Statistical Analysis of Networks

Lecture 4 – Basic concepts

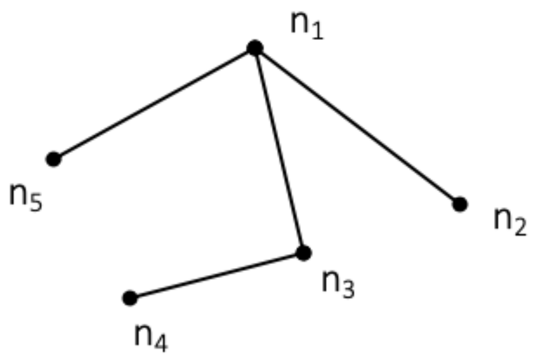
Graph and subgraph

A graph $H = (V_H, E_H)$ is a subgraph of another graph $G = (V,E)$ if $V_H \subseteq V$ and $E_H \subseteq E$.

An induced subgraph of G is a subgraph $G' = (V', E')$, where $V' \subseteq V$ is a pre-specified subset of vertices and $E' \subseteq E$ is the collection of all the edges (from the original graph) connecting pairs of vertices in that subset

Graph and subgraph

Graph



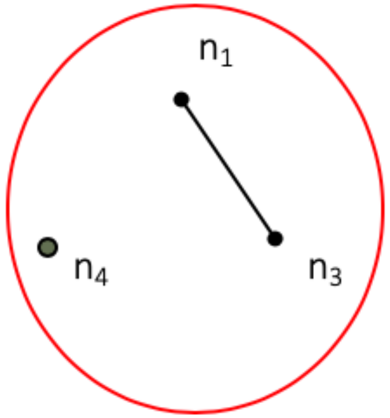
$N = (n_1, n_2, n_3, n_4, n_5)$

$L = (l_1, l_2, l_3, l_4)$

$l_1 = (n_1, n_2)$ $l_2 = (n_1, n_3)$

$l_3 = (n_1, n_5)$ $l_4 = (n_3, n_4)$

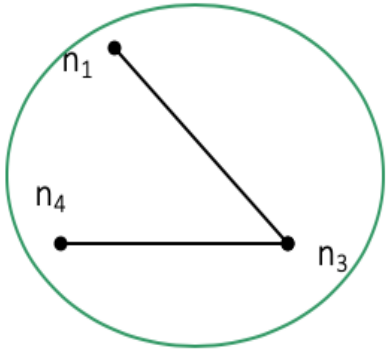
Subgraph: a subset of nodes and a subset of edges



$N_s = (n_1, n_3, n_4)$

$L_s = (l_2)$

Induced subgraph: all the edges from the graph among the subset of nodes



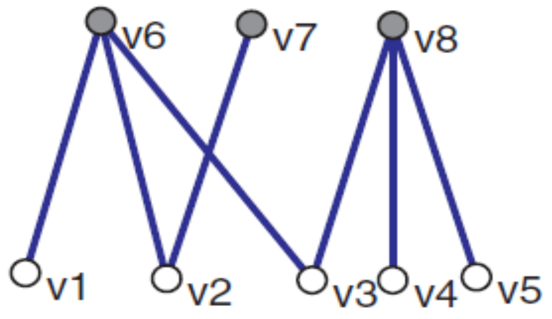
$N_s = (n_1, n_3, n_4)$

$L_s = (l_2, l_4)$

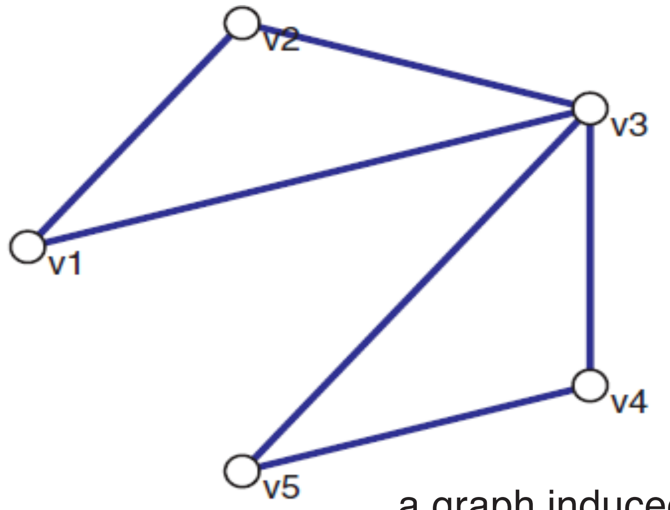
BIPARTITE GRAPH (AS DAVIS DATASET) AND INDUCED GRAPH

Usually a bipartite graph/network may be represented with at least one of two possible induced graphs.

Specifically, a graph $G_1 = (V_1,E_1)$ may be defined on the vertex set V_1 by assigning an edge to any pair of vertices that both have edges in E to at least one common vertex in V_2. Similarly, a graph G_2 may be defined on V_2.

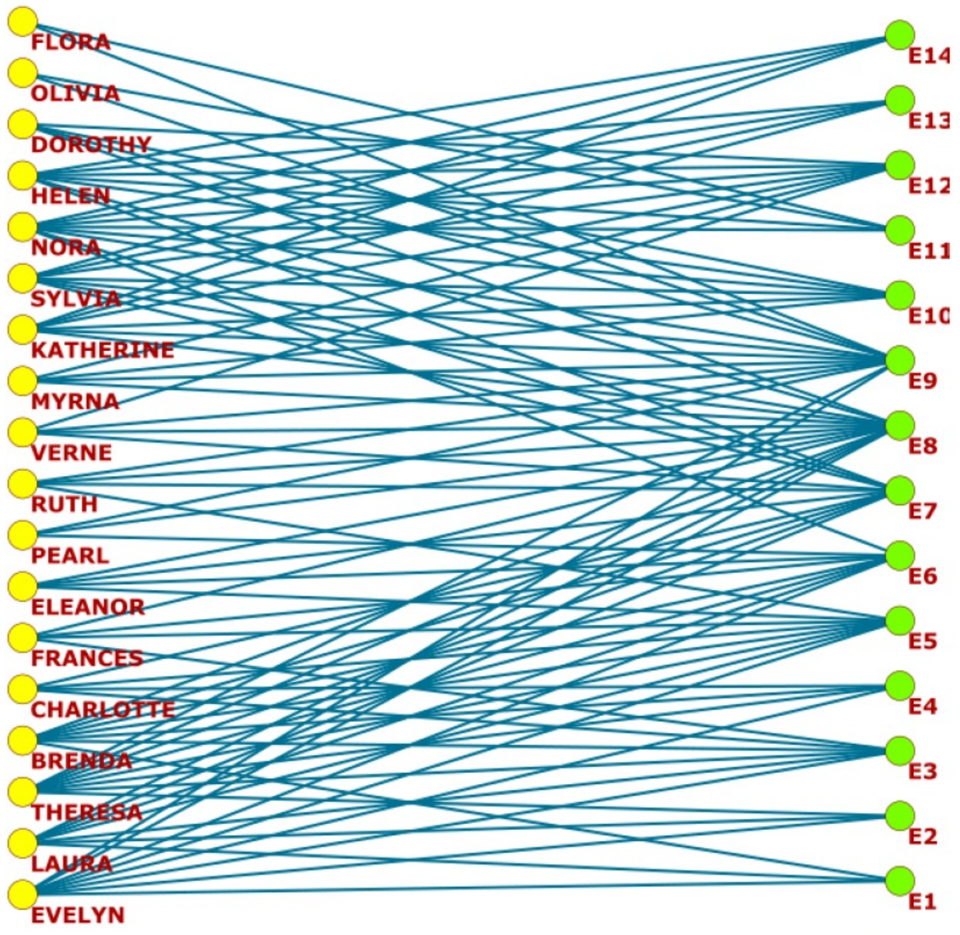


A bipartite graph



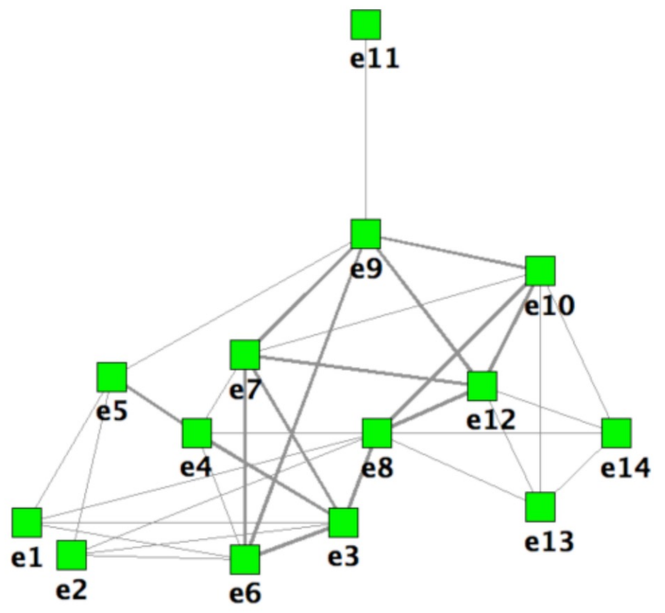
a graph induced by the bipartite graph on the ‘white’ vertex set.

BIPARTITE GRAPH - DAVIS DATASET



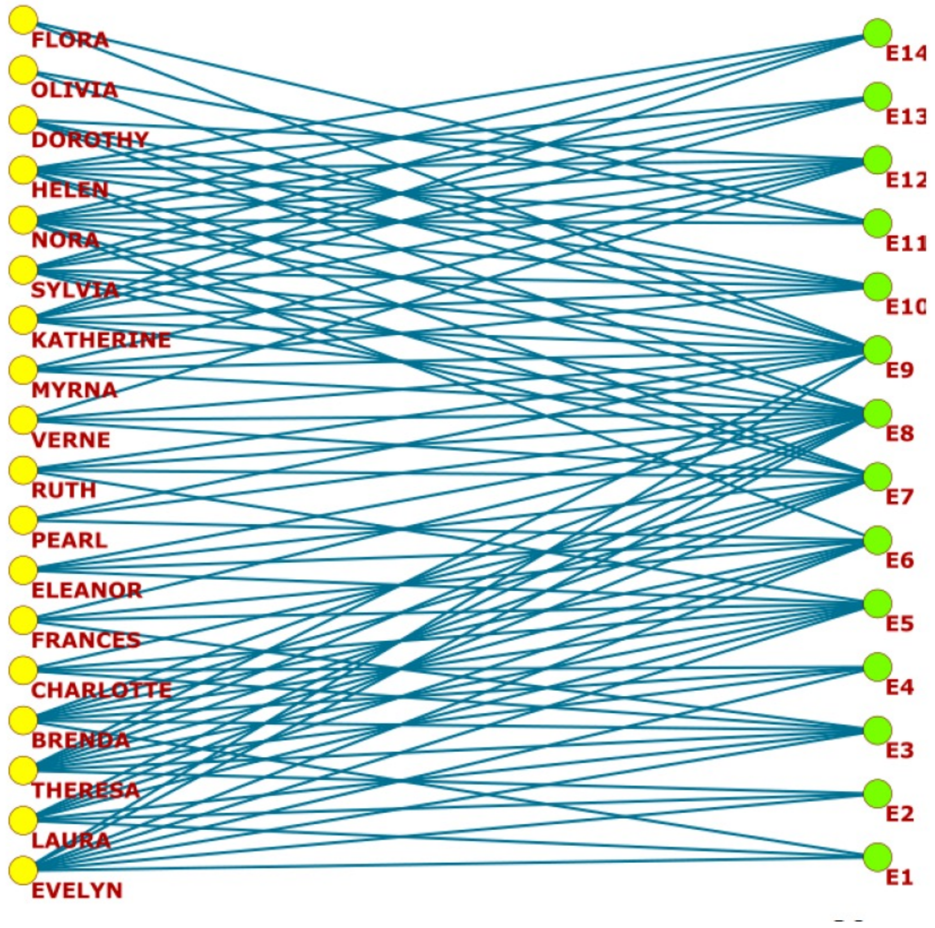
Event X is connected to event Y if they were both attended by the same woman, A.

The more women who attended the same two events, the stronger the connection is between the two events.



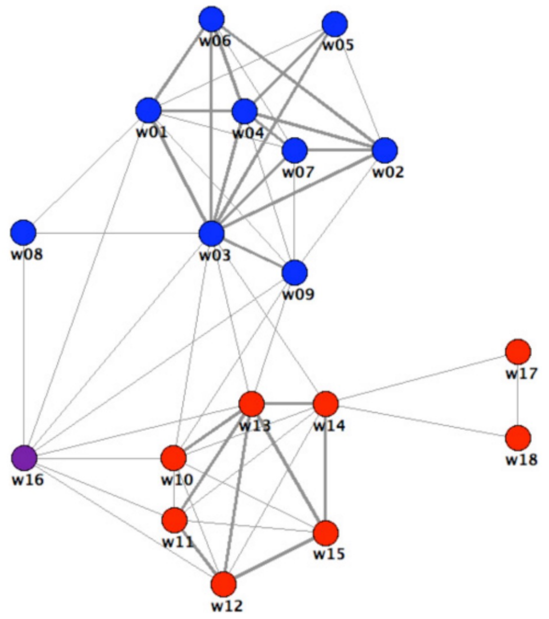
Graph of events based on attendance by people in common

BIPARTITE GRAPH - DAVIS DATASET



Woman X is connected to woman Y as they both attended Event E

The more events the women attended together, the stronger their tie is.



Graph of women based on common attendance at social events
(at least two events in common)

MOVEMENT IN A GRAPH: BASIC CONCEPTS

It is necessary to have a language for discussing the movement and the connectivity of a graph.

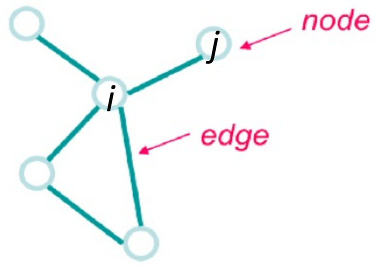
One of the most basic notions of connectivity is that of adjacency.

- Two vertices $i,j \in V$ are said to be adjacent if joined by an edge in E.

Also the notion of incidence is of interest:

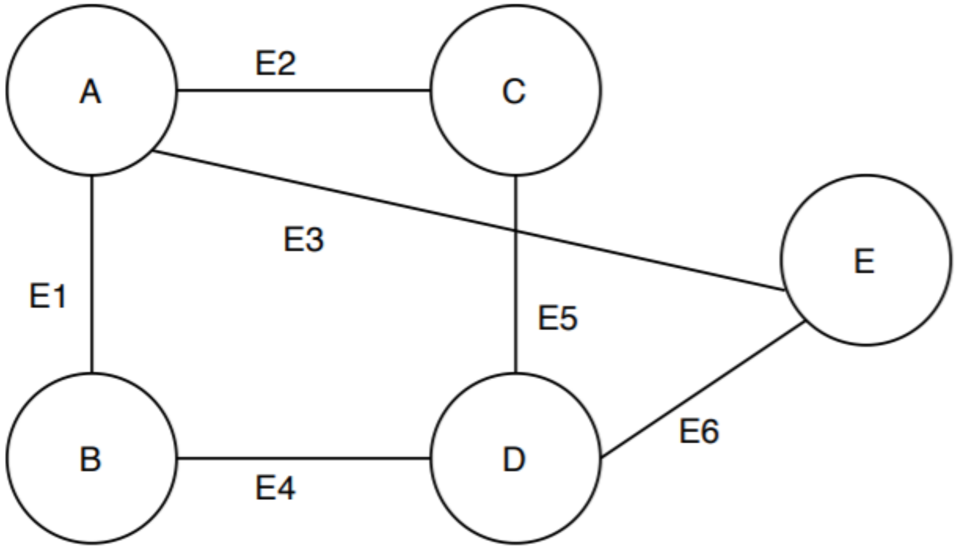
- a vertex $i \in V$ is incident of an edge $e \in E$ if i is an endpoint of e.

- two edges are incident if they share a common vertex



NETWORK AND GRAPH: BASIC CONCEPTS

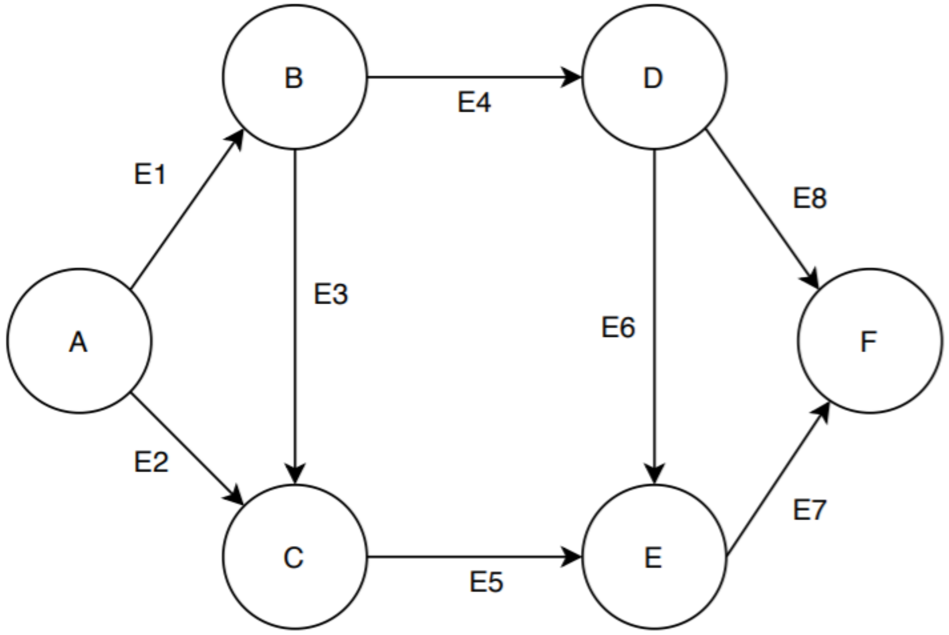
This undirected graph G
has a the vertex set V = (A, B, C, D, E)
and an edge set E = (E_1, E_2, E_3, E_4, E_5, E_6).



- The edges E_1, E_2, E_3 are incident edges as the vertex A is one of their common endpoints
- the vertex A is incident on the edges E_1, E_2, E_3
- The edges E_1 and E_4 are incident edges as the vertex B is one of their common endpoints
- Alternatively, the vertex B is incident on the edges E_1 and E_4

NETWORK AND GRAPH: BASIC CONCEPTS

This directed graph G (or digraph)
has a the vertex set V = (A, B, C, D, E, F)
and an edge set E = (E_1, E_2, E_3, E_4, E_5, E_6,E_7, E_8)



- The edges E_1 and E_2 are incident as they share the common vertex A
- The vertex A is incident on the edge E_1 and E_2
- The edge E_1 is incident on the vertex B from the vertex A
- Similarly, the edge E_2 is incident on the vertex C from the vertex A

MOVEMENT IN A GRAPH: WALKS

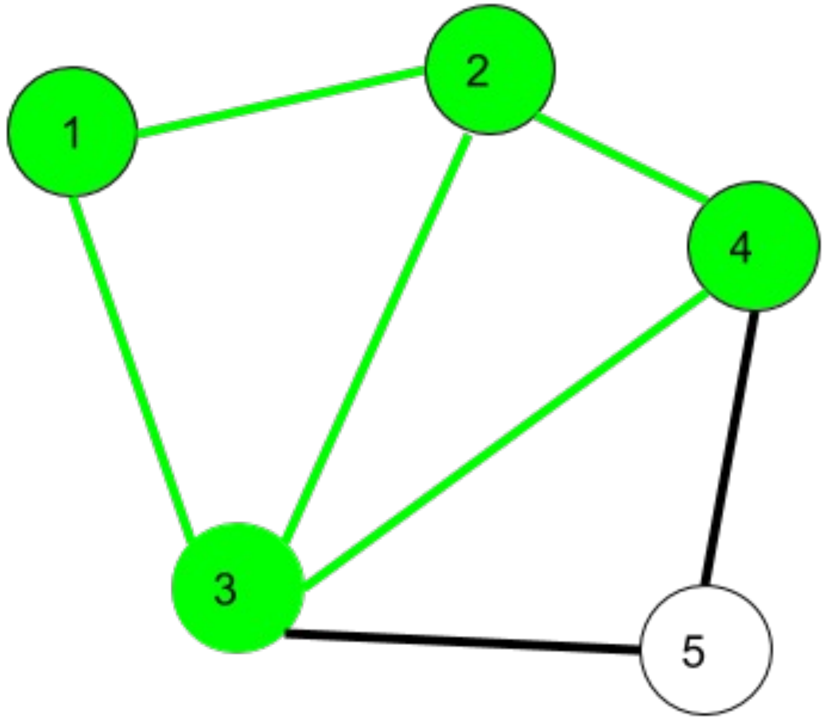
Walk from *node* n_i to node n_j: is an alternating sequence of vertices and edges that allows us to reach n_j from n_i (the two nodes can coincide)

Lenght of this walk is said *l = the # of edges*

- Vertices and Edges can be repeated
- in digraph: we speak about directed walks

Open walk- A walk is said to be an open walk if the starting and ending vertices are different
Closed walk- A walk is said to be a closed walk if the starting and ending vertices are identical

MOVEMENT IN A GRAPH: WALKS



Here, 1->2->3->4->2->1->3 is a walk

MOVEMENT IN A GRAPH: TRAILS

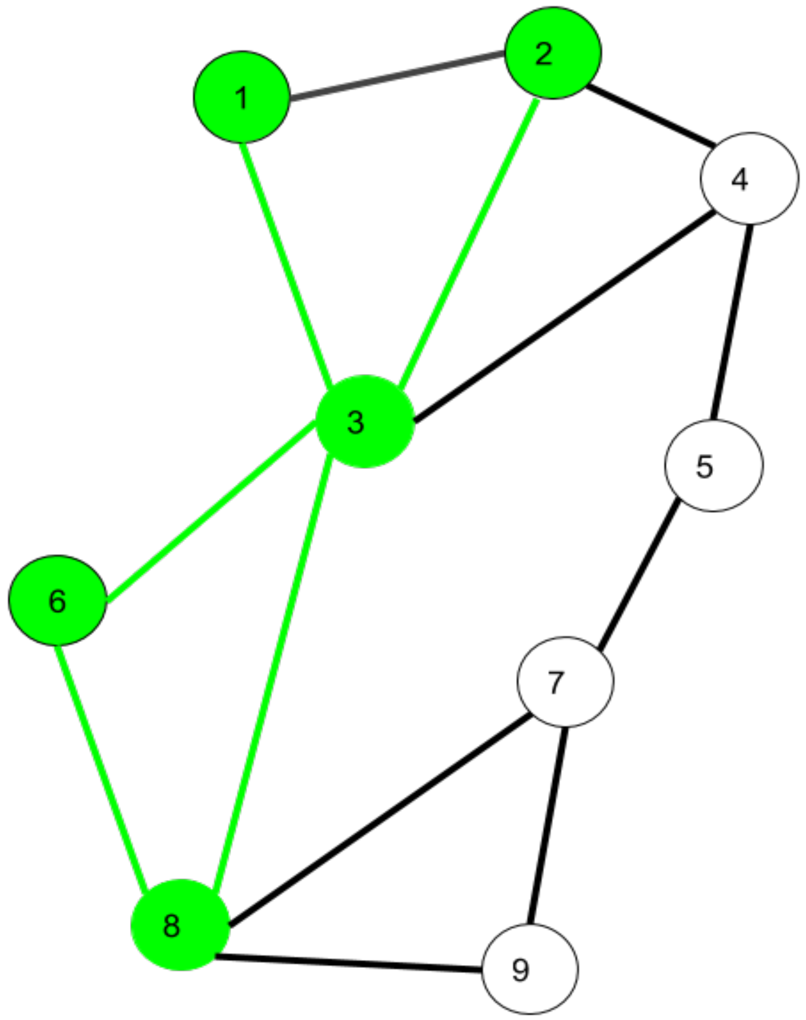
Trail from *node n_i* to node n_j*:* is a walk from n_i to n_j where no edges are traversed twice (the two nodes can coincide)

Lenght of this trail is said *l = the # of edges*

- Vertices can be repeated

Open trail- A trail is said to be an open trail if the starting and ending vertices are different
Closed trail- A trail is said to be a closed trail if the starting and ending vertices are identical (also called circuit)

MOVEMENT IN A GRAPH: WALKS, TRAILS



Here 1->3->8->6->3->2 is trail

Also 1->3->8->6->3->2->1 is a closed trail

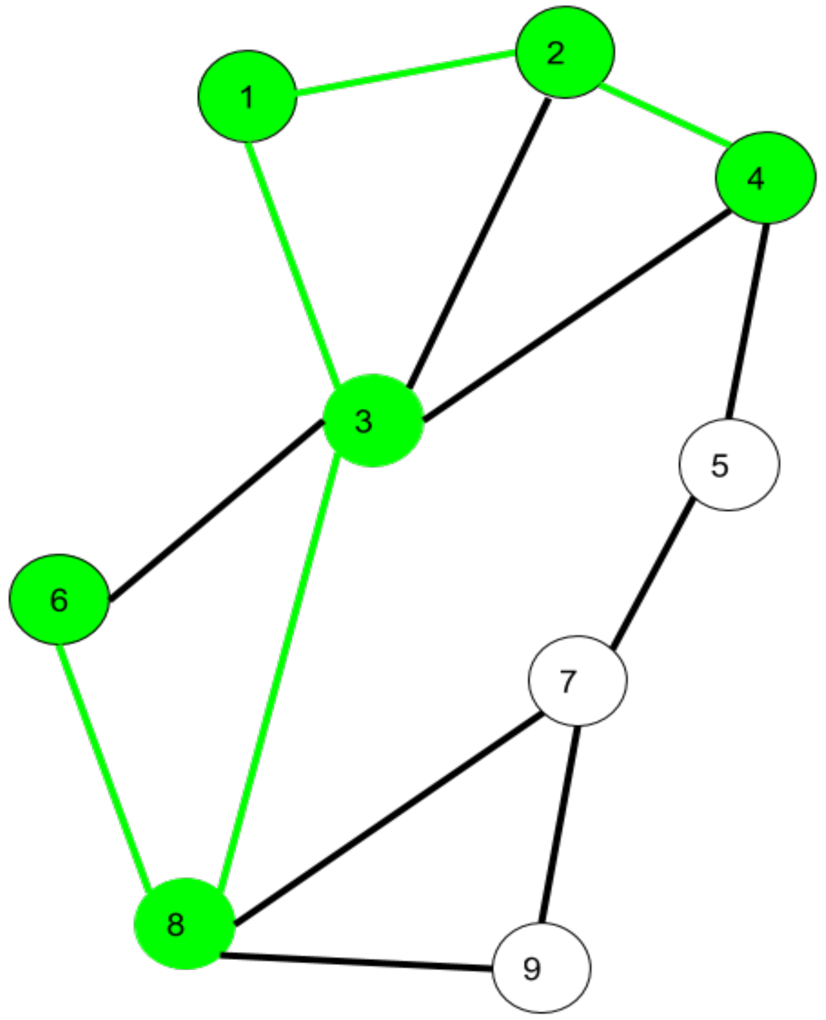
MOVEMENT IN A GRAPH: WALKS, TRAILS AND PATHS

- Path from *node n_i* to node n_j*:* a connected sequence of edges. Lenght *l = the # of edges* (with k edges: k-path)

- Path is a trail in which neither vertices nor edges are repeated i.e. if we traverse a graph such that we do not repeat a vertex and nor we repeat an edge. As path is also a trail, thus it is also an open walk

- In a directed graph, a path requires consistent direction of the arcs (otherwise it is a *semipath*)

MOVEMENT IN A GRAPH: WALKS, TRAILS AND PATHS



Here 6->8->3->1->2->4 is a Path

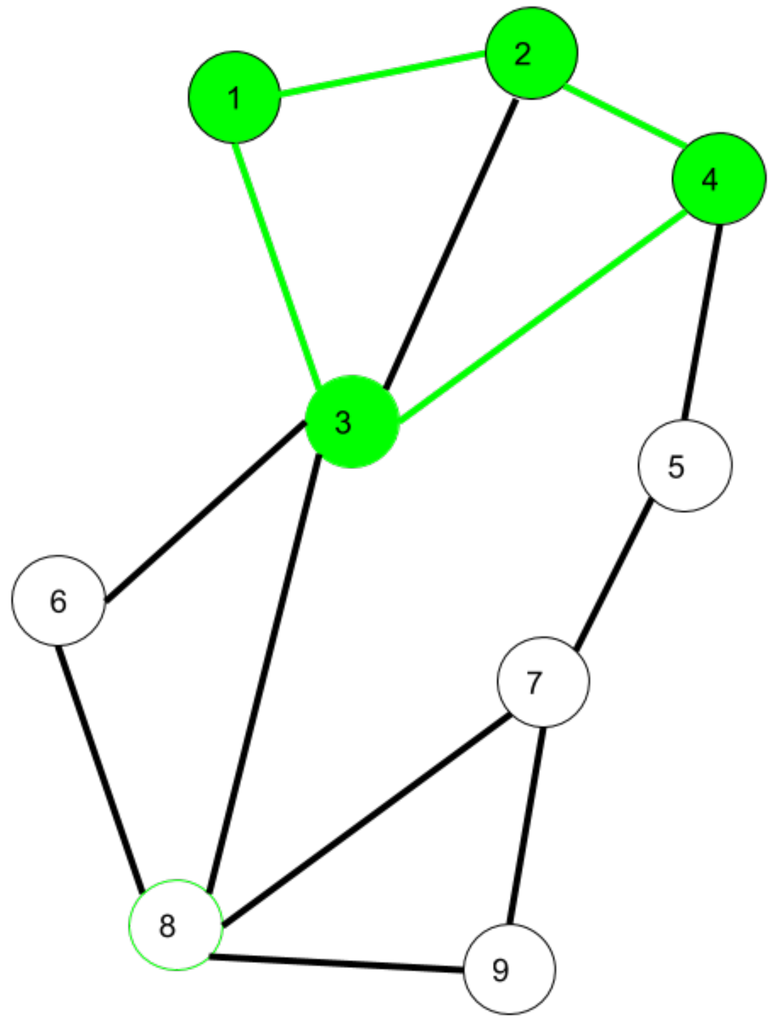
MOVEMENT IN A GRAPH: WALKS, TRAILS AND PATHS

- Cycle: a path of lenght > 2 for which the first and last nodes are the same and all other nodes are distinct

- Traversing a graph such that we do not repeat a vertex nor we repeat a edge but the starting and ending vertex must be same i.e. we can repeat starting and ending vertex only (then we get a cycle)

- In a directed graph, a path requires consistent direction of the arcs (otherwise it is a *semicycle*)

MOVEMENT IN A GRAPH: WALKS, TRAILS AND PATHS



Here 1->2->4->3->1 is a cycle

NODE REACHABILITY AND NETWORK CONNECTIVITY

- A node is reachable from an other node if there is a walk between them

Undirected

- Conneted graph (network): every node is reachable

Directed

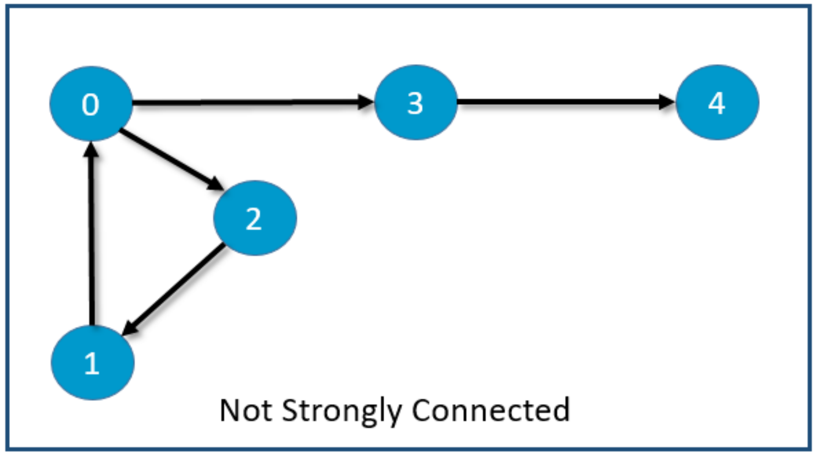
- *Weakly* connected digraph: the graph not considering arc direction is connected
- *Strongly* connected digraph: every node is reachable by a path (the path from n_i to n_j *can be different from the* path from n_j to n_i)

NODE REACHABILITY AND NETWORK CONNECTIVITY

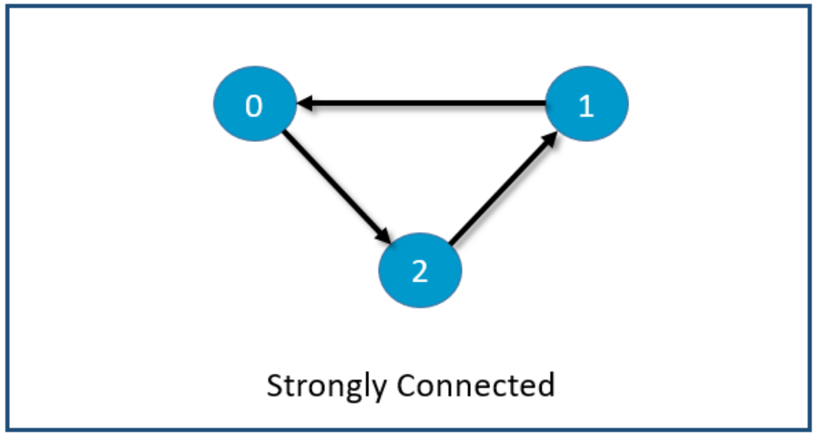
A directed graph is strongly connected if there is a path between any two pair of vertices.

Simply, if it is possible to reach any vertex starting from any other vertex in the graph that is called a Strongly Connected Graph

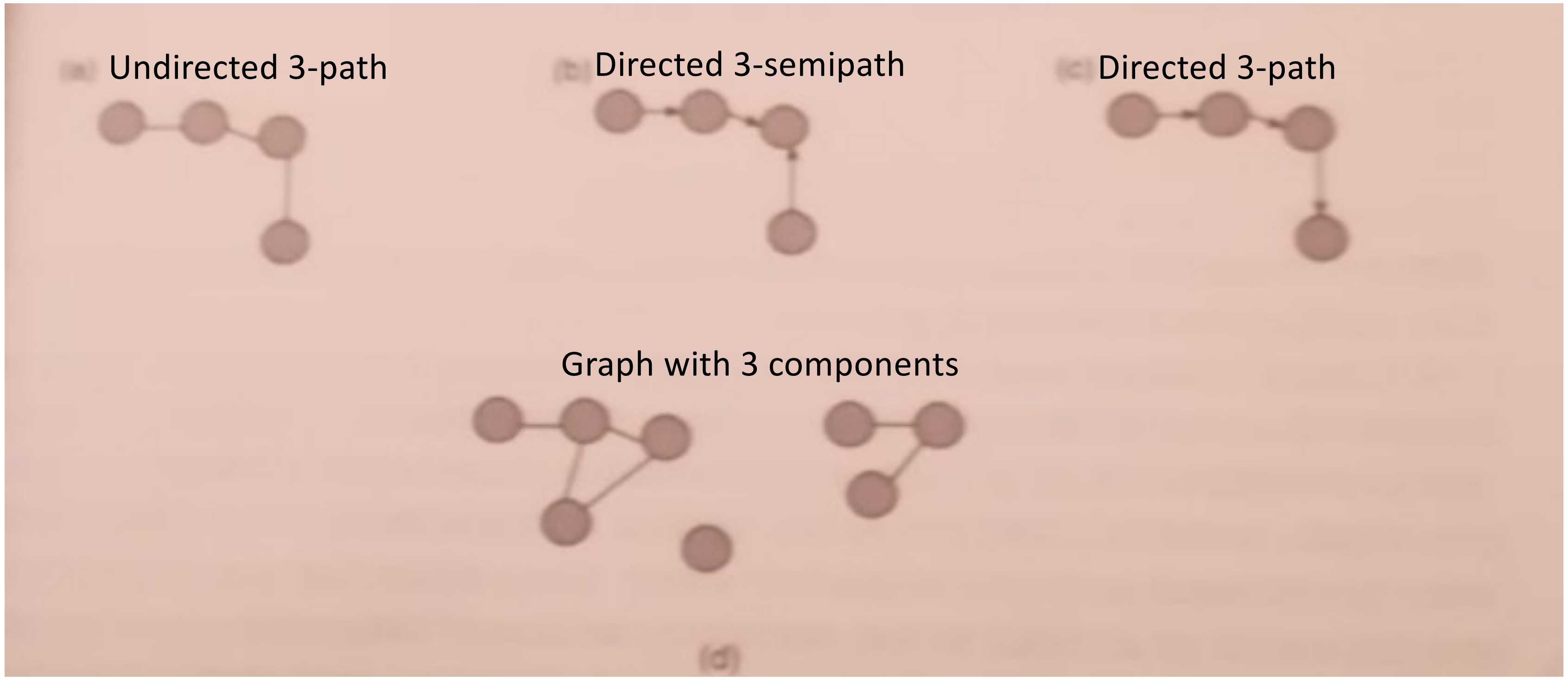
Graph 1:



Graph 2:



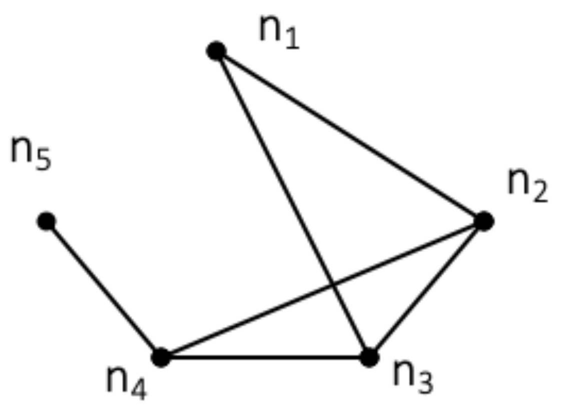
MOVEMENT IN A GRAPH: WALKS AND PATHS



Graph component: a *maximal* subgraph with paths between all nodes (clique)
Maximal: adding another node will not make the component larger
(there are no other nodes that are reachable from the nodes in the component)
Components divide the graph into separated regions

GEODESIC, DISTANCE, DIAMETER

- Geodesic: the shortest path between two nodes
- Geodesic distance: lenght of the geodesic path (if 2 nodes are not reachable, the distance is said to be *infinite*)
- Diameter: the longest shortest path (if there is a path from each node to each other node in the graph $\longrightarrow$ connected network)

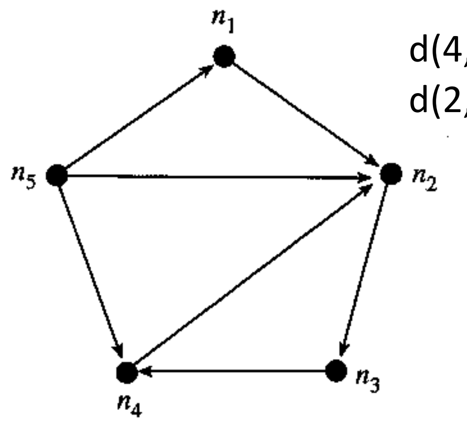


d(1,2) = 1	d(2,4) = 1
d(1,3) = 1	d(2,5) = 2
d(1,4) = 2	d(3,4) = 1
d(1,5) = 3	d(3,5) = 2
d(2,3) = 1	d(4,5) = 1

Graph diameter = max d(i,j) = d(1,5) = 3

GEODESIC, DISTANCE, DIAMETER IN DIGRAPH

Path from *node n_i* to node n_j can be different from path from *node* n_j to node n_i *(all arcs must point in the same direction):*



Directed walk	$n_5\ n_1\ n_2\ n_3\ n_4\ n_2\ n_3$
Directed path	$n_5\ n_4\ n_2\ n_3$
Semipath	$n_1\ n_2\ n_5\ n_4\ n_3$
Cycle	$n_2\ n_3\ n_4\ n_2$
Semicycle	$n_1\ n_2\ n_5\ n_1$

d(4,2) = 1 $\longrightarrow$
d(2,4) = 2

geodesic and geodesic distance *d(i, j) and d(j,i)* can be different (if there is no path than no geodesic, distance and diameter = ∞)

geodesic distance and diameter are defined only for strongly connected digraph

INTERESTING RESULTS ON GEODESIC DISTANCE

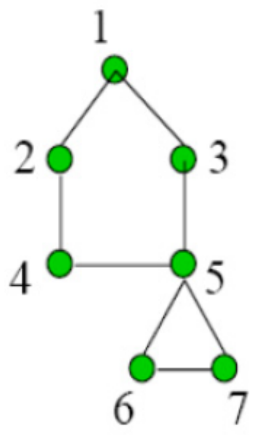
Networks with a very large number of vertices can have quite short shortest paths among vertices. For example:

- The average length of shortest path of the WWW, with over 800 million vertices, was around 19. Albert, R., Jeong, H., and Barabasi, A.-L. (1999): Diameter of the World-Wide Web. *Nature*, **401**, 130-131.
- Social networks (whom do you know) with over six billion individuals are believed to have a average length of shortest path around six. Milgram, S. (1967): The small-world problem. *Psychol. Today*, **2**, 60-67.

GRAPHS AND MATRICES

- **graphs/networks** and **matrices** are equivalent representations of relational data
- The most common matrix used to represent a network is the adjacency matrix $\mathbf{A}$ $(n \times n)$
- The adjacency matrix has as many rows and columns as the number of nodes and its properties depend on the type of graph
- In case of binary networks, the elements a_{ij} of $\mathbf{A}$ are $\neq 0$ if the link between nodes i and j exists ($a_{ij} = 1$ if an edge is present, 0 otherwise)
- In case of weighted networks, a_{ij} = value attached to the edge, 0 otherwise

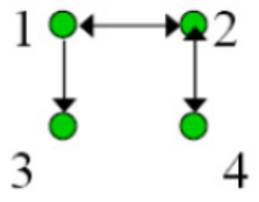
GRAPHS AND MATRICES – BINARY UNDIRECTED RELATIONS



	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	1	0	0	1	0	0	0
3	1	0	0	0	1	0	0
4	0	1	0	0	1	0	0
5	0	0	1	1	0	1	1
6	0	0	0	0	1	0	1
7	0	0	0	0	1	1	0

- The related **adjacency matrix** is a binary and symmetric matrix, with: $a_{ij} = a_{ji}$

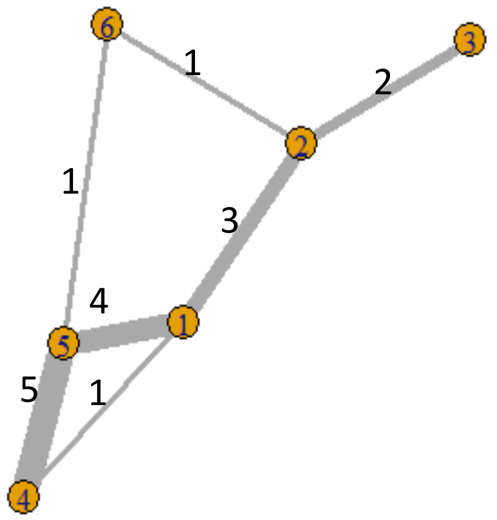
GRAPHS AND MATRICES – BINARY DIRECTED RELATIONS

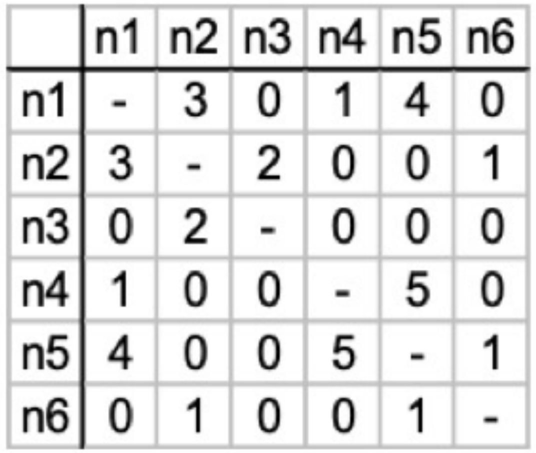


	1	2	3	4
1	0	1	1	0
2	1	0	0	1
3	0	0	0	0
4	0	1	0	0

- The related **adjacency matrix** is a binary and asymmetric matrix, with: $a_{ij} \neq a_{ji}$

GRAPHS AND MATRICES – WEIGHTED UNDIRECTED RELATIONS





	n1	n2	n3	n4	n5	n6
n1	-	3	0	1	4	0
n2	3	-	2	0	0	1
n3	0	2	-	0	0	0
n4	1	0	0	-	5	0
n5	4	0	0	5	-	1
n6	0	1	0	0	1	-

- The related **adjancecy matrix** is a valued and symmetric matrix, with $a_{ij} = a_{ji}$

GRAPHS AND MATRICES – INCIDENCE MATRIX

Incidence matrix B (*n* x *l*)
(*n* rows and *l* column):

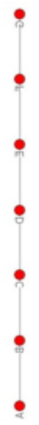
b_{ij} = 1 if node i is incident to edge j (*i* is the terminal node) , 0 otherwise

All the information of the graph G are cointained in the matrices A and B

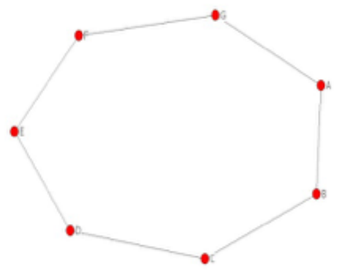
	l1	l2	l3	l4	l5	l6
n1	1	1	0	0	0	0
n2	0	0	1	0	0	0
n3	0	0	1	0	0	0
n4	0	0	0	1	1	0
n5	1	0	0	1	0	1
n6	0	1	0	0	1	1

PARTICULAR GRAPHS AND ASSOCIATED ADJANCECY MATRICES (BINARY UNDIRECTED RELATIONS)

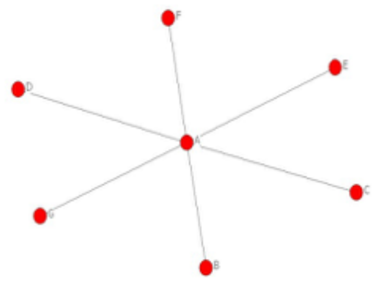
Line graph



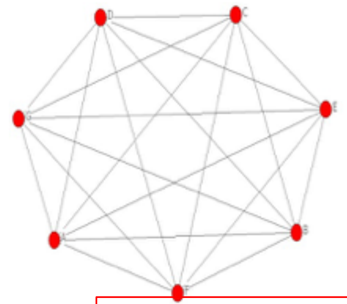
Circle graph



Star graph



Complete graph



Node Degree = # of nodes adjacent to i

	A	B	C	D	E	F	G	Gradi
A	0	1	0	0	0	0	0	1
B	1	0	1	0	0	0	0	2
C	0	1	0	1	0	0	0	2
D	0	0	1	0	1	0	0	2
E	0	0	0	1	0	1	0	2
F	0	0	0	0	1	0	1	2
G	0	0	0	0	0	1	0	1
	1	2	2	2	2	2	1	12

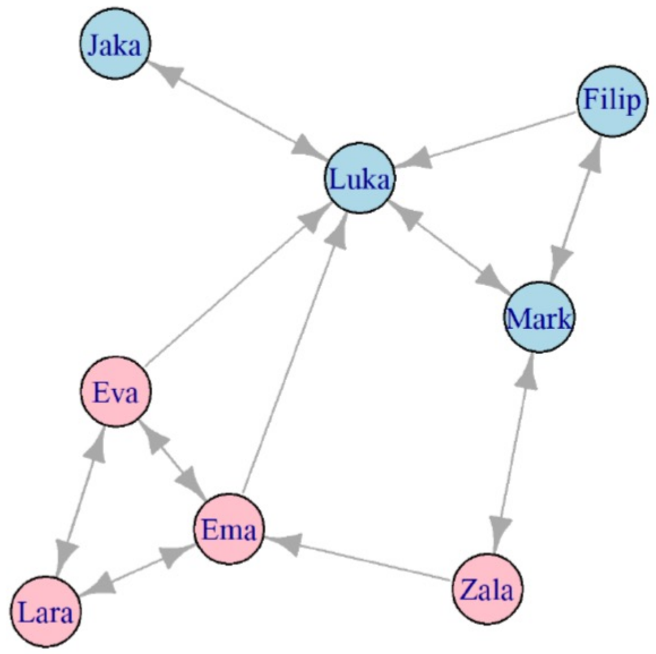
	A	B	C	D	E	F	G	Gradi
A	0	1	0	0	0	0	1	2
B	1	0	1	0	0	0	0	2
C	0	1	0	1	0	0	0	2
D	0	0	1	0	1	0	0	2
E	0	0	0	1	0	1	0	2
F	0	0	0	0	1	0	1	2
G	1	0	0	0	0	1	0	2
	2	2	2	2	2	2	2	14

	A	B	C	D	E	F	G	Gradi
A	0	1	1	1	1	1	1	6
B	1	0	0	0	0	0	0	1
C	1	0	0	0	0	0	0	1
D	1	0	0	0	0	0	0	1
E	1	0	0	0	0	0	0	1
F	1	0	0	0	0	0	0	1
G	1	0	0	0	0	0	0	1
	6	1	1	1	1	1	1	12

	A	B	C	D	E	F	G	Gradi
A	0	1	1	1	1	1	1	6
B	1	0	1	1	1	1	1	6
C	1	1	0	1	1	1	1	6
D	1	1	1	0	1	1	1	6
E	1	1	1	1	0	1	1	6
F	1	1	1	1	1	0	1	6
G	1	1	1	1	1	1	0	6
	6	6	6	6	6	6	6	42

$2 * L$ (# of lines) $= \sum$ degrees

NETWORK DATA STRUCTURE - NOT ONLY MATRICES
LIST OF ADJACENCY NODES

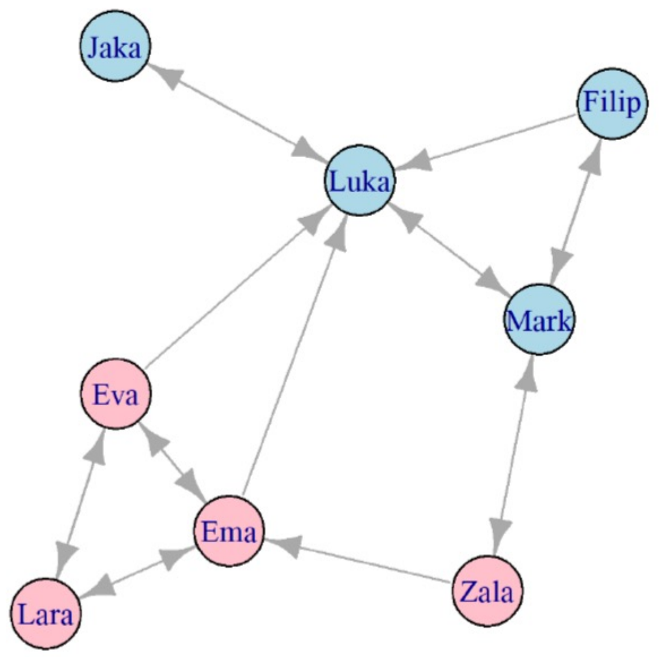


```
*Vertices 8          *Arcslist
 1 "Luka"            1 2 3
 2 "Jaka"            2 1
 3 "Mark"            3 1 4 5
 4 "Filip"           4 1 3
 5 "Zala"            5 3 7
 6 "Eva"             6 1 7 8
 7 "Ema"             7 1 6 8
 8 "Lara"            8 6 7
```

More convenient data structure, especially with large network

NETWORK DATA STRUCTURES - NOT ONLY MATRICES

LIST OF EDGES/ARCS



```
*Vertices 8   4 1 1   6 7 1
 1 "Luka"     6 1 1   8 7 1
 2 "Jaka"     7 1 1   6 8 1
 3 "Mark"     1 2 1   7 8 1
 4 "Filip"    1 3 1
 5 "Zala"     4 3 1
 6 "Eva"      5 3 1
 7 "Ema"      3 4 1
 8 "Lara"     3 5 1
*Arcs         7 6 1
2 1 1         8 6 1
3 1 1         5 7 1
```

What is the third column?

= Edge weight