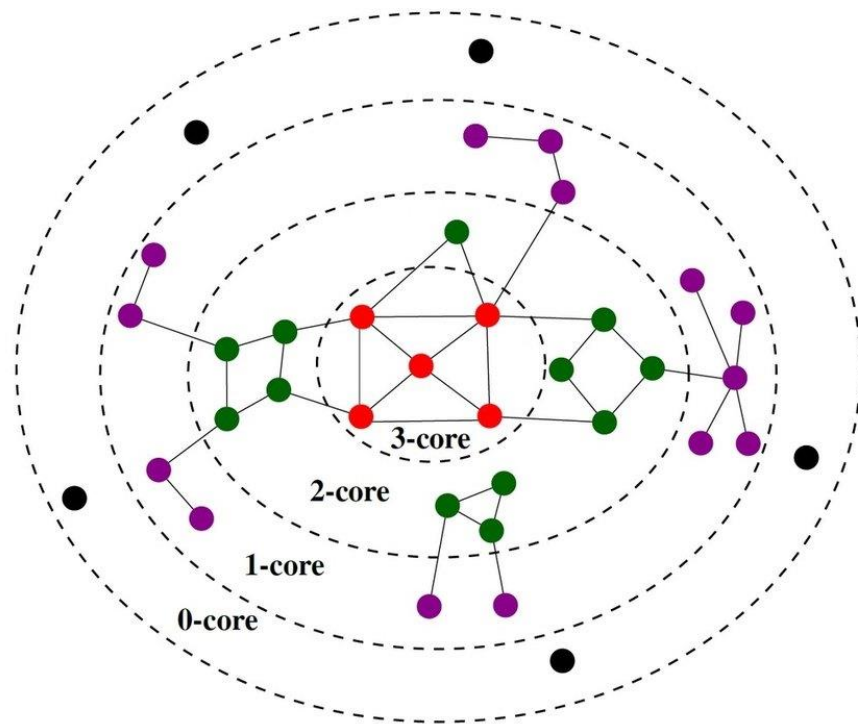
- A clique is a maximally complete subgraph.
- It is a subset of nodes that have all possible ties among them.



SUBGROUPS ANALYSIS:

ii) k-Cores:

- Cliques sometimes difficult to determine because they require a condition of maximally complete subgraph.
- k-core is modification of cliques which refer to a maximal subgraph where each vertex is connected to at least k other vertices in the subgraph.

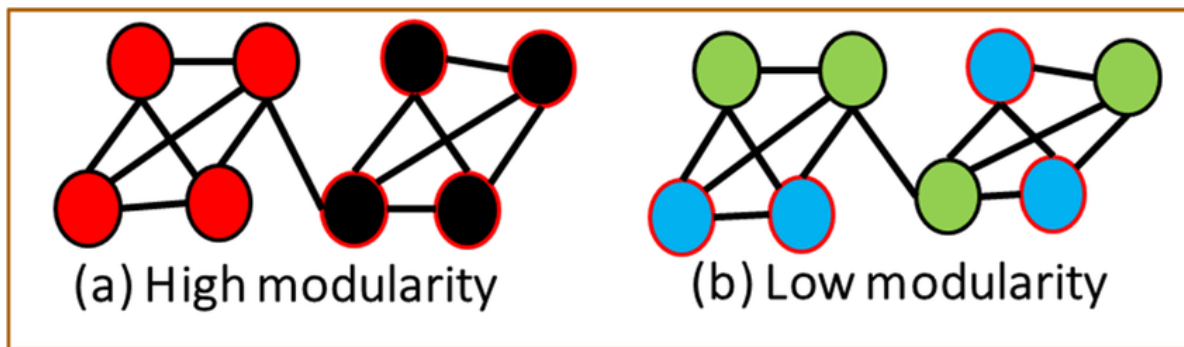


SUBGROUPS ANALYSIS:

Other approach to investigate a subgroup structure is based on community detection which include the techniques of:

i) Modularity:

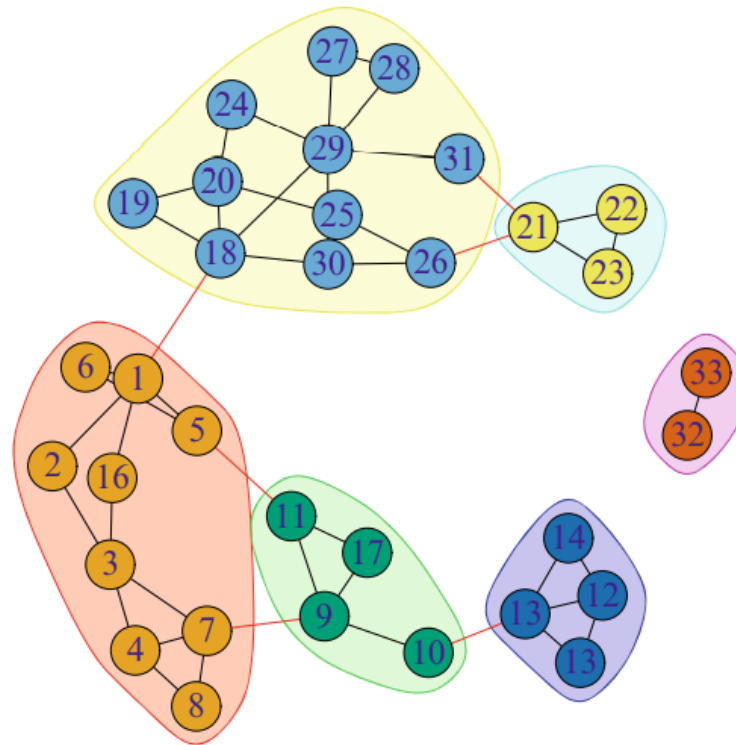
- This is a measure of the structure of the network, in which nodes exhibit clustering if there exist greater density within the clusters or less density between them.
- Value close to 1 indicates strong community structure. While, value equal to 0 indicates the community division is just a random.



SUBGROUPS ANALYSIS:

ii) Community Detection:

- A community in graph refer to a subset of nodes that are densely connected to each other and loosely connected to the nodes in the other communities.



REFERENCES:

- Brath, R., Jonker, D. (2015). *Graph analysis and visualization: Discovering business opportunity in linked data*. Wiley.
- Csardi, G., Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal Complex Systems*, 1695.
- Gosnell, D., Broecheler, M. (2020). *The practitioner's guide to graph data: Applying graph thinking and graph technologies to solve complex problems*. O'Reilly Media
- Kolaczyk, E.D., Csárdi, G. (2020). *Statistical analysis of network data with R. Second Edition*. Cham: Springer.
- Luke, D.A. (2015). *A user's guide to network analysis in R*. Cham: Springer.
- Samatova, N.F., Hendrix, W., Jenkins, J., Padmanabhan, K., Chakraborty, A. (2014). *Practical graph mining with R*. Boca Raton: CRC Press.



FINAL EXAM:

DATE: 17 FEBRUARY 2025
(**MONDAY**)

TIME: 9.00 AM - 12.00 PM

VENUE: BETA LABORATORY,
DEPARTMENT OF
MATHEMATICAL SCIENCES

GOOD LUCK!!!

