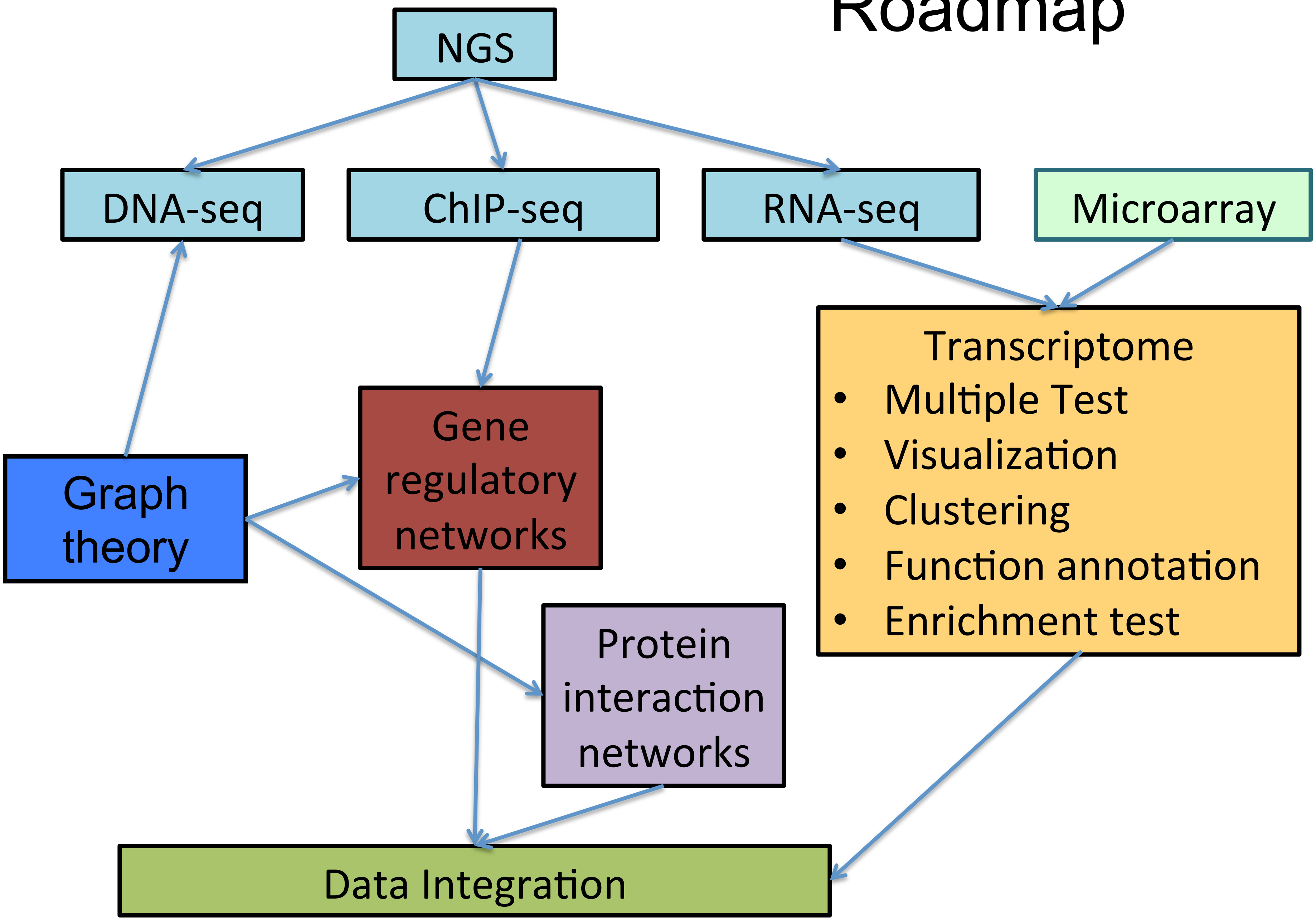
Graph Theory

Lecture 1

Roadmap

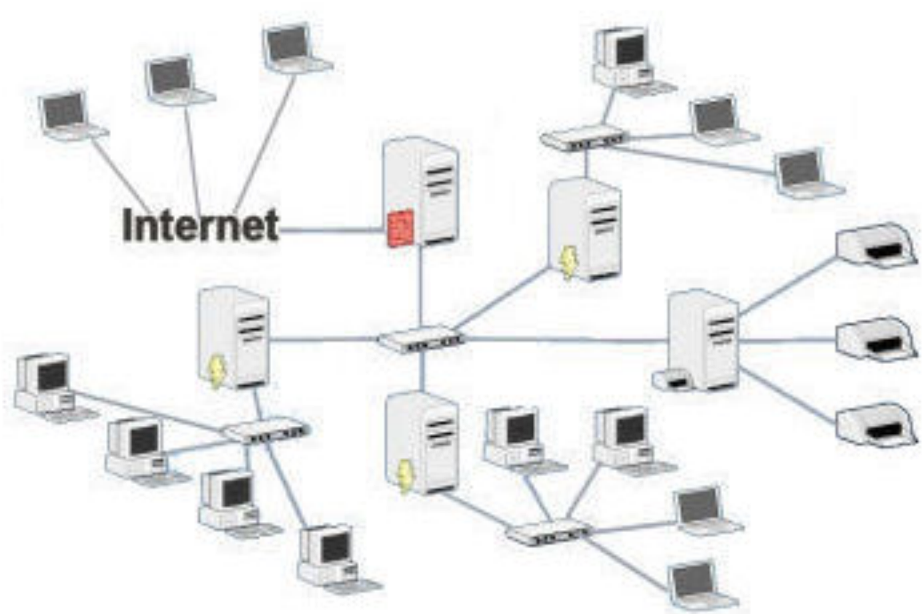


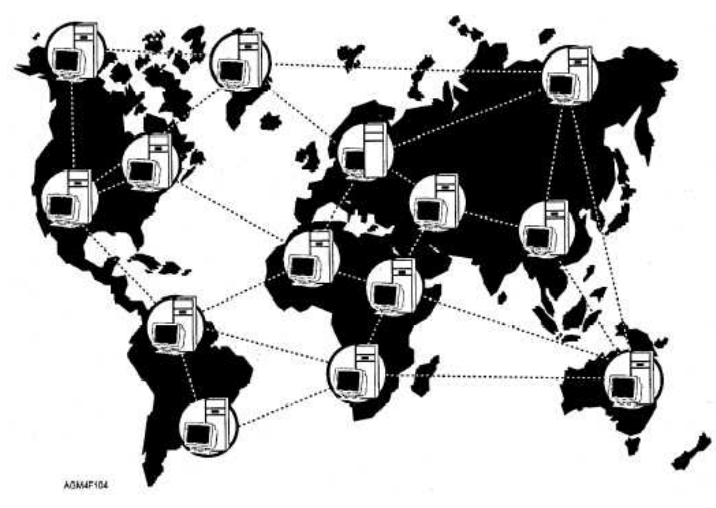
NGS
DNA-seq
ChIP-seq
RNA-seq
Microarray
Transcriptome
• Multiple Test
• Visualization
• Clustering
• Function annotation
• Enrichment test
Graph theory
Gene regulatory networks
Protein interaction networks
Data Integration

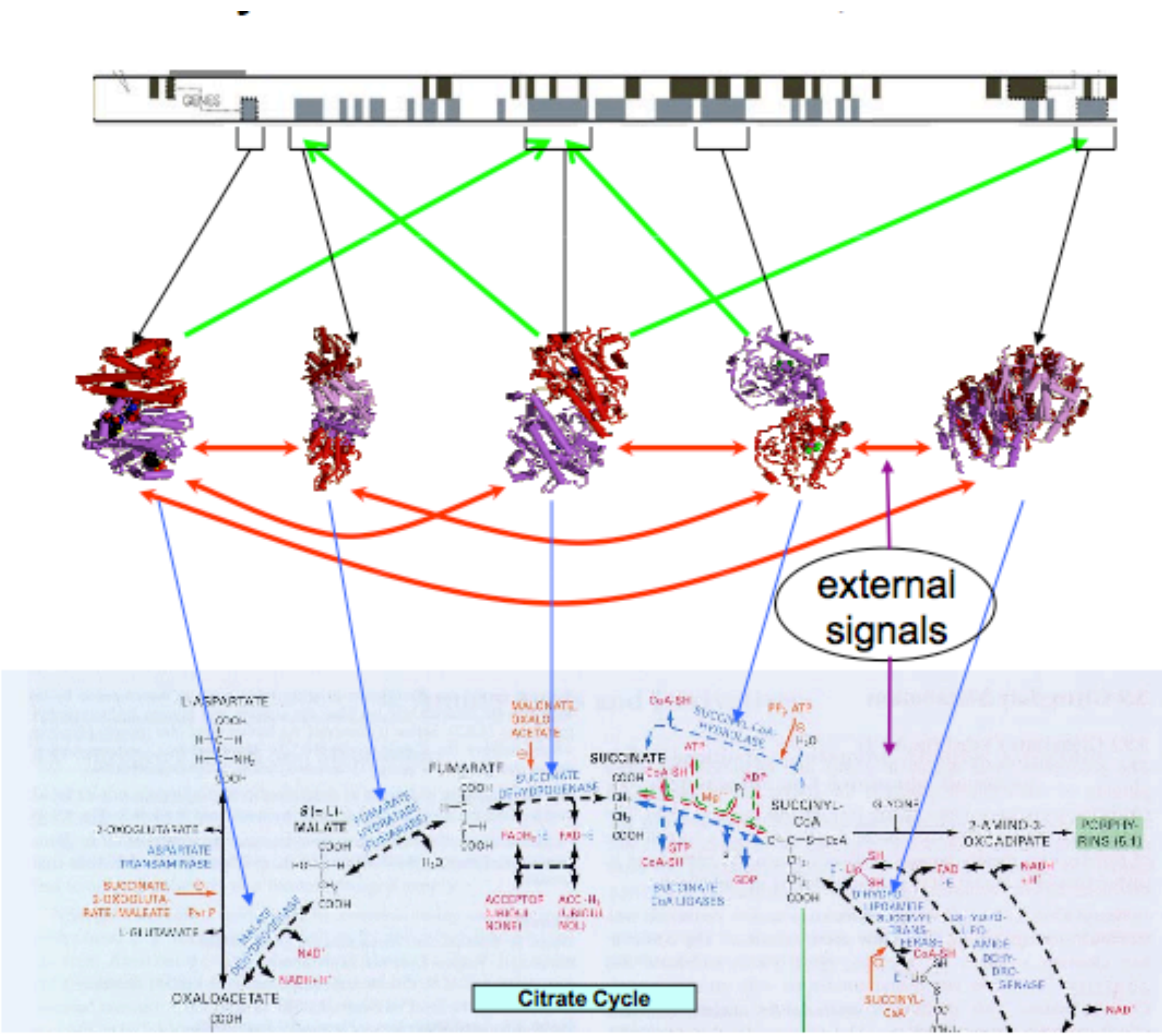
Many complicate systems have an underlying network topology

- Computer networks
- Social networks
- Biological networks
  - Food webs (chains)
  - Gene networks
  - Protein interaction networks
  - Signal transduction networks

Computer networks







GENOME

gene regulation

PROTEOME

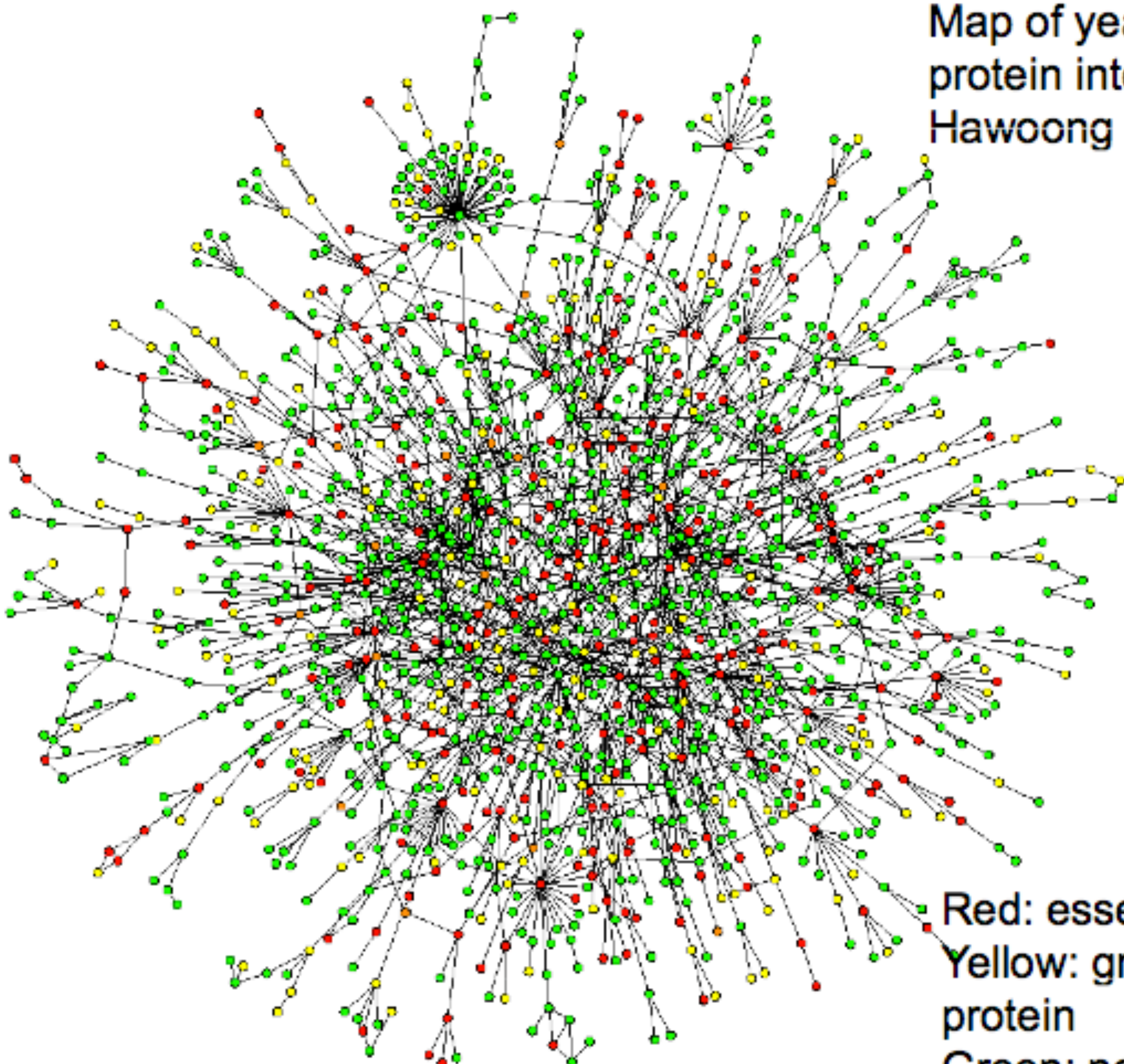
protein-protein interactions

signal transduction

METABOLISM

Bio-chemical reactions

Map of yeast protein-protein interactions, by Hawoong Jeong



Red: essential protein
Yellow: growth- affecting protein
Green: non-essential protein

Why study networks?

- It is increasingly recognized that complex systems cannot be described in a reductionist view.
- Understanding the behavior of such systems starts with understanding the topology of the corresponding network.
- Topological information is fundamental in constructing realistic models for the function of the network.

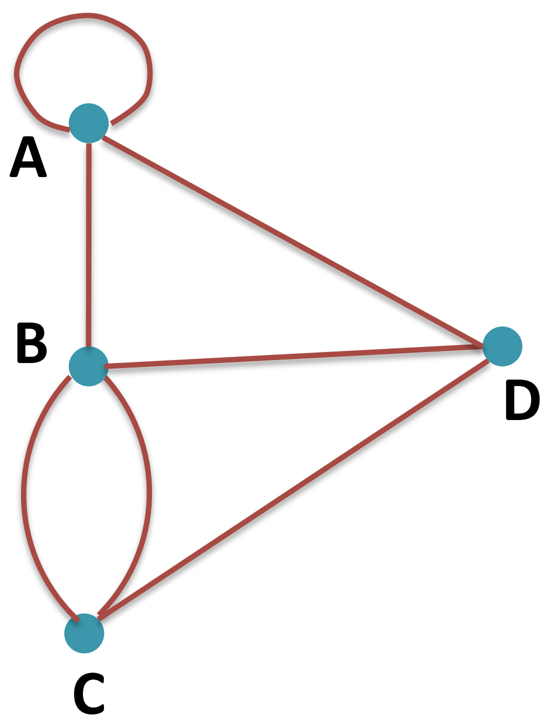
Network related questions

- How do we determine or infer network topology ? How do we build a network?
- How can we quantitatively describe large networks?
- How did networks get to be the way they are?
- What are the consequences of a specific network organization?

Graph Theory

- Some basic concepts of Graph theory
- Some examples of Special graphs
- Graph paths and cycles
- Graph connectivity
- Tree and Bipartite graph
- Network models

Graph concepts



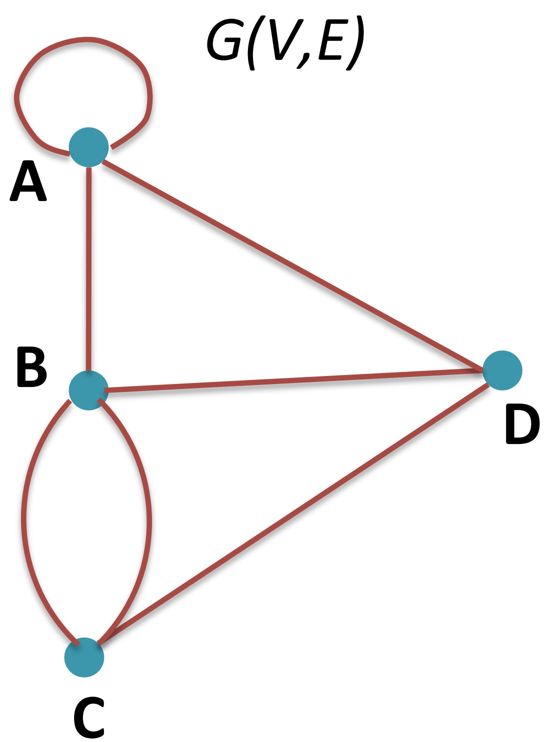
- Graphs are made up by vertices (nodes) and edges (links).
- An edge connects two vertices, or a vertex with itself.

$G=(V,E)$

V: a finite set of vertices.

E: edges of the graph

Graph concepts: some terms



- **Order** of graph *G*: the number of all vertices,

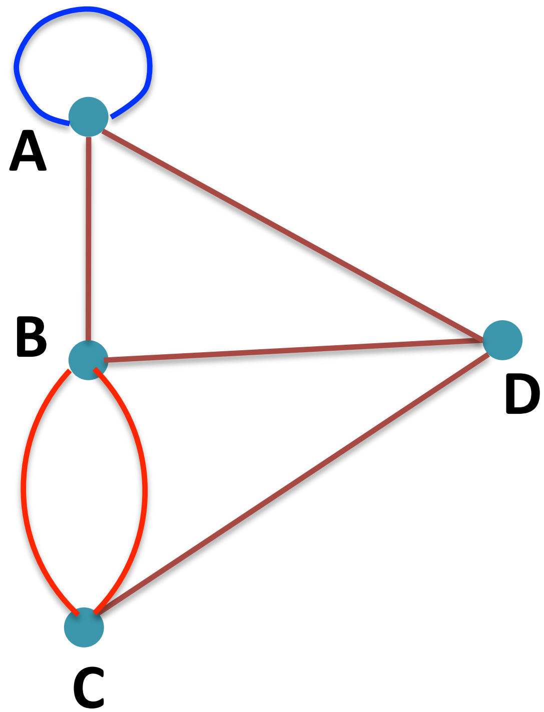
  $v_G = |V|$

- **Size** of graph: the number of all edges,

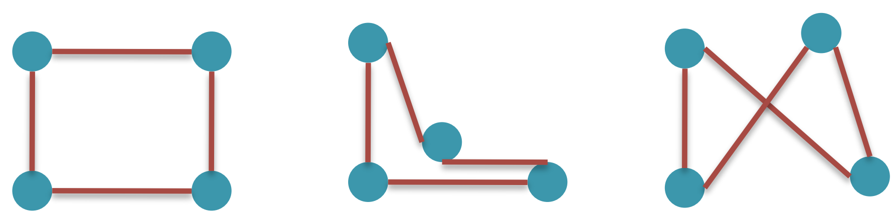
  $e_G = |E|$

- For an edge $e = uv$, the vertices u and v are the **Ends** of this edge; u and v are **neighbors**.

Graph concepts

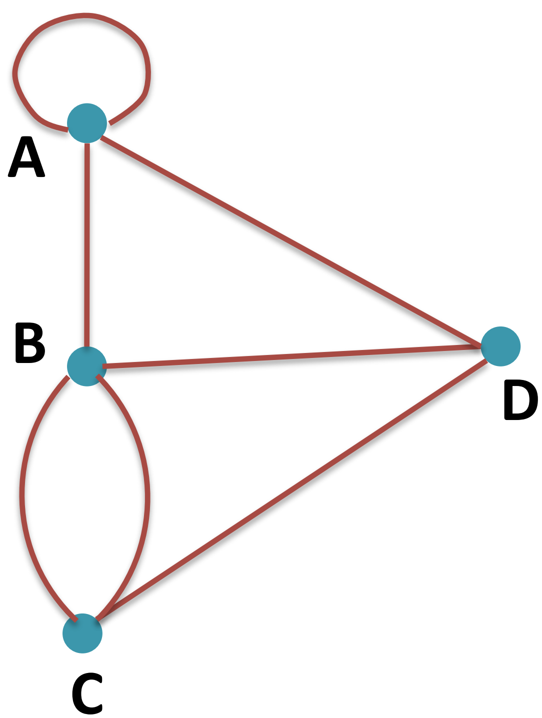


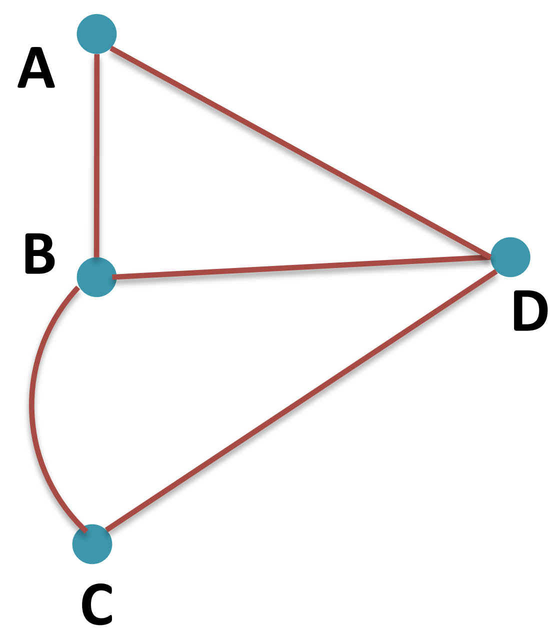
- Edges between B and C- multiple edges
- AA – loop
- The shape of the graph does not matter; only the way the nodes are connected to each other.



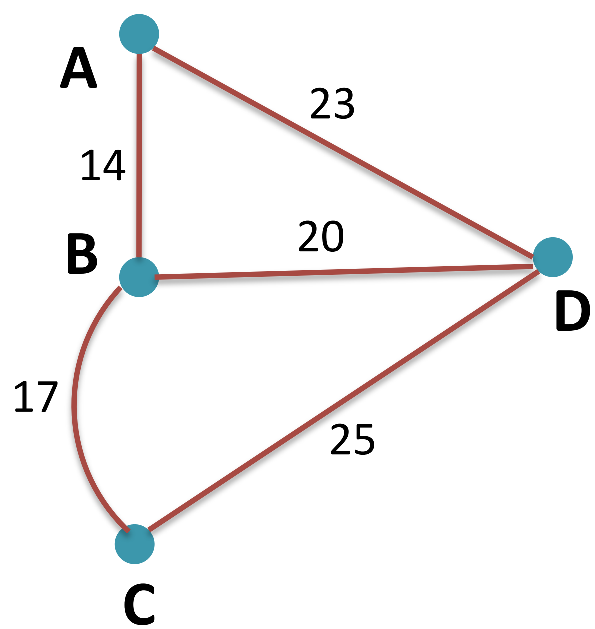
Simple graph

- A simple graph does not have loops (self edges) and multiple identical edges.



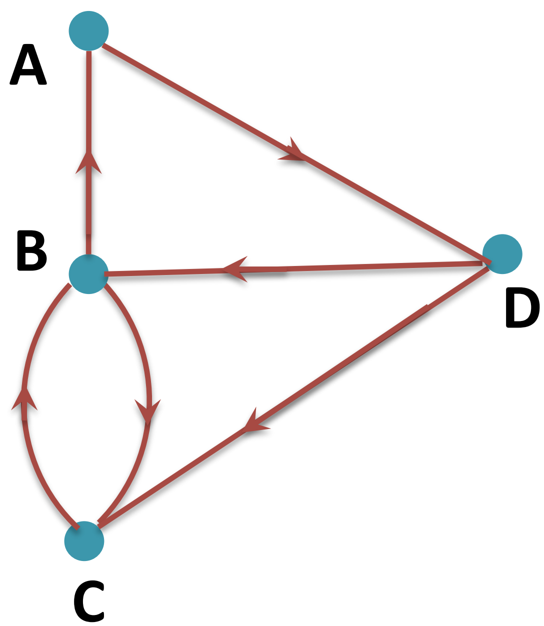


Weighted graph



- A edge has a weight value.
- In some applications, the weights, e.g., correspond to travel costs or geographical distances.

Directed Graph (Digraph)

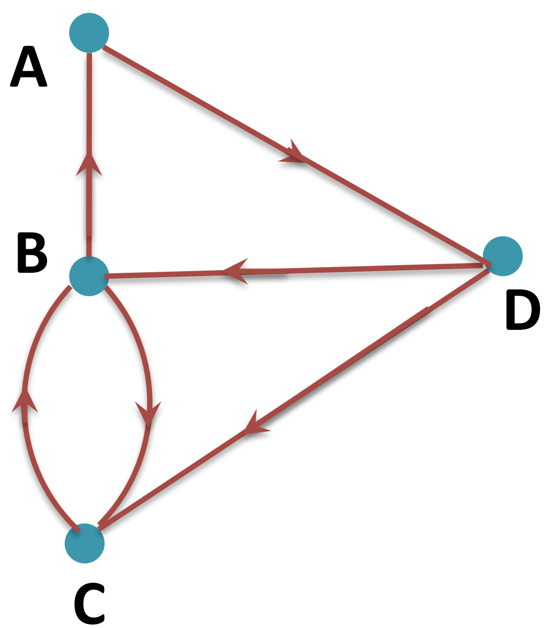


- Edges have directions, where the edges are drawn as arrows.
- The edges in the digraph are also called “arcs”.
- A digraph can contain edges BC and CB of opposite directions.

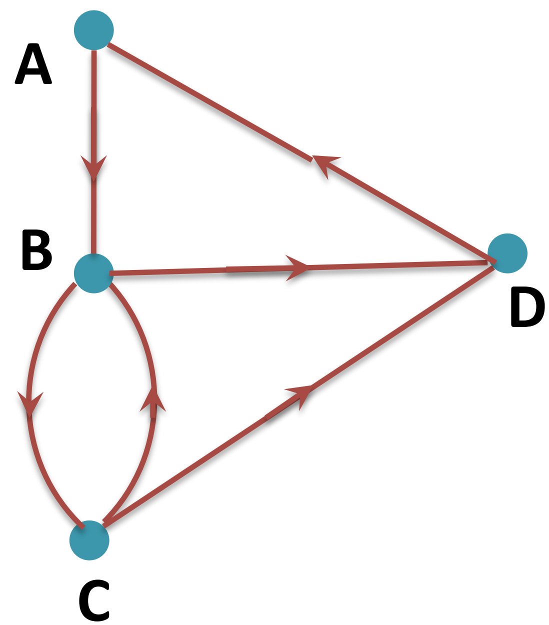
Question: Is this a simple graph?

Opposite of a Digraph

G(E,V)



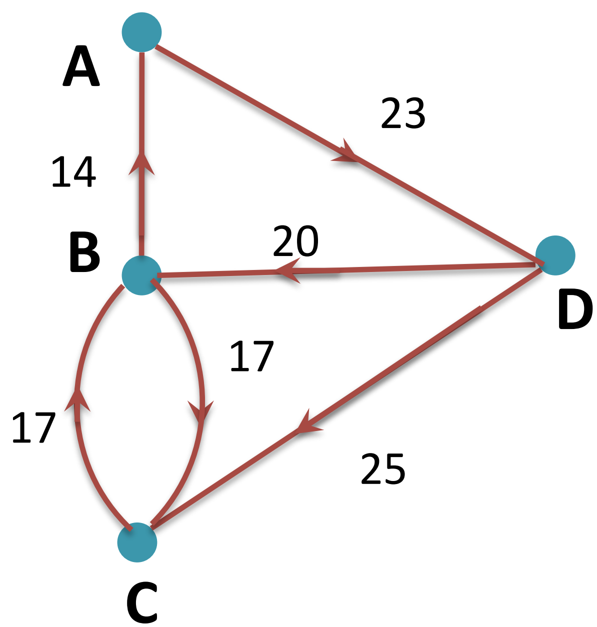
G^{op}(E,V)



All vertices are same, but the arrows reversed.

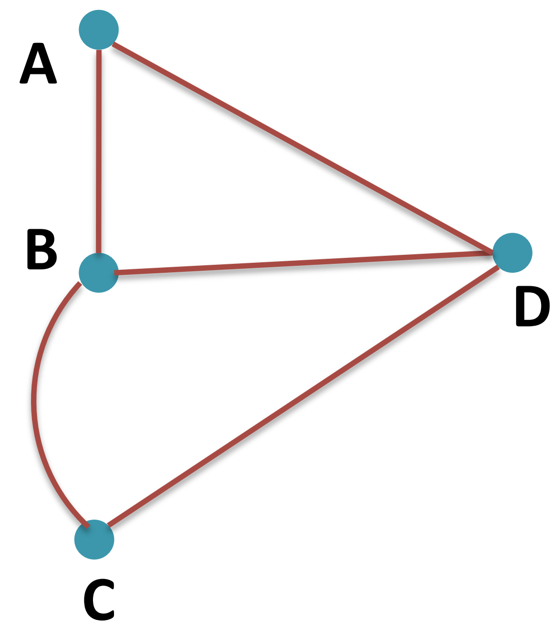
Weighted Digraph

- Edges have both weights and directions.
-



Representations of a graph

- Plane figures.
- List of edges.
- Adjacency matrix.



Representations: List of edges

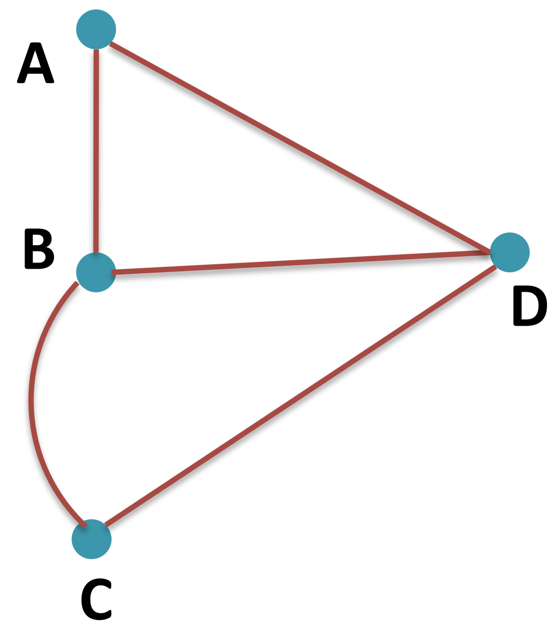
A – B

A – D

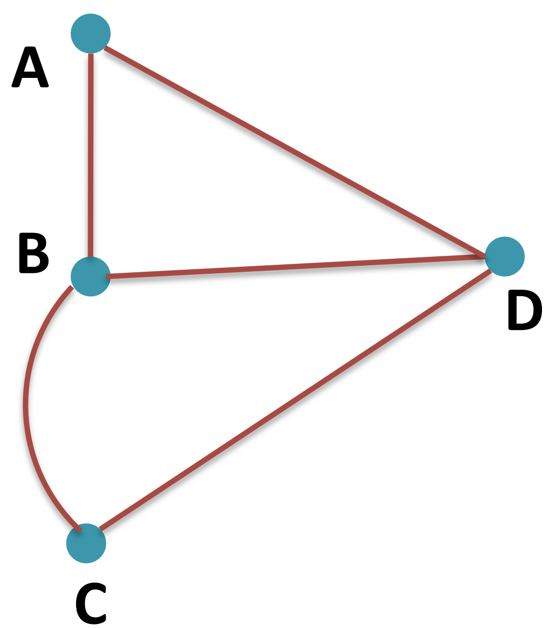
B – C

B – D

C – D



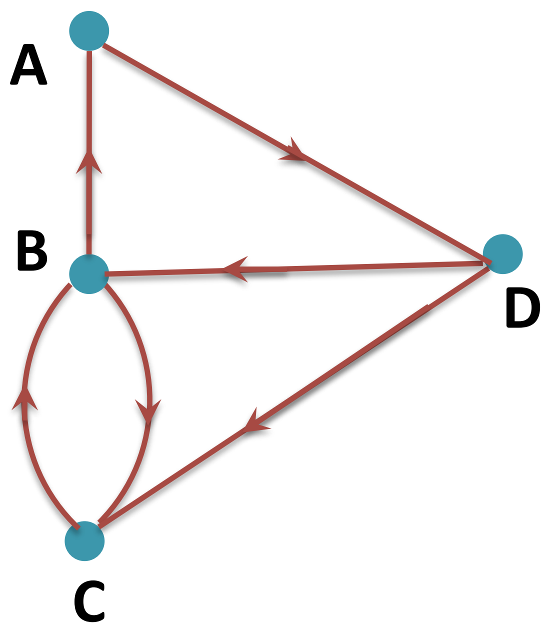
Representations: adjacency matrix



	A	B	C	D
A		1		1
B	1		1	1
C		1		1
D	1	1	1	

(1) Symmetric matrix.
(2) What does it mean if there is a number for a diagonal entry?

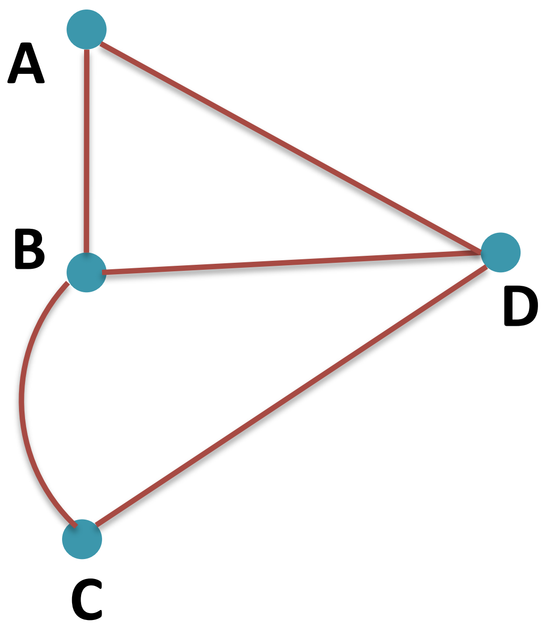
Representations: adjacency matrix for a digraph



	A	B	C	D
A				A->D 1
B	B->A 1		B->C 1	
C		C->B 1		
D		D->B 1	D->C 1	

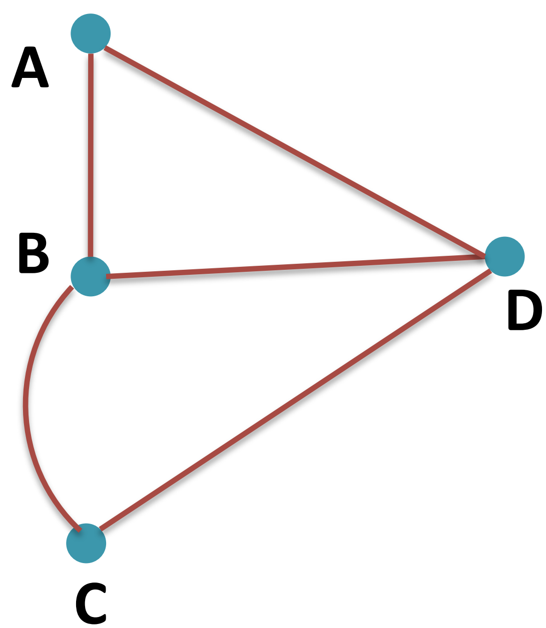
This matrix is not necessary to be symmetric.

Node degrees



- Neighborhood: all neighbors of a node.
- Degree: the number of edges connected to the nodes; the number of neighbors of a node (vertex).
- Maximum degree and minimum degree. In a graph, the largest degree and the smallest degree.

Degrees in the adjacency matrix

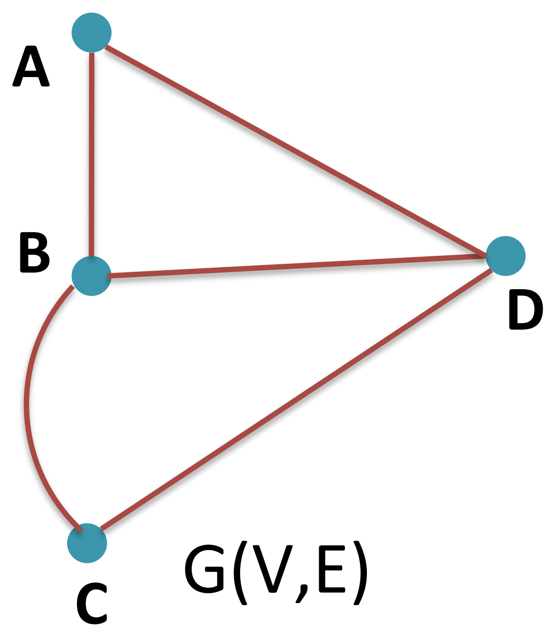


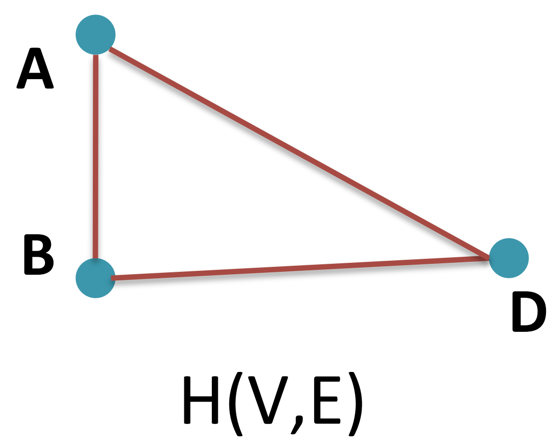
	A	B	C	D
A		1		1
B	1		1	1
C		1		1
D	1	1	1	

Degrees: 2 3 2 3

Subgraph

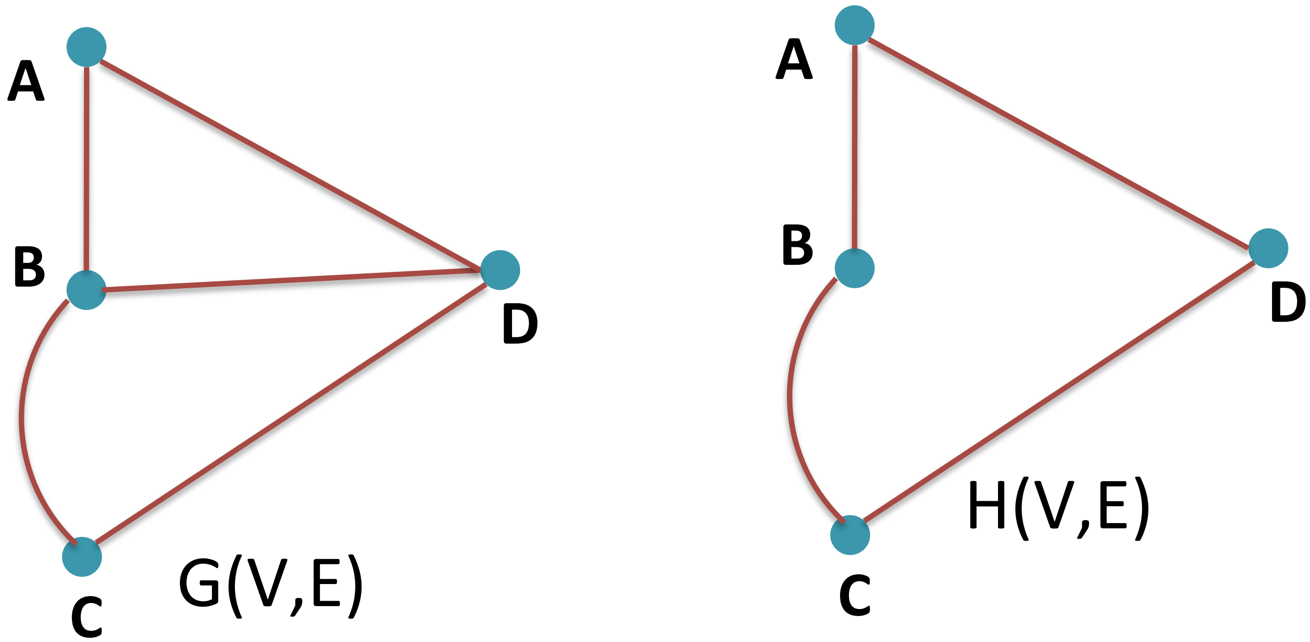
- If *H* is a subgraph of *G*, $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.





Spanning Subgraph

- If *H* is a spanning subgraph of *G*, $V(H)=V(G)$ and $E(H) \subseteq E(G)$.

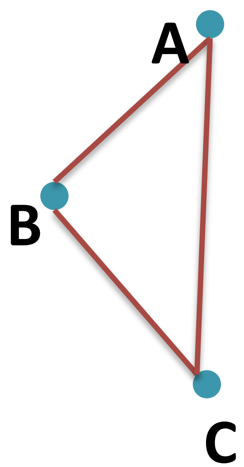


Special graphs

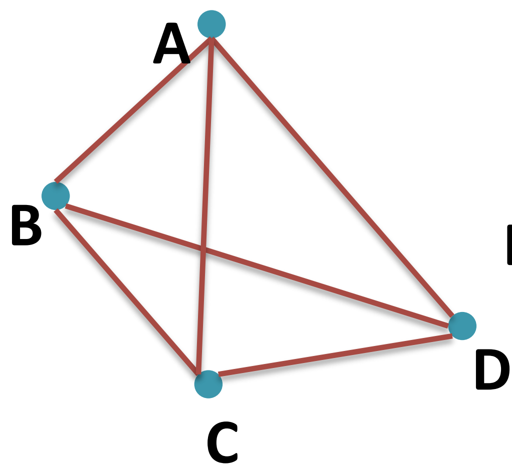
- Trivial graph
- Complete graph
- Connected and disconnected graph
- Trees
- Bipartite graph
- Regular graph
- Line graph

Trivial graph

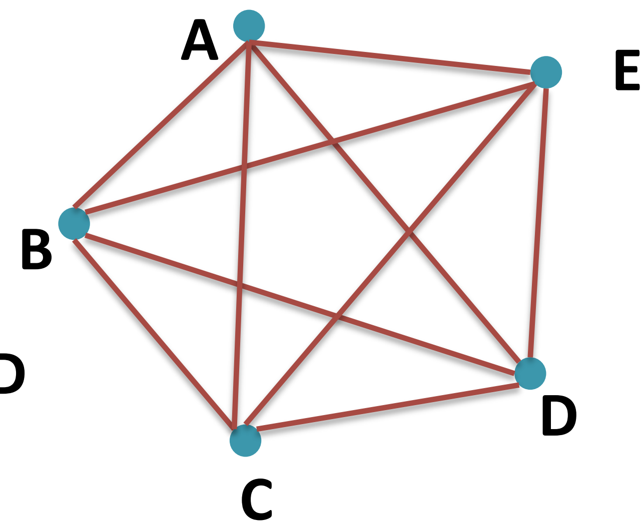
Complete graph (clique)



K_3



K_4



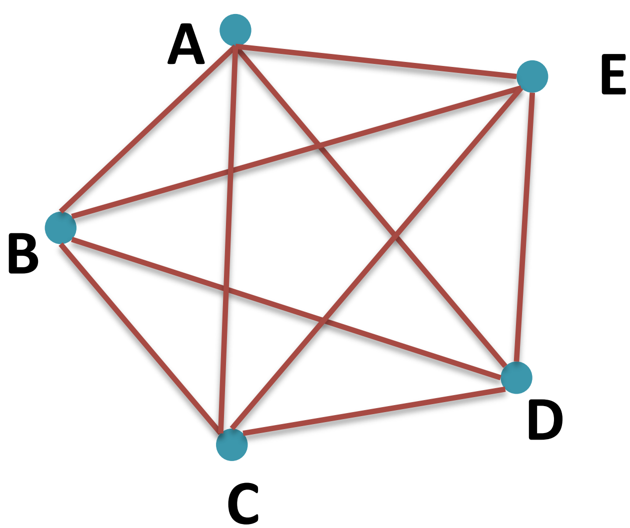
K_5

What the total number of edges in a complete graph?

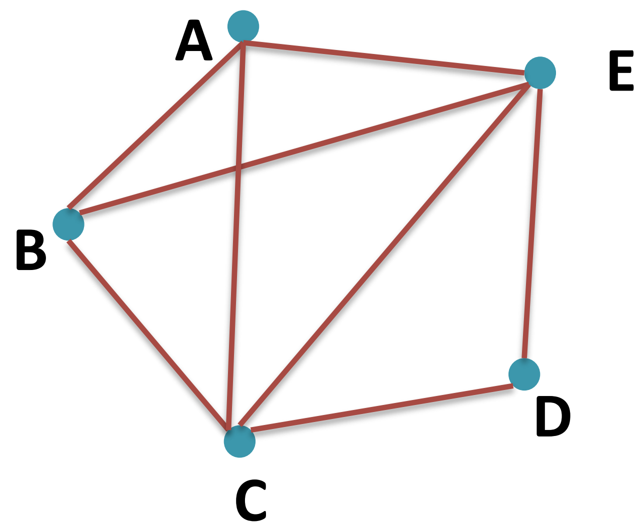
$$|E| = \binom{n}{2} = \frac{n(n-1)}{2}$$

Connection Density

$$Q = \frac{\#\text{of Connections}}{\text{Max.}\#\text{ of possible Connections}} = \frac{E}{V(V-1)/2}$$



Q=1

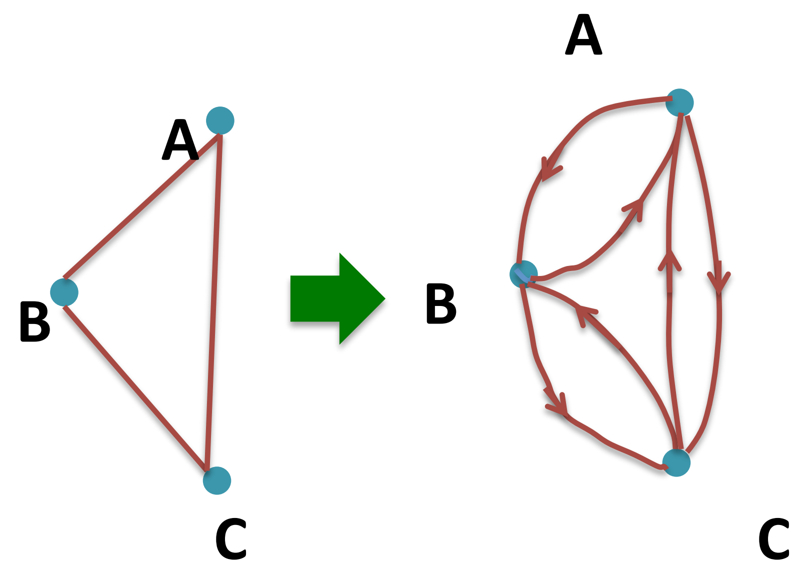


Q=8/10

Complete digraph

What is the largest number of arcs that a simple digraph with *N* nodes can have?

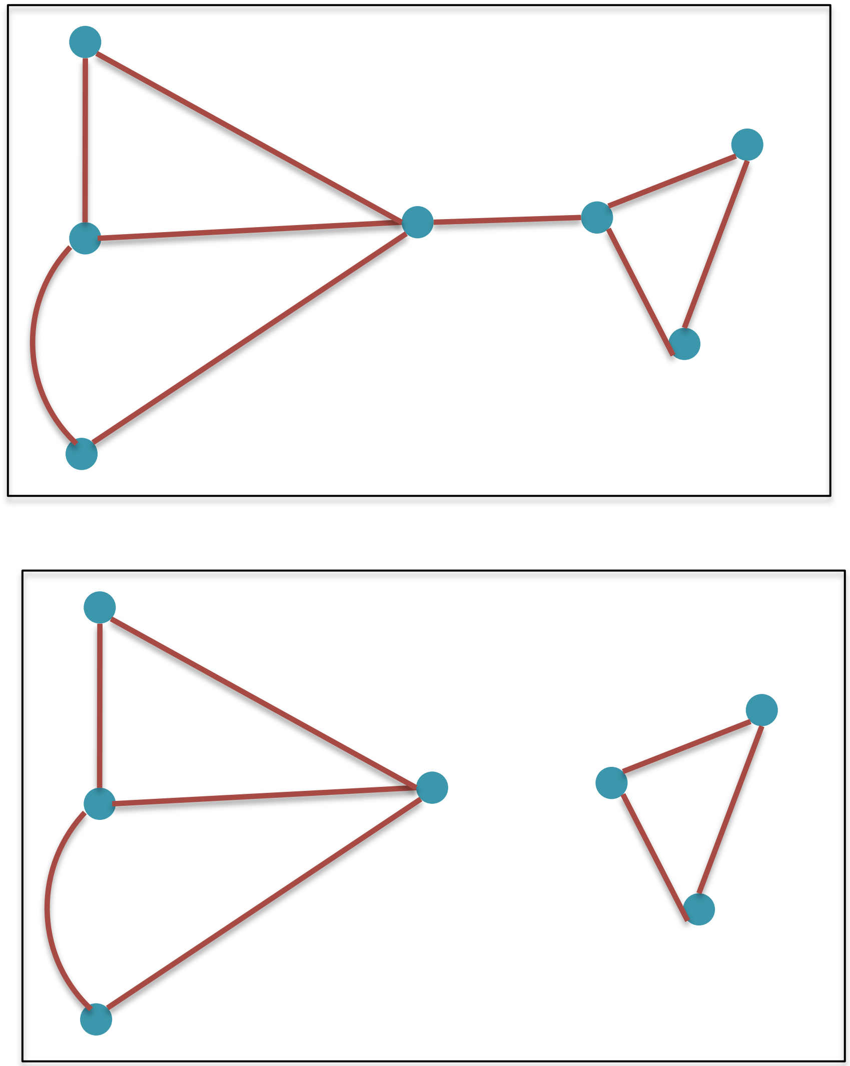
$$|\boldsymbol{E}| = 2 \times \binom{n}{2} = 2 \times \frac{\boldsymbol{n}(\boldsymbol{n}-1)}{2}$$



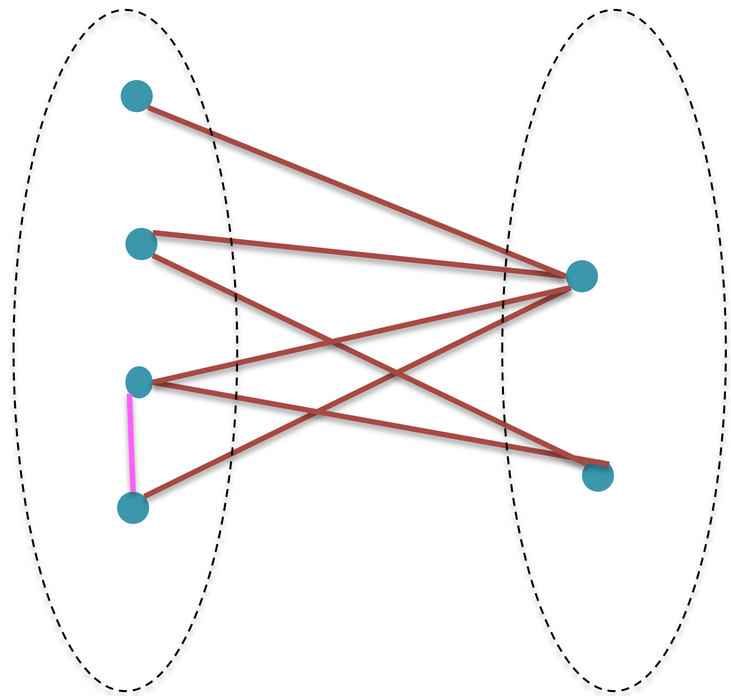
Connected and disconnected

G1

G2



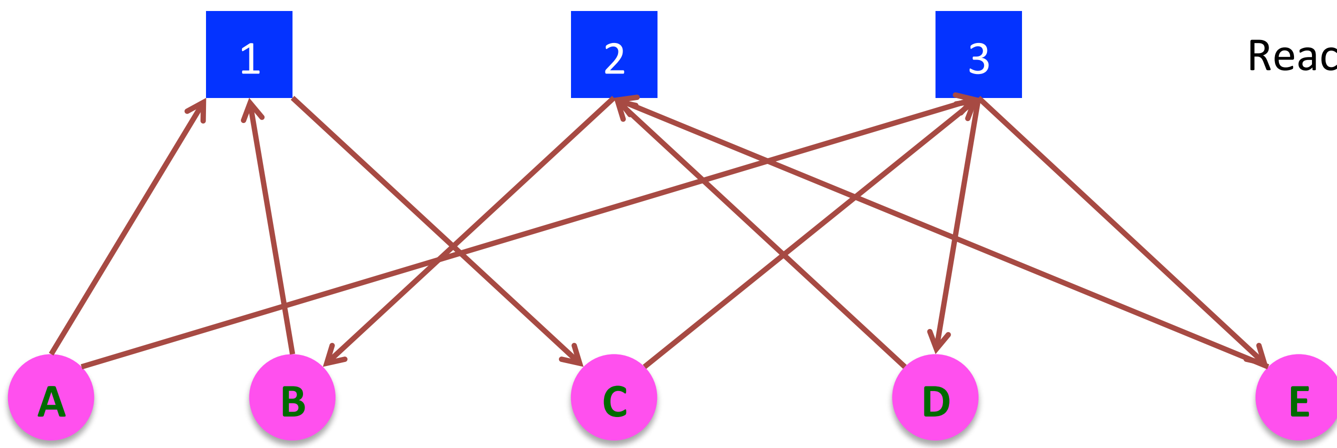
Bipartite graph



Metabolic interaction networks

Bipartite graph

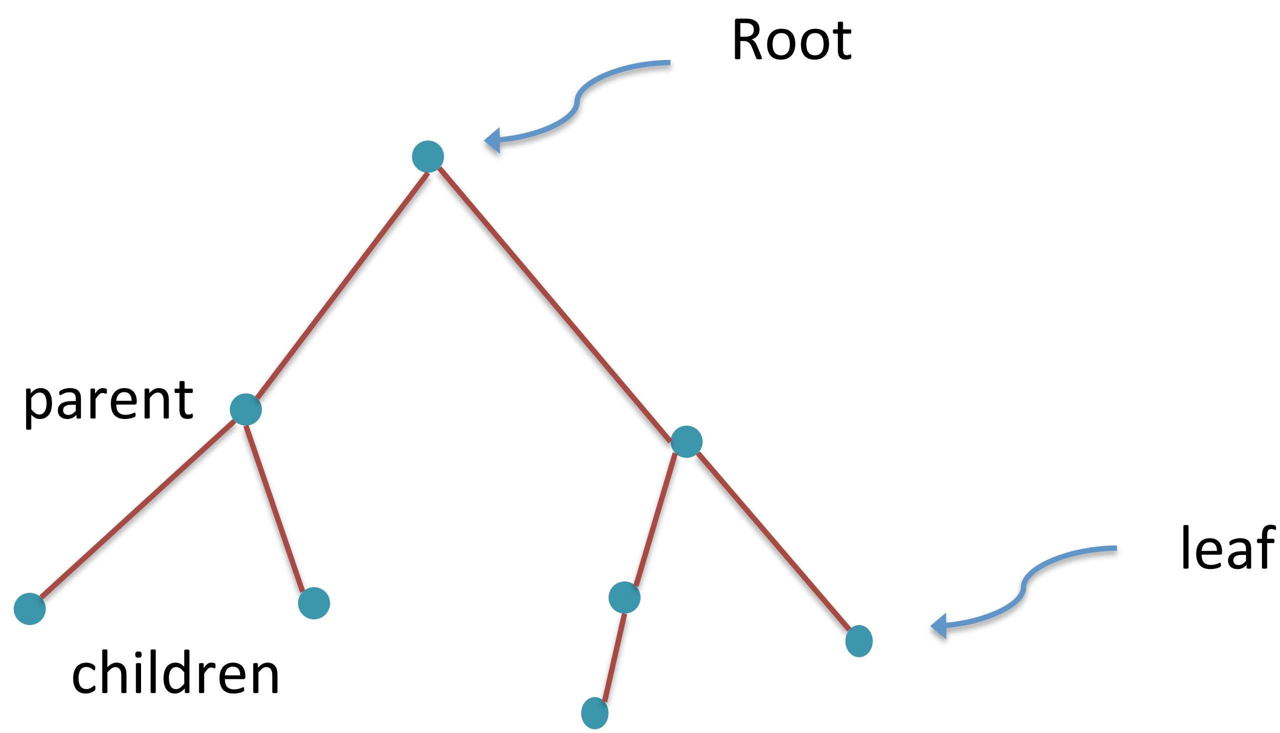
Reactions



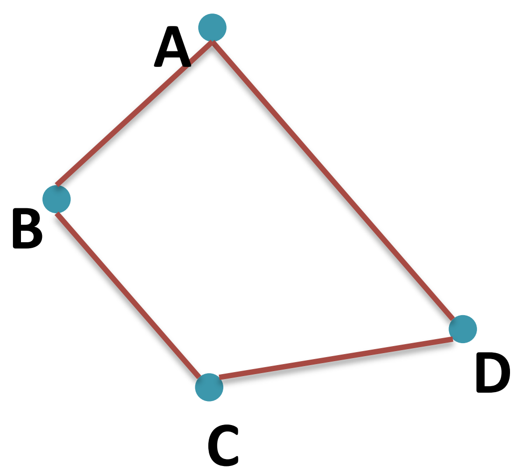
Substrates

1	A + B → C
2	D + E → B
3	A + C → D + E

Trees



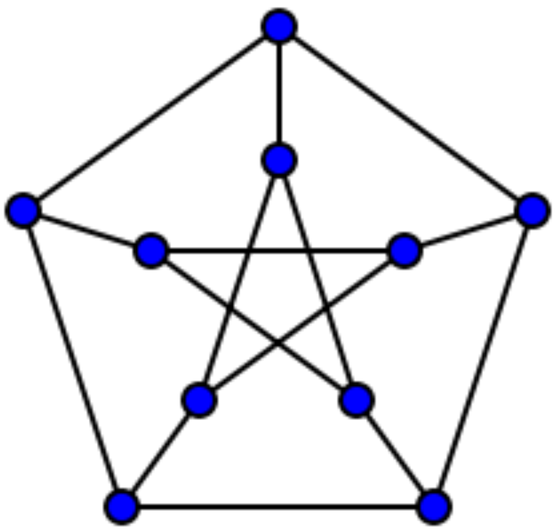
Regular graph



- A graph *G* is said to be regular, if every vertex of *G* has the same degree.
- If the degree is *r*, then *G* is *r*-regular.

Is the complete graph a regular graph?

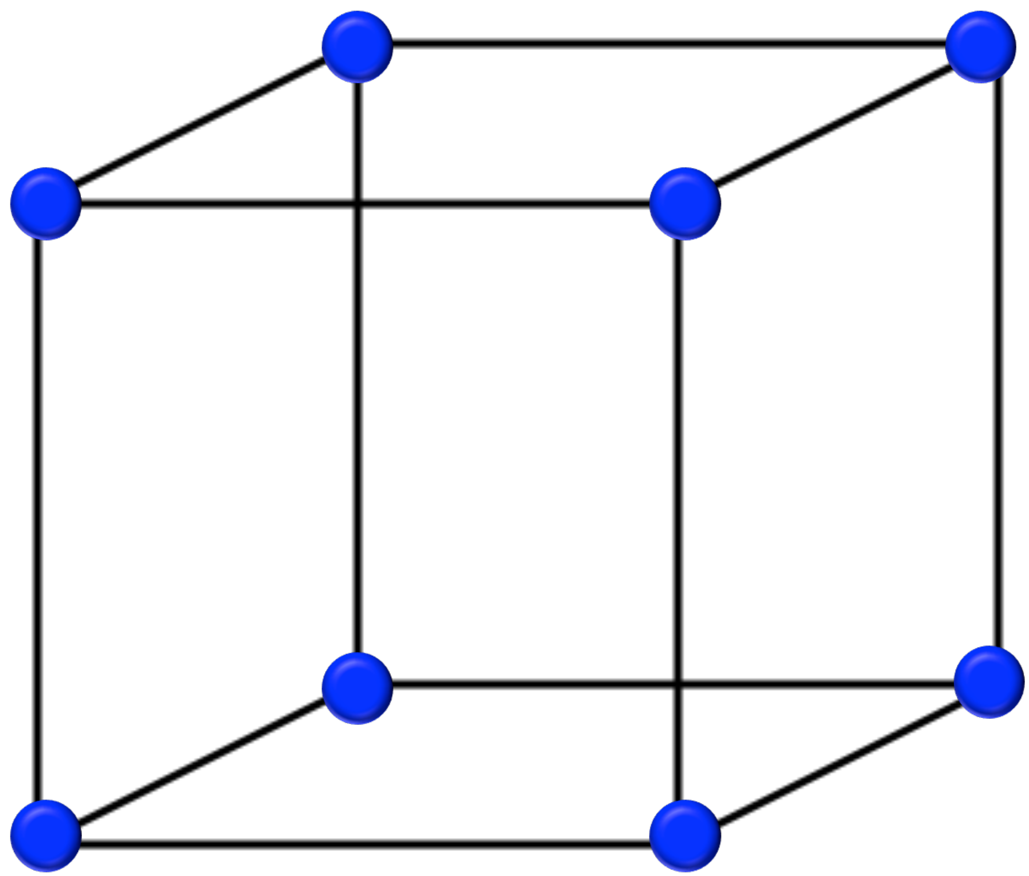
Regular graph: Petersen Graph



- Order = 10.
- Size = 15.
- 3-regular graph

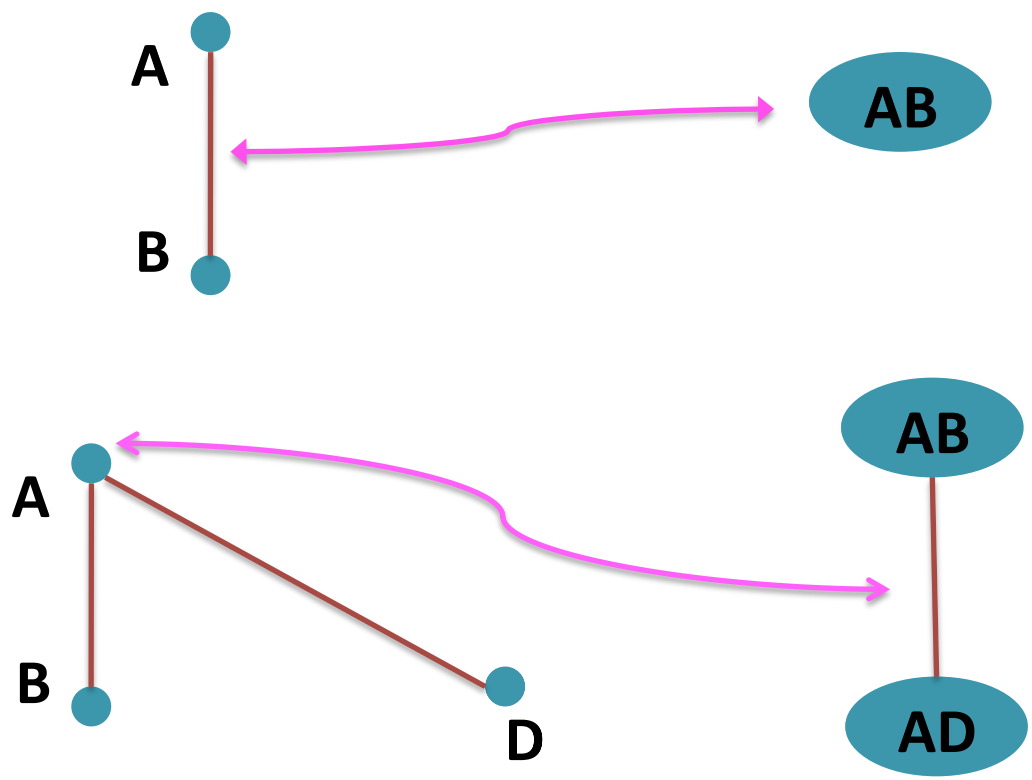
Regular graph

Can you find a 3-regular graph whose order is 8?



Line graph

G(V,E) → G'(E,V)



Graph Theory

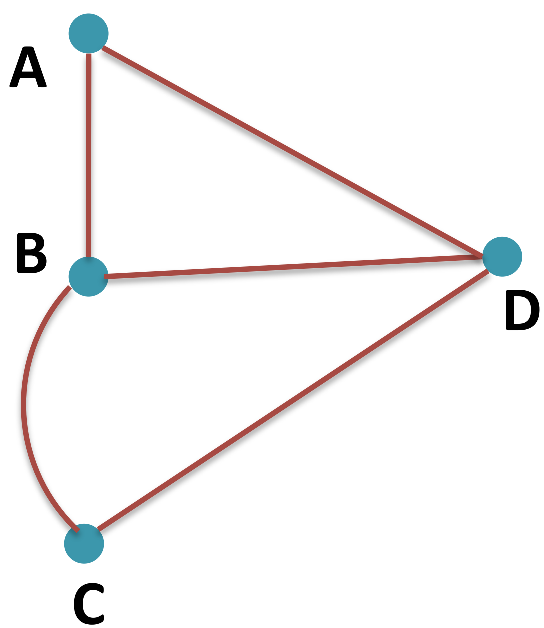
- Some basic concepts of Graph theory
- Some examples of Special graphs
- Graph paths and cycles
- Graph connectivity
- Tree and Bipartite graph
- Network motifs

Subgraphs again, special

- Walks
- Paths
- Circuits
- Cycles

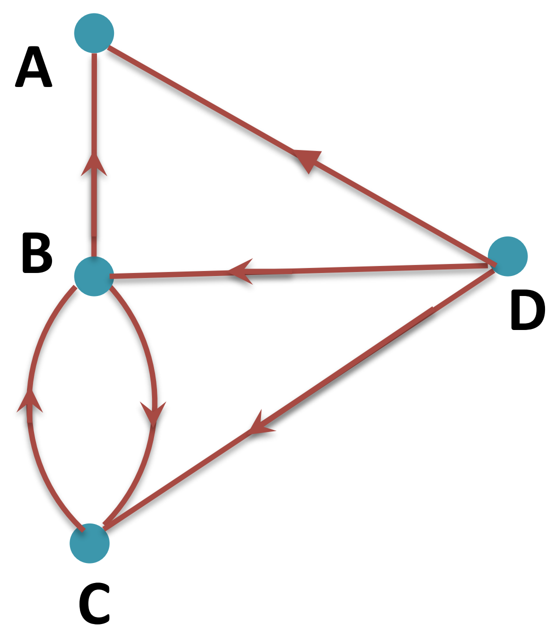
Walks

- Walk: a sequence of nodes in which each node is adjacent to the next one.
- In the digraph, a walk needs to follow the direction of edges.



A-B-D-C

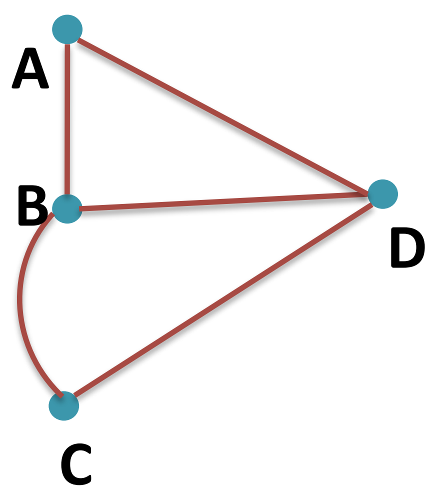
A-B-D-B-C-D



D→ C→ B→A

Paths

- Path: a sequence of nodes in which each node is adjacent to the next one, and edges can be part of a path only once.

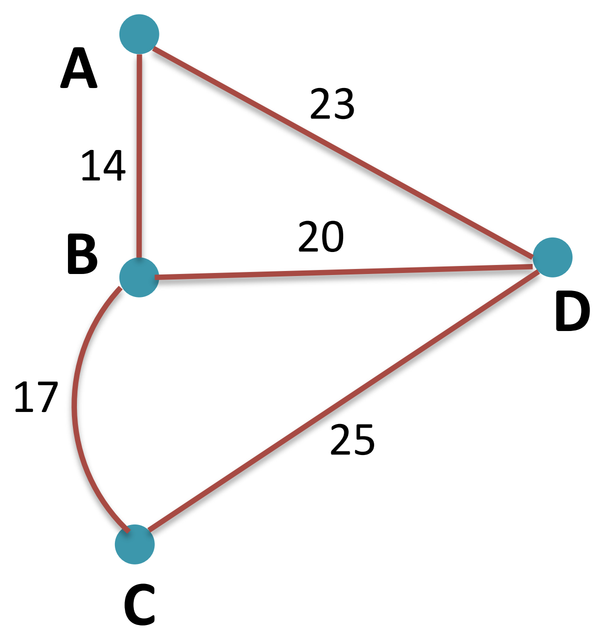


A-B-D-C

~~A-B-D-B-C-D~~

A path having k vertices, is denoted by P_k . The length of this path is k-1.

Paths in Weighted graph

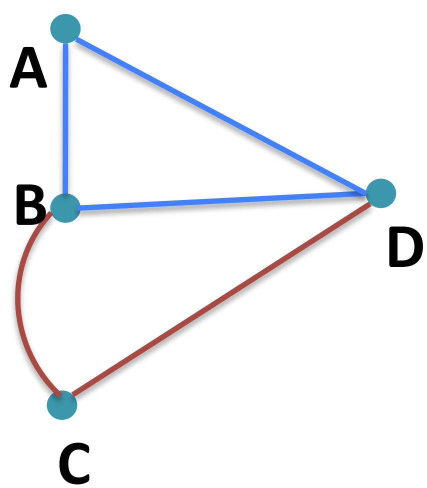


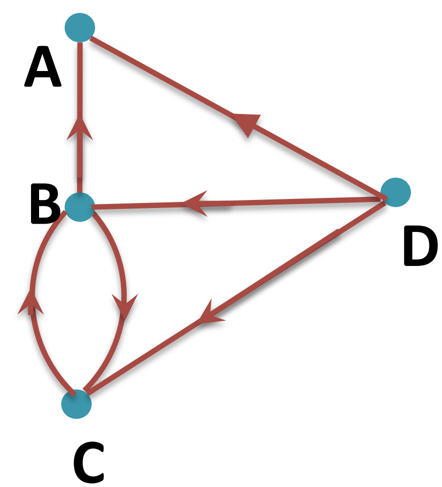
- The length of a path is the sum of all edge weights in the path.

Length of (A –B–C –D) = 14+17+25

Circuits

- Circuit: a walk that starts and ends at the same vertex.



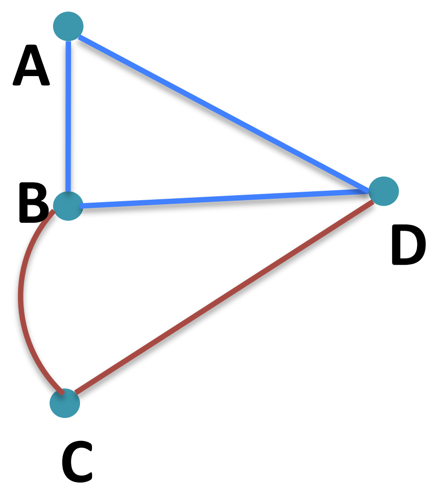


A - B – D - A
A - B - D - C - B - A

C→ B→C

Cycles

- Cycle: a circuit that does not revisit any nodes.

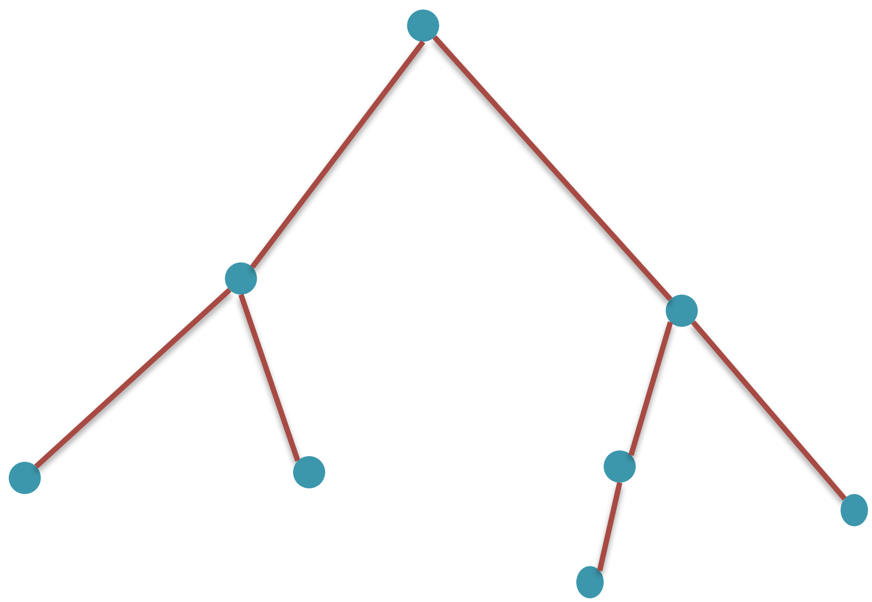


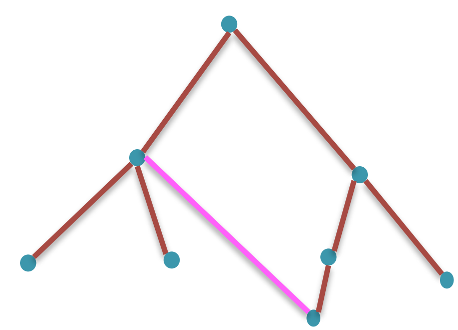
A cycle having *k* vertices, is denoted by C_k . The length of this cycle is also *k*.

A - B – D - A

~~A - B - D - C - B - A~~

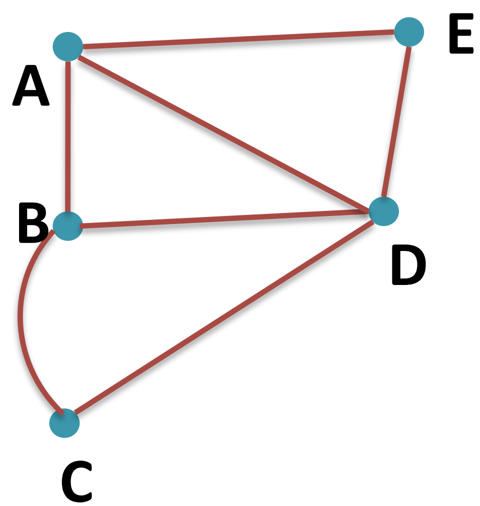
A tree does not have any cycle

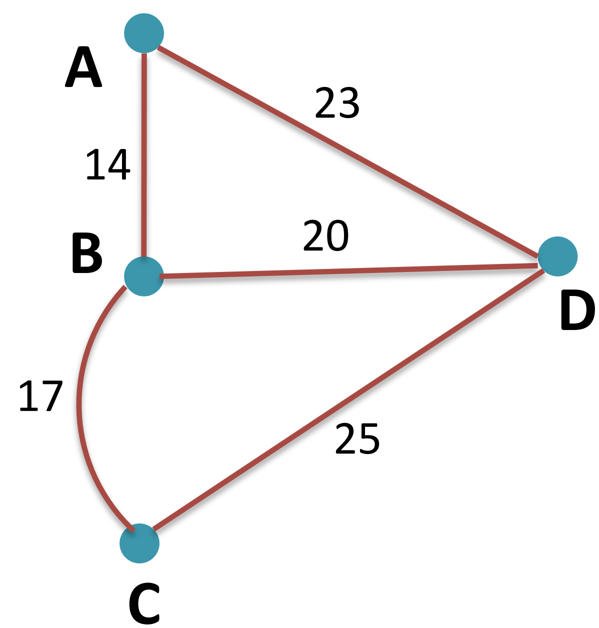




Shortest path

Between two nodes, there are multiple paths. The path having the shortest length is called the shortest path.





A – B – C

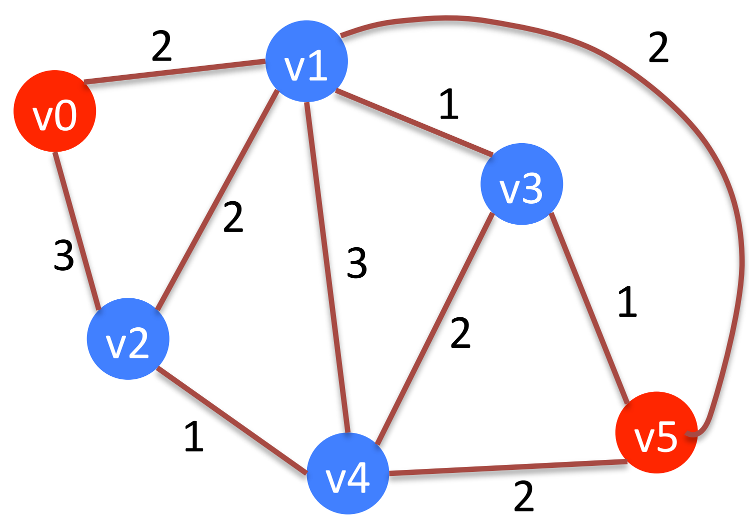
A – E – D – C

A – B – C : 14+7

A – D – C : 23+25

Distances between nodes

- The distance between two nodes is defined as the length of the shortest path.
- If the two nodes are disconnected, the distance is infinity.

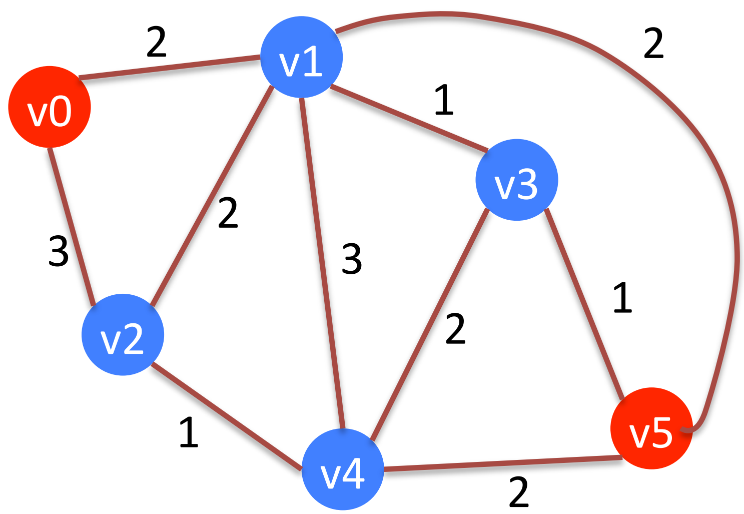


Distances between nodes

- In digraphs, path needs to follow the direction of the arrows.
- Thus in a digraph the distance from node A to B (on an AB path) is generally different from the distance from node B to A (on a BA path).

Diameter

- Graph diameter: the maximum distance between any pair of nodes in the graph.
- Note: not the longest path.



Average distance

- Average distance for a connected graph (component).

$$\langle l \rangle \equiv \frac{1}{2N_{pairs}} \sum_{i,j \neq i} l_{ij}$$

, where l_{ij} is the distance from node i to node j, $N_{pairs} = \binom{N}{2} = \frac{N(N-1)}{2}$ and N is the number of nodes in the graph or component.

Since in a (symmetrical) graph $l_{ij} = l_{ji}$, we only need to count them once

$$\langle l \rangle \equiv \frac{1}{N_{pairs}} \sum_{i,j>i} l_{ij}$$

Betweenness centrality (load)

- For all node pairs (*i, j):*
  - Find all the shortest paths between nodes *i* and *j* - $C(i,j)$
  - Determine how many of these pass through node k - $C_k(i,j)$
- The betweenness centrality of node k is

$$g_k = \sum_{i \neq j} \frac{C_k(i,j)}{C(i,j)}$$

Graph efficiency

- To avoid infinity distance in graphs that are not connected and digraphs that are not strongly connected, one can define a graph efficiency (= average inverse distance)

$$\langle l \rangle = \frac{1}{2N_{\text{paris}}} \sum_{i,j \neq i} \frac{1}{l_{ij}}$$

N_pairs is the number of node pairs

A graph has a small average distance, it has a large graph efficiency.