INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

Andrea Scozzari

University Niccolò Cusano
Dept. of Economics, Psychological and Communication Science.

andrea.scozzari@unicusano.it

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

Basic references

- Frank Harary: Graph Theory, Addison-Wesley Publishing Company, 1969, ISBN 8185015554, 9788185015552.

- Nicos Christofides: Graph Theory: An algorithmic approach Academic Press, London, 1975, ISBM =-12-174350-0.

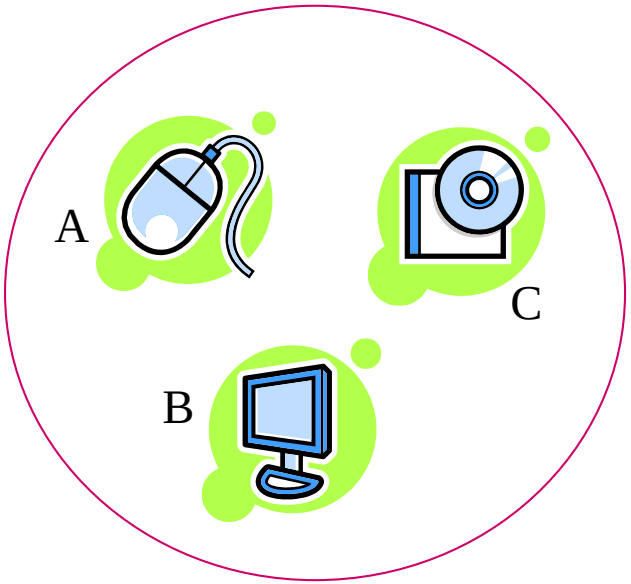
- Ahuja, Ravindra K., Thomas L. Magnanti, and James B. Orlin: Network flows: Theory, Algorithms and Applications, Pretience Hall, Upper Saddle River, New Jersey, 1988, ISBN 0-13-617459-X.

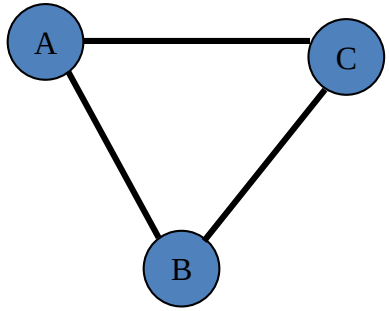
- Andreas Brandstadt, Van Bang Le, Jeremy P. Spinrad: Graph Classes. a Survey, SIAM, Philadelphia, 1999, ISBN:9780898719796, 0898719798

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

Many problems can conveniently be represented by means of graphs or networks.

Graphs make it possible to represent the relationships between the elements of a set (individuals, objects, etc.), i.e. the connections existing between these elements:

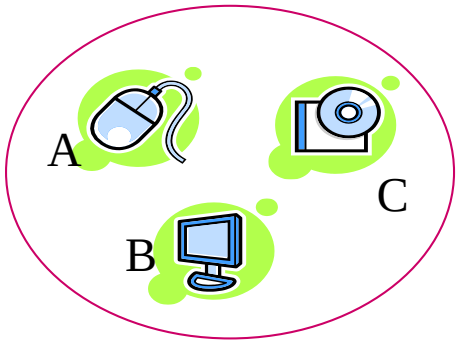


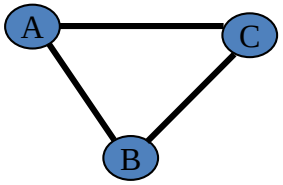


INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

A graph consists of a set N called the set of nodes or vertices of the graph that represent the objects of the set, and the set E of edges of the graph that represent the relationships (connections) existing between the nodes

G=(N,E)





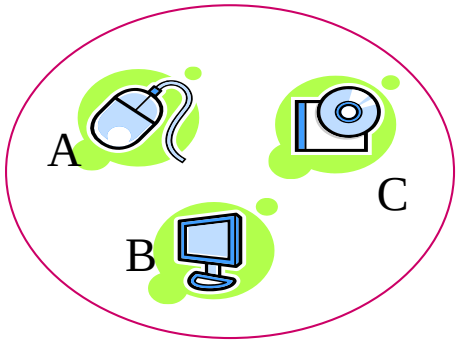
N = {A,B,C} is the set of nodes

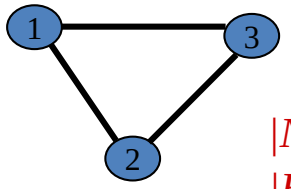
E = {(A,B), (B,C), (A,C)} is the set of edges

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

A graph consists of a set N called the set of nodes or vertices of the graph that represent the objects of the set, and the set E of edges of the graph that represent the relationships (connections) existing between the nodes

G=(N,E)





$|N| = n = 3$

$|E| = m = 3$

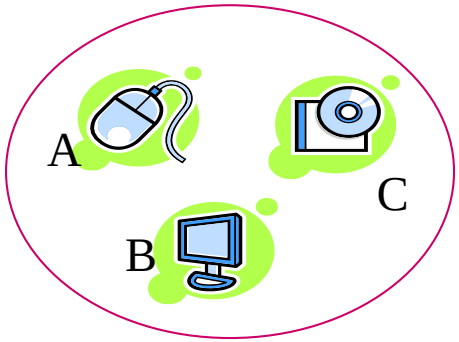
$N = \{1,2,3\}$

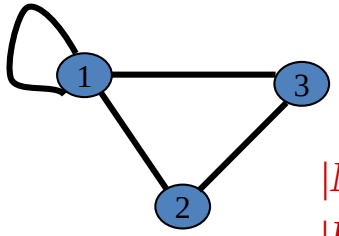
$E = \{(1,2), (2,3), (1,3)\}$

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

The ends of an edge are said to be incident with the edge, and vice versa. Two vertices which are incident with a common edge are adjacent, as are two edges which are incident with a common vertex. An edge with identical ends is called a loop.

G=(N,E)





$|N| = n = 3$

$|E| = m = 4$

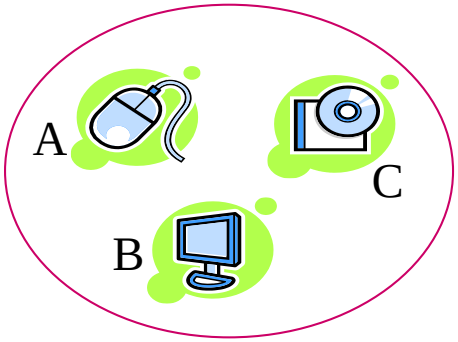
$N = \{1,2,3\}$

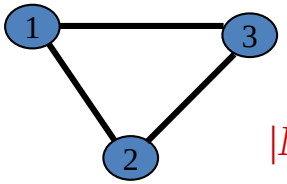
$E = \{(1,1), (1,2), (2,3), (1,3)\}$

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

If the pairs (i,j), (j,i) represent the same relationship, then the graph is said to be undirected. If the relationship between two nodes only occurs in one 'direction', i.e. $(i,j) \neq (j,i)$, the graph is said to be oriented or directed.

G=(N,E)

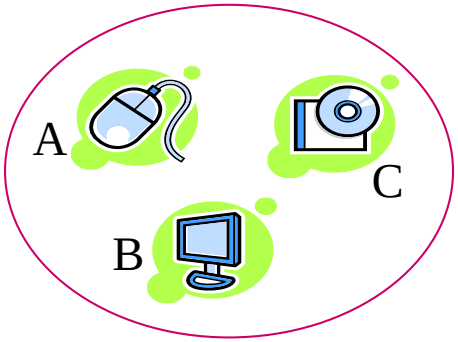


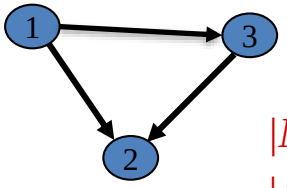


$|N| = n = 3$

$|E| = m = 3$

G=(N,A)





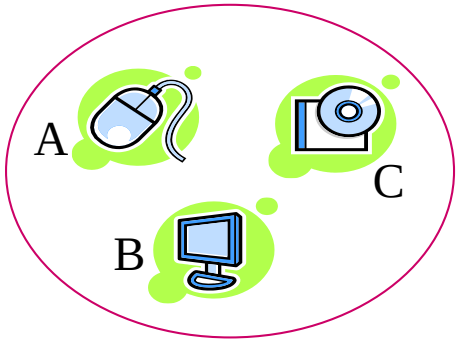
$|N| = n = 3$

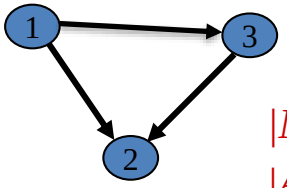
$|A| = m = 3$

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

In an directed graph G, we denote by N the set of nodes of the graph and by A the set of ordered pairs of nodes, also called the set of arcs of the graph.

G=(N,A)





$|N| = n = 3$

$|A| = m = 3$

$N = \{1,2,3\}$

$A = \{(1,2), (1,3), (3,2)\}$

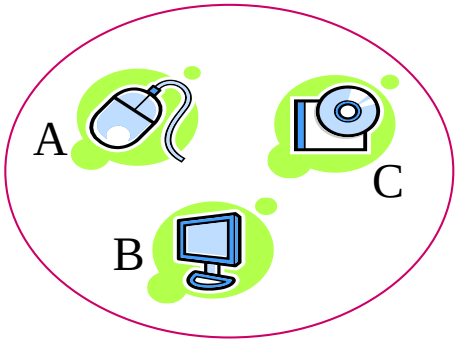
INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

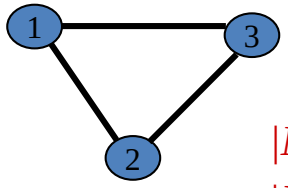
An undirected graph is finite if both its vertex set and edge set are finite. We consider only finite graphs, and so the term 'graph' always means 'finite graph'.

We call a graph with just one vertex trivial and all other graphs nontrivial.

A graph is simple if it has no loops and no two of its edges join the same pair of vertices.

G=(N,E)



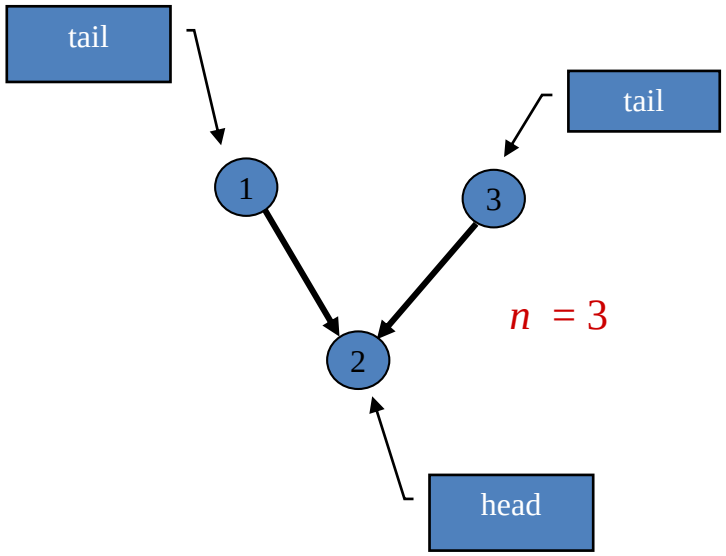


$|N| = n = 3$

$|E| = m = 3$

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

In a directed graph G, given an arc (i,j), node i is called the tail of the arc, while node j is the head of the arc.



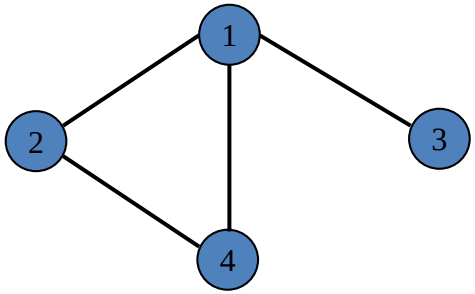
$N = \{1,2,3\}$

$A = \{(1,2), (3,2)\}$

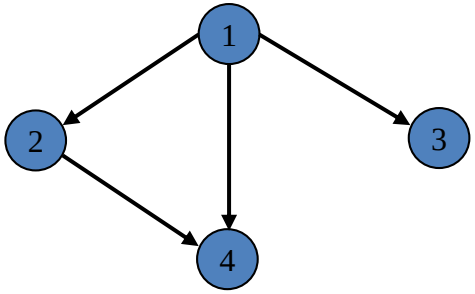
$|N| = n = 3$

$|A| = m = 2$

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

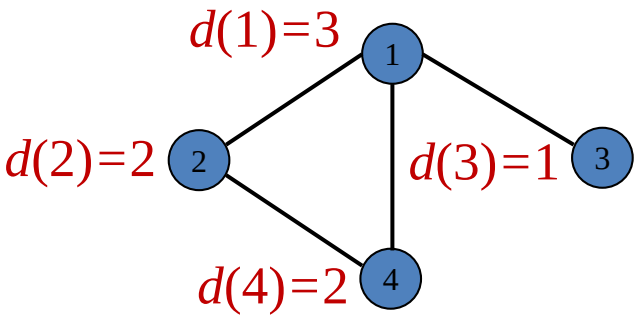


In an undirected graph, given an edge (i,j) the nodes i and j are adjacent, or the edge is said to be incident to i and j

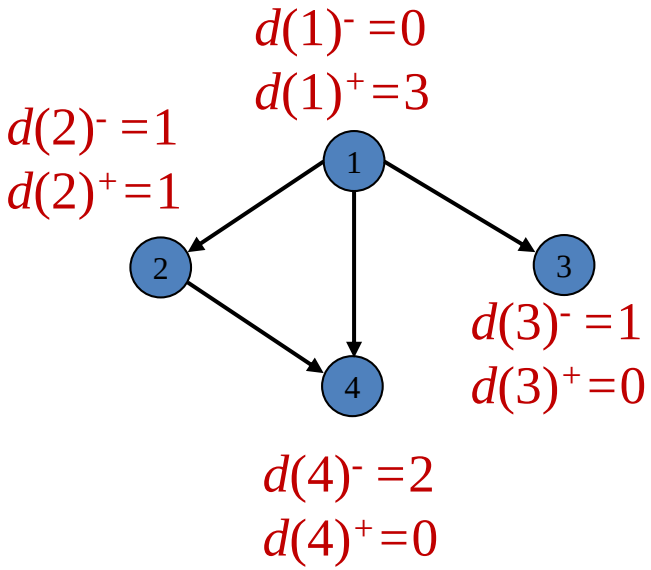


In a directed graph, given an arc (i,j) it is said that the arc is incident from i to j.

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

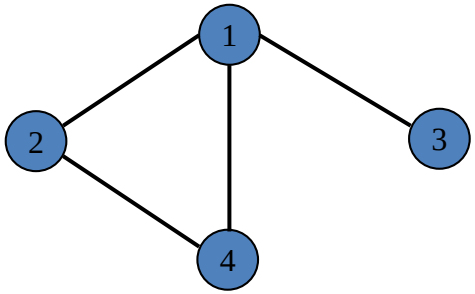


In an undirected graph, we define the degree of a node i, and denote it by d(i): the number of edges incident to i.

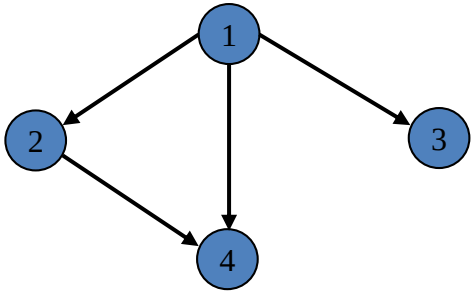


In a directed graph, we define the incoming and outgoing degree of a node i, and denote them by $d(i)^-$ and $d(i)^+$: the number of arcs entering and leaving i, respectively

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS



Proposition 1: Given an undirected graph G, the sum of the degrees of the nodes of G is equal to twice the number of edges.



Proposition 2: Given a directed graph G, the sum of the incoming degrees of all nodes is equal to the sum of the outgoing degrees of all nodes, which is equal to |A|.

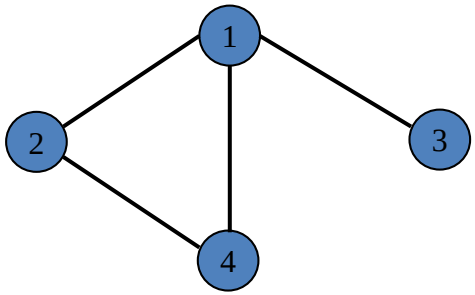
INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

To any undirected graph G there corresponds a $n \times n$ matrix called the adjacent matrix denoted by

$$A(G) = [a_{ij}]$$

where the element a_{ij} is:

1 if edge (i,j) exists in G
0 if edge (i,j) does not exist in G



$$A(G)=\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

- Square matrix
- Zero diagonal matrix
- Symmetric

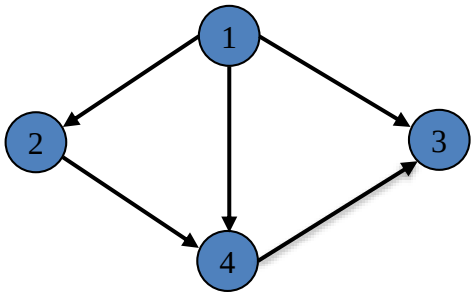
INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

To any directed graph G corresponds a $n \times m$ matrix called the incidence matrix denoted by

$$B(G) = [b_{ij}]$$

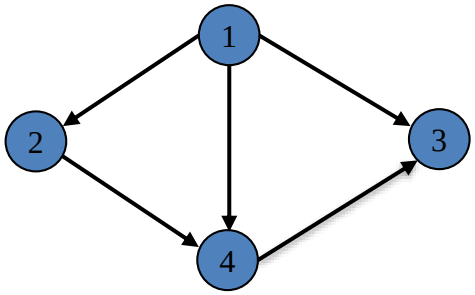
where the element b_{ij} is:

-1 if i is the tail of an arc (i,j) in G
1 if j is the head of an arc (i,j) in G



$$B(G)=\begin{pmatrix} -1 & -1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 \end{pmatrix}$$

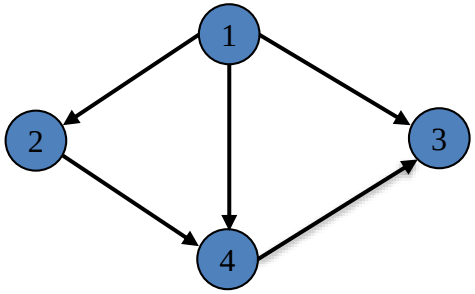
INTRODUCTION TO GRAPH THEORY AND APPLICATIONS



$$B(G)=\begin{pmatrix} -1 & -1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 \end{pmatrix}$$

Theorem. Let $G(N, A)$ be a directed (and connected) graph and let B(G) its incidence matrix. The rank of B(G) is $n - 1$.

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS



$$B(G)=\begin{pmatrix} -1 & -1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & -1 \end{pmatrix}$$

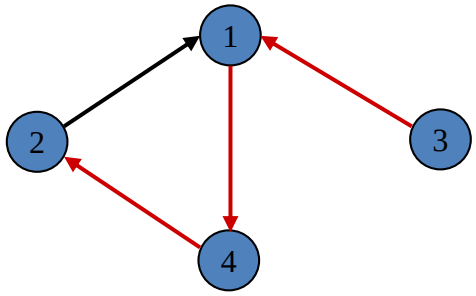
Definition 1: A matrix A is said to be Totally Unimodular (TU) if, for each square submatrix C of A, $\det(C) \in \{0, +1, -1\}$.

Theorem: The incidence Matrix of a directed graph G is Totally Unimodular (TU)

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

Given a directed graph G, consider two vertices i_0 and i_q, we define a path P between the two nodes i_0 and i_q, a (directed) sequence of nodes and arcs:

$$P = \{i_0, (i_0,i_1), i_1, (i_1,i_2), i_2,\ldots, i_{q-1}, (i_{q-1},i_q), i_q\}$$



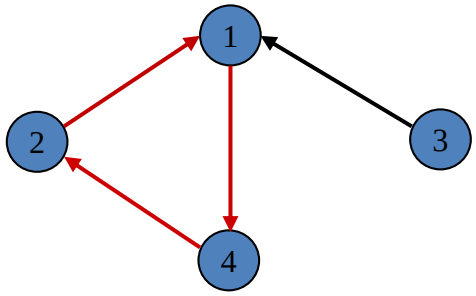
$$P = \{3, (3,1), 1, (1,4), 4, (4,2), 2\}$$

$P = \{(3,1), (1,4), (4,2)\}$ set of arcs in the path

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

Given a directed graph G, and given two nodes i_0 and i_q, with $i_0 = i_q$, a circuit C is defined as a (directed) sequence of nodes and arcs:

$$C = \{i_0, (i_0,i_1), i_1, (i_1,i_2), i_2,\ldots, i_{q-1}, (i_{q-1},i_0), i_0\}$$



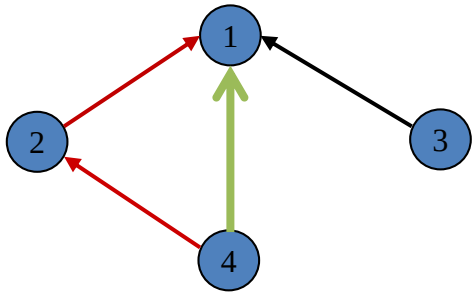
$$C = \{2, (2,1), 1, (1,4), 4, (4,2), 2\}$$

$C = \{(2,1), (1,4), (4,2)\}$ set of arcs in the circuit

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

Given a directed graph G, and given two nodes i_0 and i_q, with $i_0 = i_q$, a chain C is defined as a sequence of nodes and arcs with the direction of its arcs disregarded:

$$C = \{i_0, (i_0,i_1), i_1, (i_1,i_2), i_2,\ldots, i_{q-1}, (i_{q-1},i_0), i_0\}$$



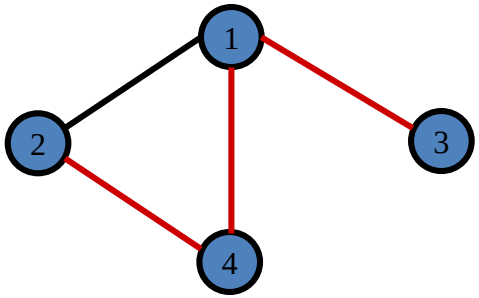
$$C = \{2, (2,1), 1, (4,1), 4, (4,2), 2\}$$

$C = \{(2,1), (4,1), (4,2)\}$ set of arcs in the chain

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

Given an undirected graph G, and given two nodes i_0 and i_q, we define a path P between the two nodes i_0 and i_q, a sequence of nodes and edges:

$$P = \{i_0, (i_0,i_1), i_1, (i_1,i_2), i_2,\ldots, i_{q-1}, (i_{q-1},i_q), i_q\}$$



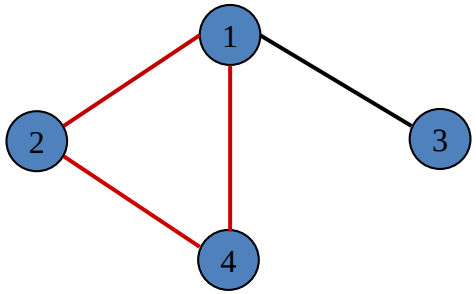
$$P = \{3, (3,1), 1, (1,4), 4, (4,2), 2\}$$

$P = \{(3,1), (1,4), (4,2)\}$ set of edges in the path

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

Given an undirected graph G, and given two nodes i_0 and i_q, with $i_0 = i_q$, a cycle C is defined as a sequence of nodes and arcs:

$$C = \{i_0, (i_0,i_1), i_1, (i_1,i_2), i_2,\ldots, i_{q-1}, (i_{q-1},i_0), i_0\}$$

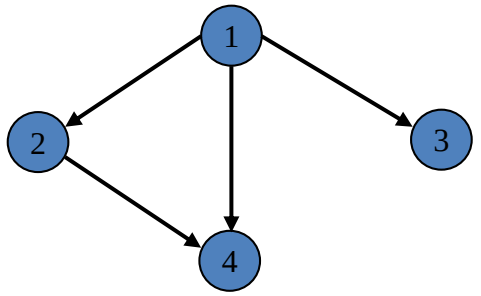


$$C = \{2, (2,1), 1, (1,4), 4, (4,2), 2\}$$

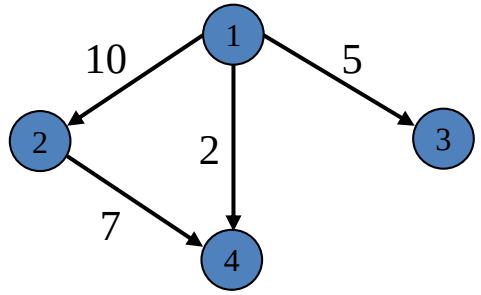
$C = \{(2,1), (1,4), (4,2)\}$ set of arcs in the cycle

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

A network is a graph to whose nodes and/or arcs we associate weights (cost, length, time etc…). These weights, depending on the context, take on different meanings; for example, in a road network they represent the time required to go from node i to node j.



Directed Graph G=(V,A)



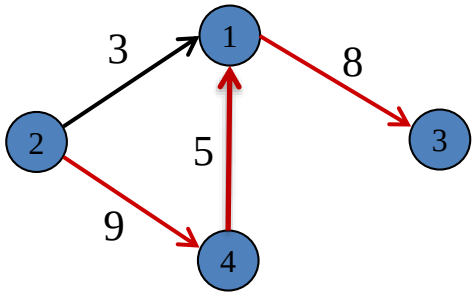
Directed weighted Graph G=(V,A,w)

INTRODUCTION TO GRAPH THEORY AND APPLICATIONS

Given a directed graph G, and given two nodes i_0 and i_q, we define a path P between the two nodes i_0 and i_q, a sequence of nodes and edges:

$$P = \{i_0, (i_0,i_1), i_1, (i_1,i_2), i_2,\ldots, i_{q-1}, (i_{q-1},i_q), i_q\}$$

The weight of a path P is equal to the sum of the weights on the edges forming the path.



Weight of P = 22

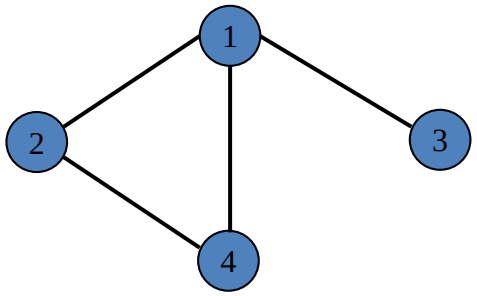
$$P = \{2, (2,4), 4, (4,1), 1, (1,3), 3\}$$

$P = \{(2,4), (4,1), (1,3)\}$ set of arcs in the path

CONNECTIVITY AND SUBGRAPHS

An undirected graph G=(N,E) is connected if for each pair of nodes i and j there is a path from i to j.

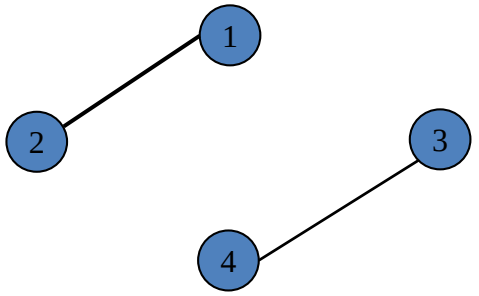
This definition for undirected graphs implies that any two vertices of G are mutually reachable. Hence, G is said to be a strong graph



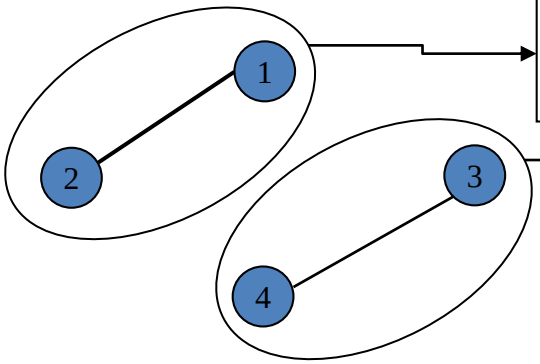
A strong (connected) graph

CONNECTIVITY AND SUBGRAPHS

If, for a graph G, the previous definition does not hold, then the graph is not connected (or disconnected). Non-connectedness implies, however, that it is possible to identify or decompose the graph into connected components



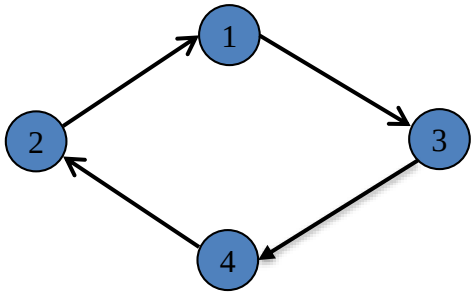
Not connected undirected graph G



Two connected (i.e., strong) components

CONNECTIVITY AND SUBGRAPHS

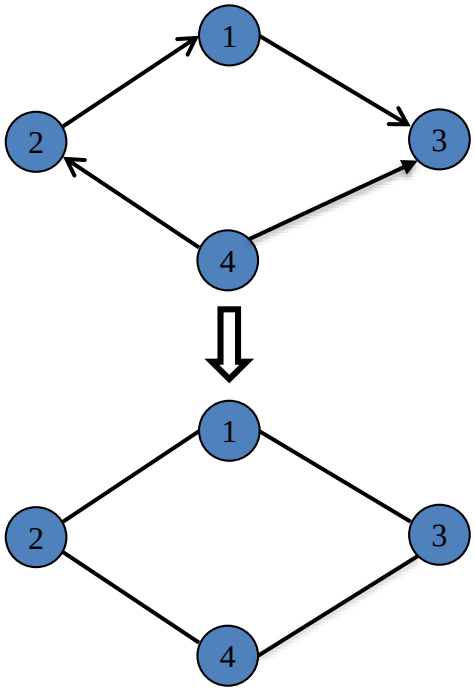
A directed graph G=(N,A) is strongly connected or strong if for any two vertices i and j there is a (directed) path from i to j and viceversa.



Strong directed graph G=(V,A)

CONNECTIVITY AND SUBGRAPHS

A directed graph G is (weakly) connected if disregarding the direction of its arcs, the corresponding (undirected) graph is connected.



Directed graph (not strong)

⇨ Connected undirected graph

⇩

G is weakly connected

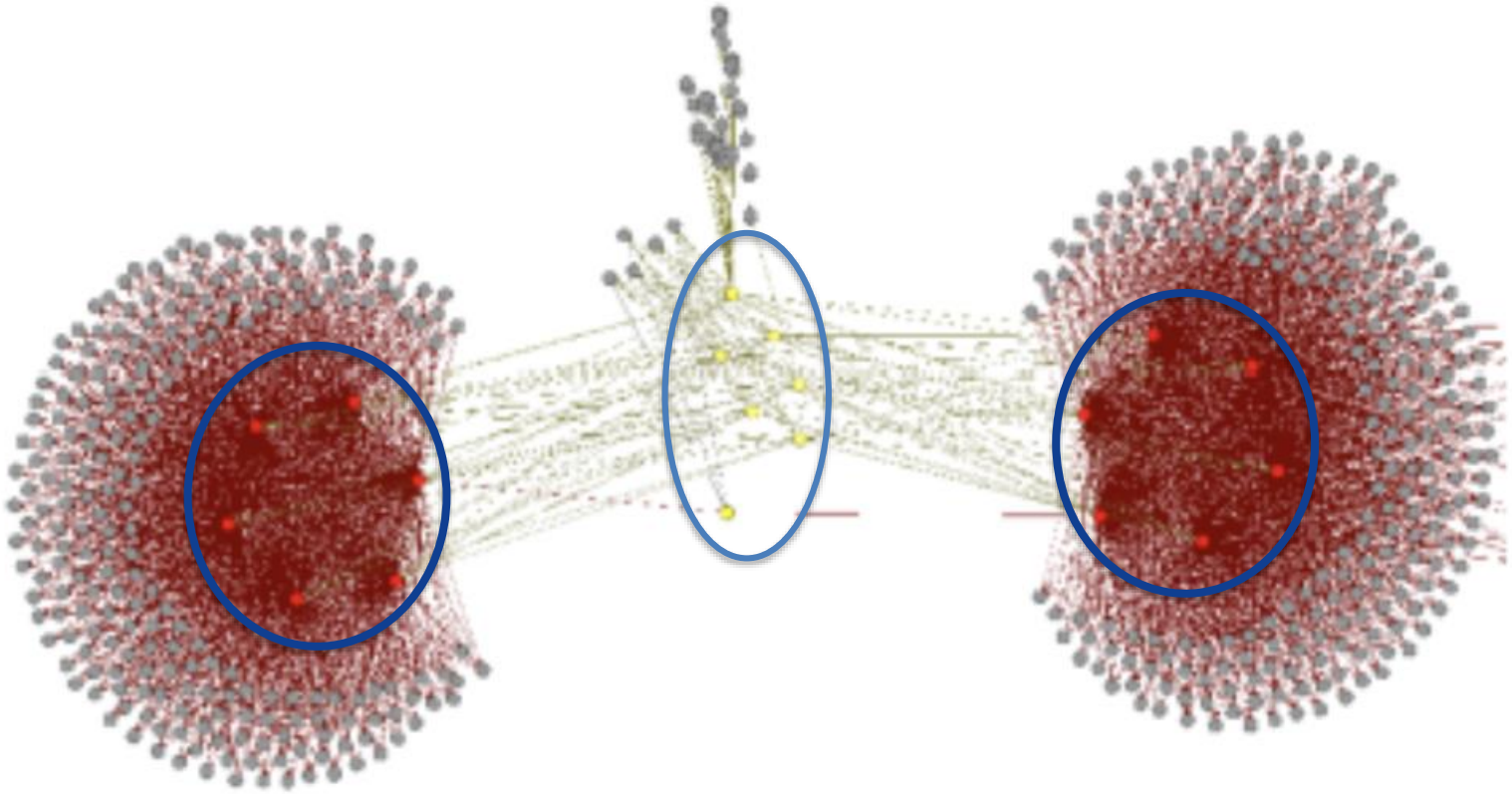
CONNECTIVITY AND SUBGRAPHS: APPLICATION

Gulino, A., Ceri, S., Gottlob, G., Sallinger, E., & Bellomarini, L. (2021, April). Distributed company control in company shareholding graphs. In 2021 IEEE 37th International Conference on Data Engineering (ICDE).

The Italian ownership graph *is characterized by a main large weakly connected component. It contains the main 12 shareholding companies, with the highest out-degree. Six of them own more than 200 other companies, whereas another group of six owns in turn a similar number. The main 12 shareholders are all owned, though not controlled, by 7 non-Italian companies, located at the center of the graph and displayed in yellow. More in detail, it exhibits a high number of small strongly connected components (SCC), 4.058M conglomerates with 15 nodes in the largest one; also, although the graph has one huge weakly connected component (WCC) with 1.598M nodes, most of the nodes are scattered around small WCCs: 635.324K in total, with 6.390 nodes on average and none with more than 175 nodes. On average, each node owns 1.431 companies and is owned by 2.716. There are 30 nodes owning more than 225 firms, with 2 owning more than 1K.*

CONNECTIVITY AND SUBGRAPHS: APPLICATION

Gulino, A., Ceri, S., Gottlob, G., Sallinger, E., & Bellomarini, L. (2021, April). Distributed company control in company shareholding graphs. In 2021 IEEE 37th International Conference on Data Engineering (ICDE).



CONNECTIVITY AND SUBGRAPHS: APPLICATION

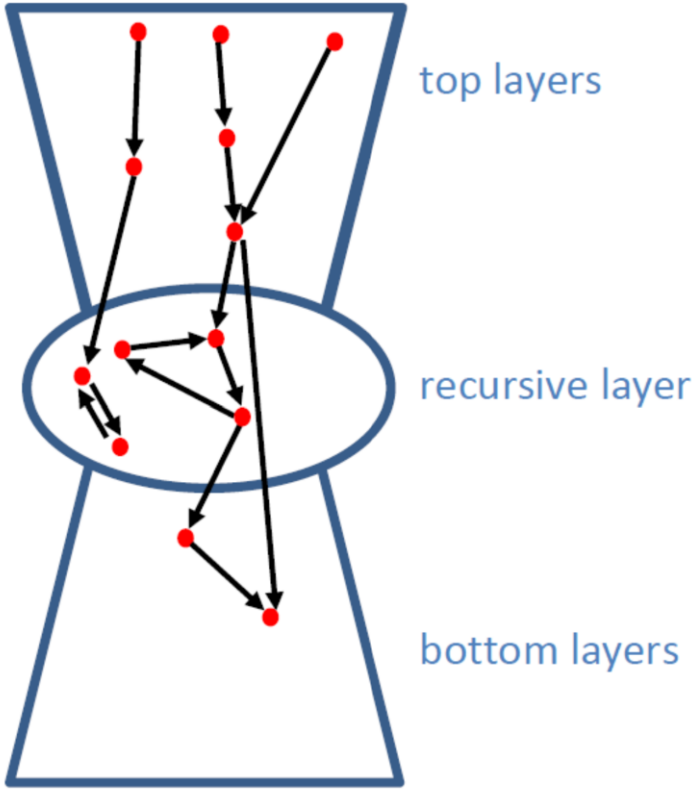
Gulino, A., Ceri, S., Gottlob, G., Sallinger, E., & Bellomarini, L. (2021, April). Distributed company control in company shareholding graphs. In 2021 IEEE 37th International Conference on Data Engineering (ICDE).

The Register of Intermediaries and Affiliates. It is the ownership network of European financial companies run by the Bank of Italy for the European Central Bank. This network can be used as a predictor of the collateral eligibility for asset-backed securities, a core application RIAD is used for in the ESCB. The structure of the RIAD graph is: 91% of the nodes are within a one-node SCC, with one large SCC containing 88 nodes, and all the others with less than 10 nodes; there is one huge WCC, with 57% of the nodes, with the others scattered around small WCCs with 11.968 nodes on average and (apart from the largest one), none with more than 472 nodes.

CONNECTIVITY AND SUBGRAPHS: APPLICATION

Gulino, A., Ceri, S., Gottlob, G., Sallinger, E., & Bellomarini, L. (2021, April). Distributed company control in company shareholding graphs. In 2021 IEEE 37th International Conference on Data Engineering (ICDE).

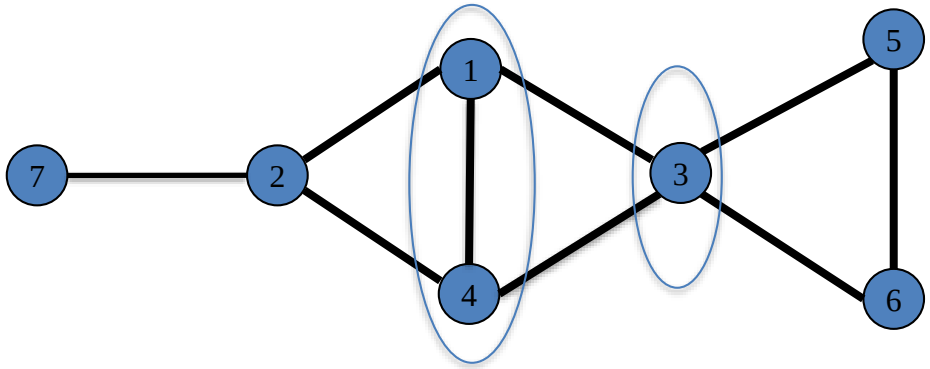
The main characteristic of these networks is that the SCC present a well-known structure called: **Bow-Tie Structure**



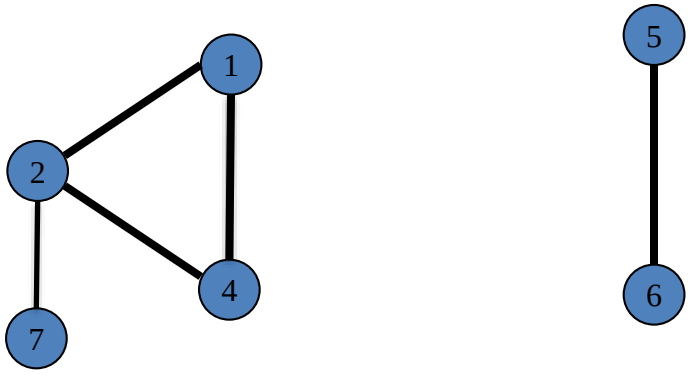
CONNECTIVITY AND SUBGRAPHS

A vertex cut of G=(V,E) is a subset V' of V such that G - V’ is disconnected.

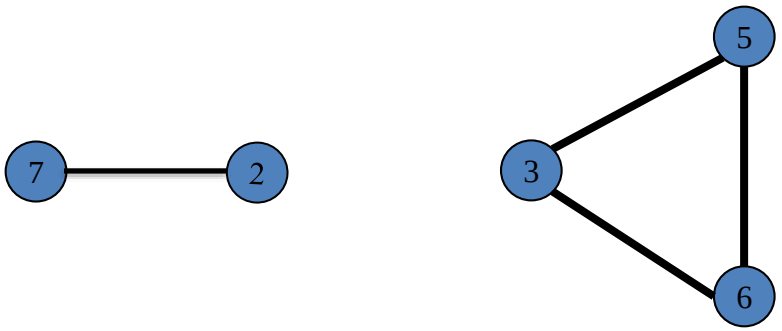
A k-vertex cut is a vertex cut of k elements.



V’={3} is a 1-vertex cut



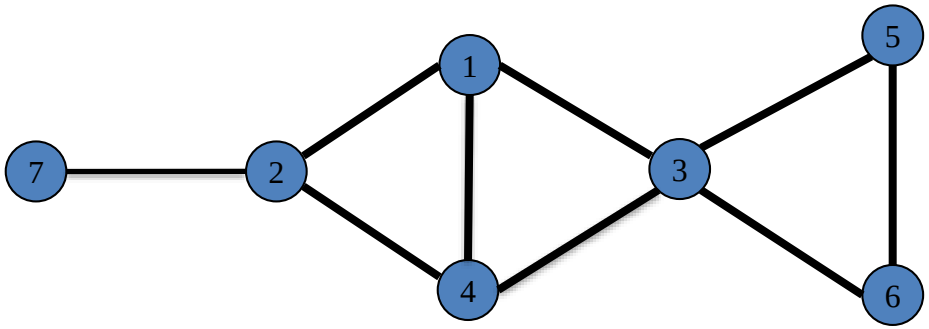
V’={1,4} is a 2-vertex cut



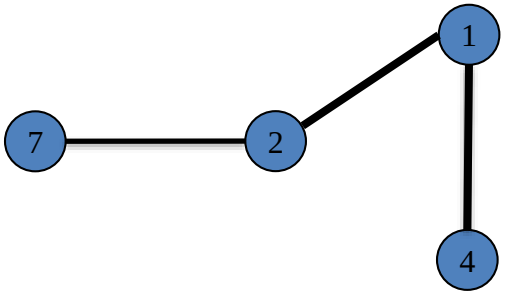
CONNECTIVITY AND SUBGRAPHS

Given G = (N,E), a subgraph H of G is a graph H = (N′,E′) with N′ ⊆ N and E′ ⊆ E.

The subgraph *induced* by Y ⊆ N on G is the graph G(Y) = (Y,E∩(Y × Y)).

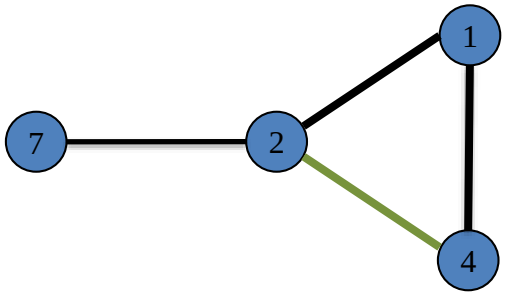


A subgraph H=(N′,E′)



N′={1,2,4,7}
E′={(1,2), (1,4), (2,7)}

An *induced* subgraph H=(Y, E∩(Y × Y))

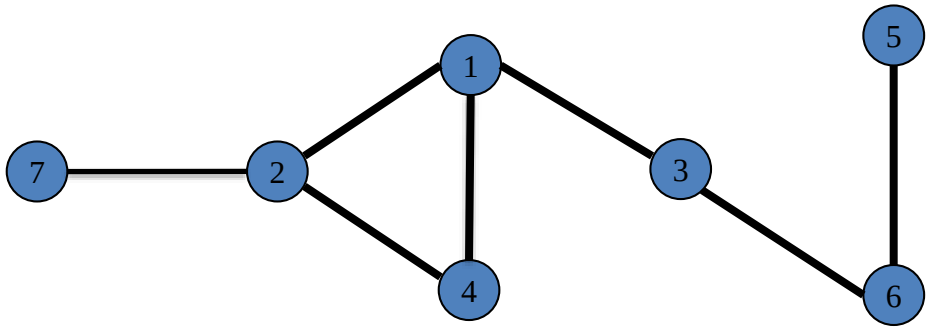


Y={1,2,4,7}
E∩(Y × Y) ={(1,2), (1,4), (2,7), (2,4)}

CONNECTIVITY AND SUBGRAPHS

Given G = (N,E), a *partial subgraph* H of G is a graph H = (N,E′) with E′ ⊂ E.

A *partial subgraph* obtained from G is the graph:

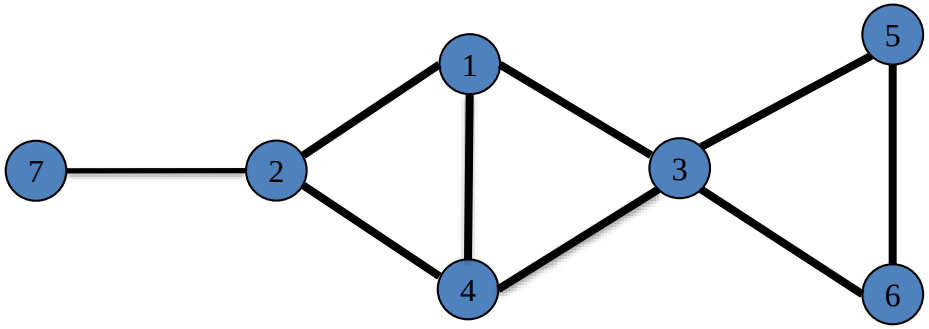


N={1,2,3,4,5,6,7}

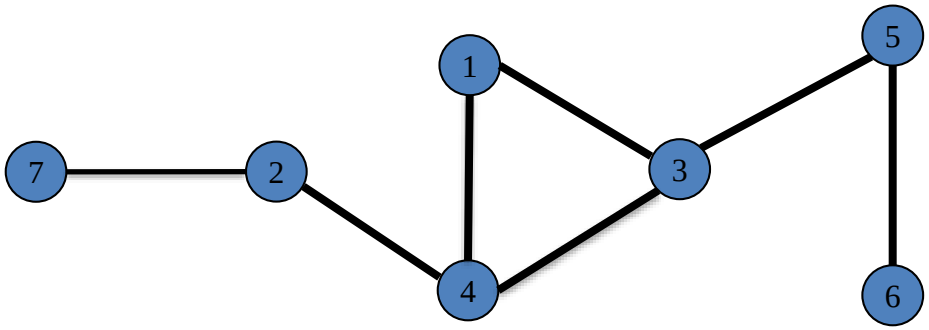
E′ ={(1,2), (1,4), (1,4), (2,4) (2,7), (3,6), (5,6)}

CONNECTIVITY AND SUBGRAPHS

Given G = (N,E), if a subgraph H of G is H = (N,E′) and E′ ⊂ E, the subgraph H is called a *spanning* subgraph of G.



A spanning subgraph H=(N,E′)

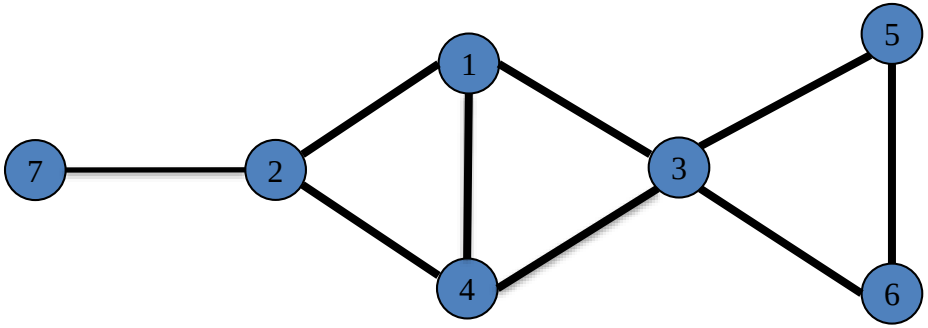


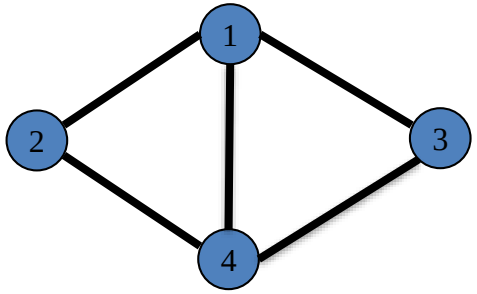
H={1,2,3,4,5,6,7}
E′={(1,3), (1,4), (2,4), (2,7), (3,4), (3,5), (5,6)}

CONNECTIVITY AND SUBGRAPHS

A member of a collection of sets is said to be maximal if it cannot be expanded to another member by addition of any element.

A maximal connected subgraph of a given graph G that has no 1-cut vertex is called a block, that is, it is maximal since is not contained in any other maximal connected subgraph.





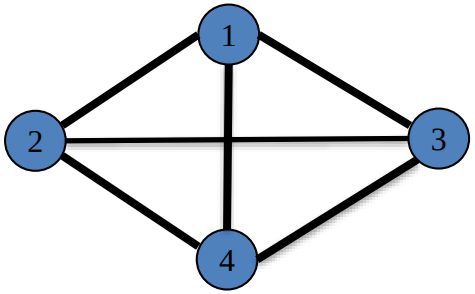
Maximal connected subgraph G=(N',E')

⇩

Block

TYPES OF GRAPHS

A graph G=(V,E) is said to be *complete* if for every pair of vertices i and j there exits an edge joining them.

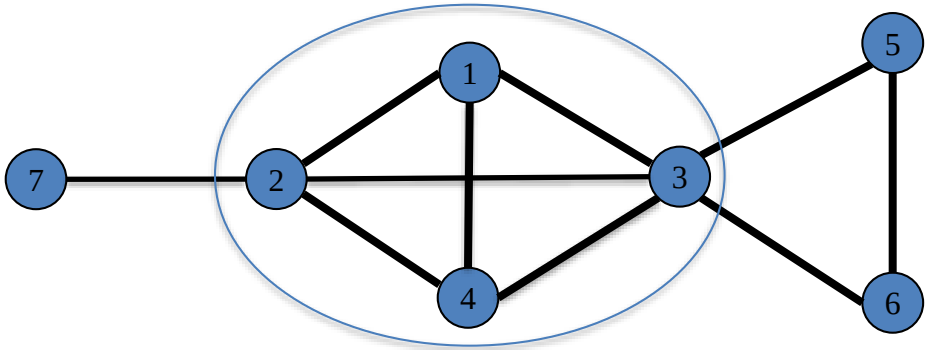


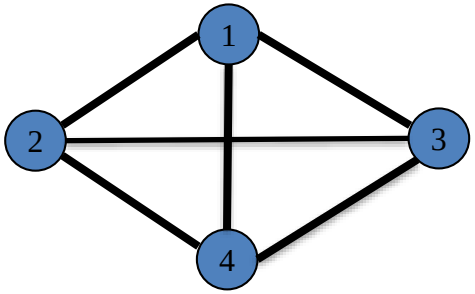
Complete graph G=(V,E), with |V|=n

$$d(i) = \frac{n(n-1)}{2} \qquad i=1,\ldots,n$$

TYPES OF GRAPHS

A complete graph can be also an induced subgraph H of a graph G=(V,E)

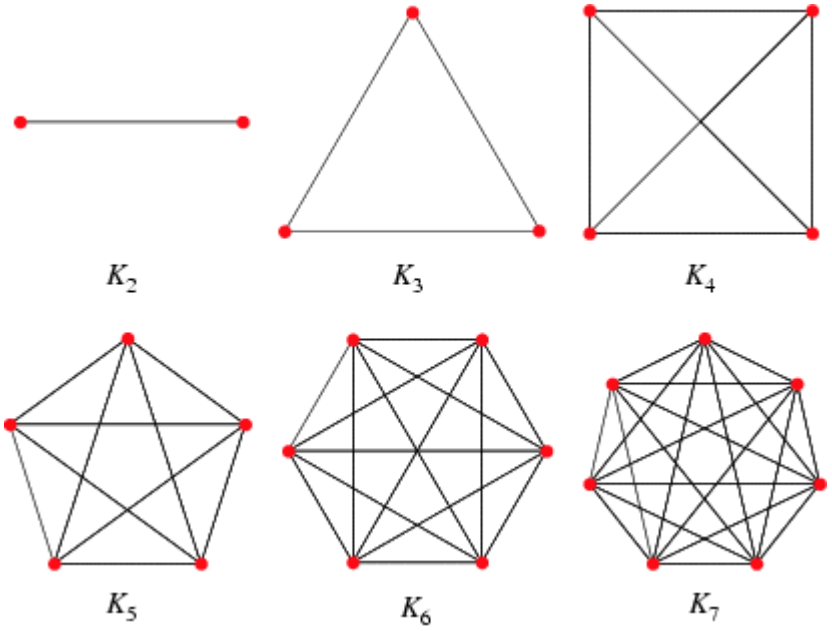




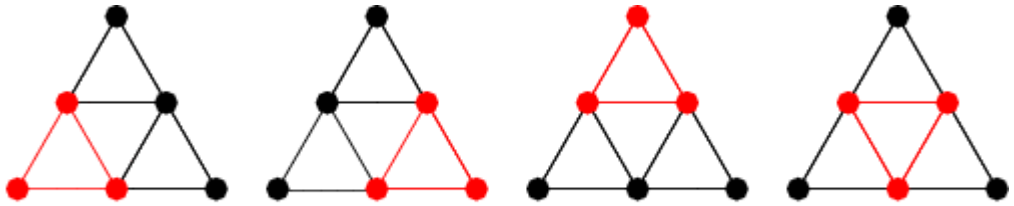
Complete subgraph H=(V',E') ⇨ Clique of G

TYPES OF GRAPHS

Complete graph/cliques

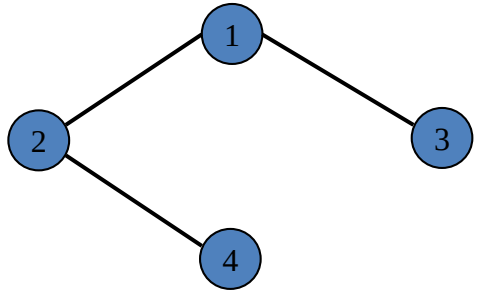


Cliques of order 3



TYPES OF GRAPHS

Given a set of vertices V a minimal graph G that connects all the vertices of G is a Tree. That is, deleting any edge disconnects G.



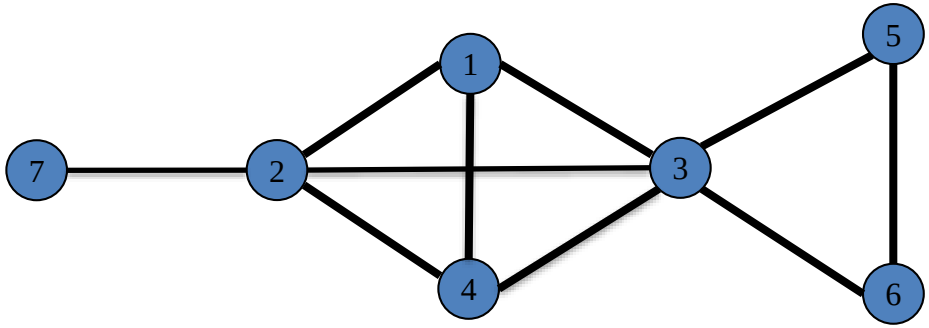
An undirected graph T = (V,E) is a Tree if one of the three (equivalent) definitions holds:

- T is a graph on n vertices and n-1 edges
- T is a connected graph without a cycle
- T is a graph in which every pair of vertices is connected with one and only one «elementary» path

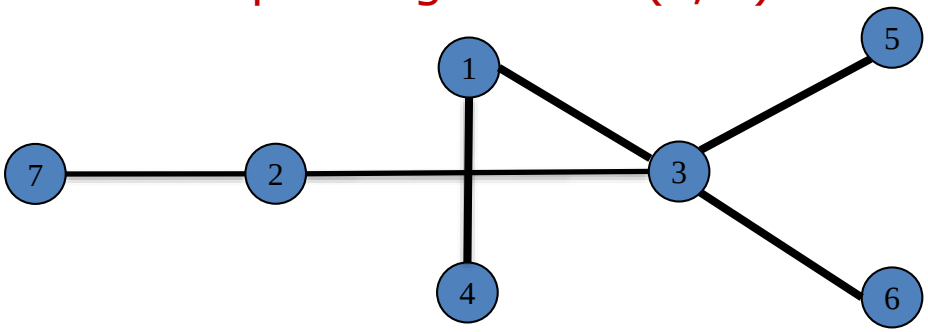
TYPES OF GRAPHS

Given a graph G=(V,E) a subgraph T of G which is a tree on the same set of vertices of G is a spanning tree of G

A graph G=(V,E)

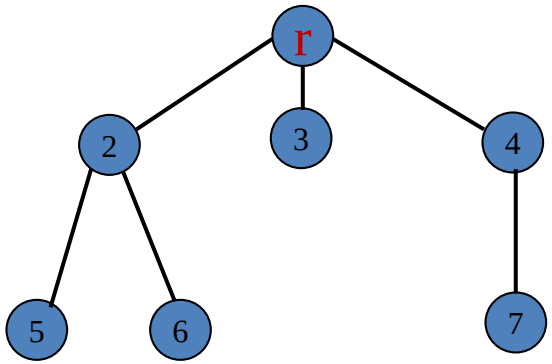


A spanning tree T=(V,E')



TREES

A tree T is rooted if one of its vertices is is identified as the root r

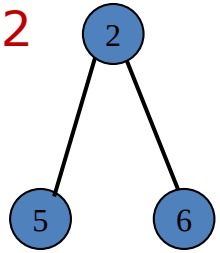


Level 0

Level 1

Level 3

T(2) is a «subtree» rooted at vertex 2



2 and 4 are «internal nodes»

3, 5, 6 and 7 are «leaves»

4 is the «father» of 7

7 is the «son» of 4

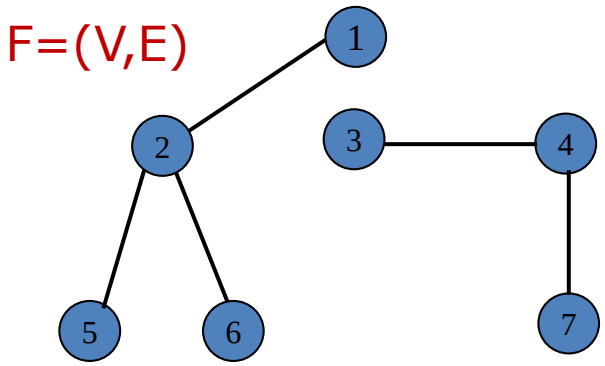
6 is the «descendant» of r

The root r is the «ancestor» of 6 and of 5

TREES

Given an undirected graph G=(V,E), it is «forest» if it is cycle-free.

A forest is a graph whose connected components are trees



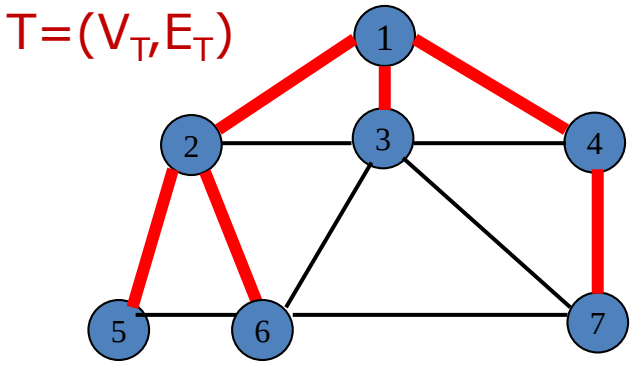
MINIMUM SPANNING TREE - MST

Given a graph G=(V,E), find a Minimum (cost) Spanning Tree of G=(V,E)

Assume that to each edge $e=(i,j)\in E$ is assigned a weight or cost $w(e) = w_{ij} \geq 0$.

To each spanning tree T of G we define the total weight or cost of T

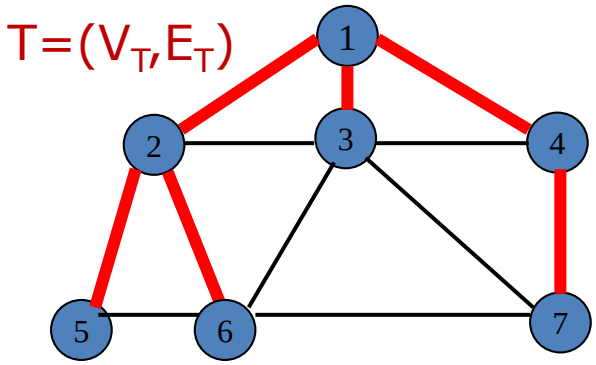
$$w(T) = \sum_{e \in E_T} w(e)$$



MINIMUM SPANNING TREE - MST

Problem: Given a graph G=(V,E), find a MST $T^*=(V_T,E_T)$

$$\text{MIN } w(T) = \sum_{e \in E_T} w(e) = w(T^*)$$



MINIMUM SPANNING TREE - MST

Greedy Algorithm: a greedy algorithm is a procedure that works by making the best local choice at each step.

In some classes of problems (such as the MST) this behaviour leads to the identification of the global optimum (i.e. the best possible solution), but more generally, it is NOT TRUE that a greedy strategy succeeds in identifying the optimum of any problem.

The Minimum Spanning Tree (MST) problem belongs to a "class" of problems for which a greedy procedure guarantees to always find the optimal solution.

MINIMUM SPANNING TREE - MST

Greedy Algorithm

Step 0: set $E_T = \varnothing$ Start with an empty set of edges

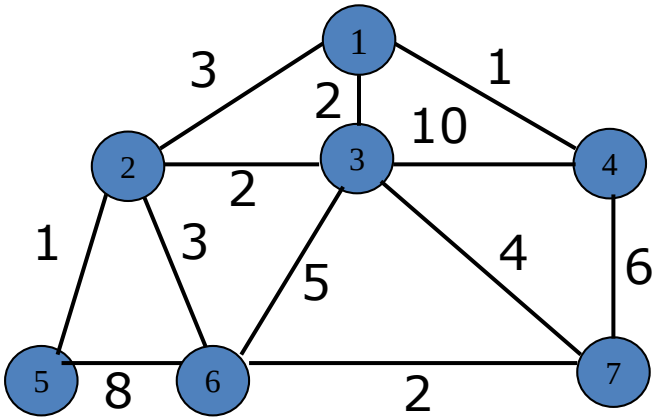
Step 1: Find the least cost edge that does not belong to E_T, such that, together with the edges already in E_T forms a forest (i.e., the new edge must not form cycles)

update $E_T = E_T \cup \{e\}$

Step 2: if $|E_T|=|V|-1$ STOP, otherwise go to Step 1

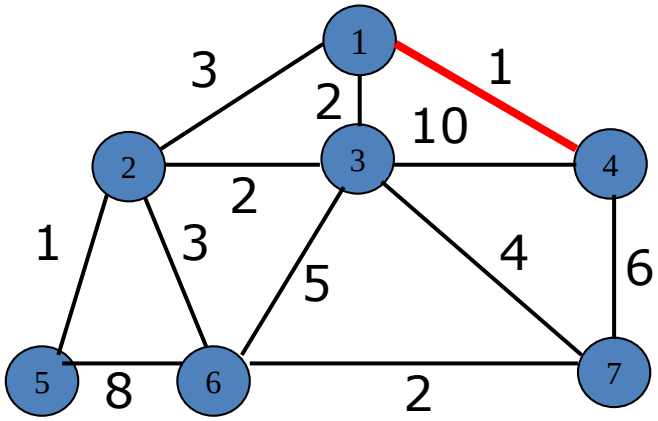
MINIMUM SPANNING TREE - MST

Example: let $G=(V,E)$ be the weighted graph below with $n=|V|=7$ and $m=|E|=12$.



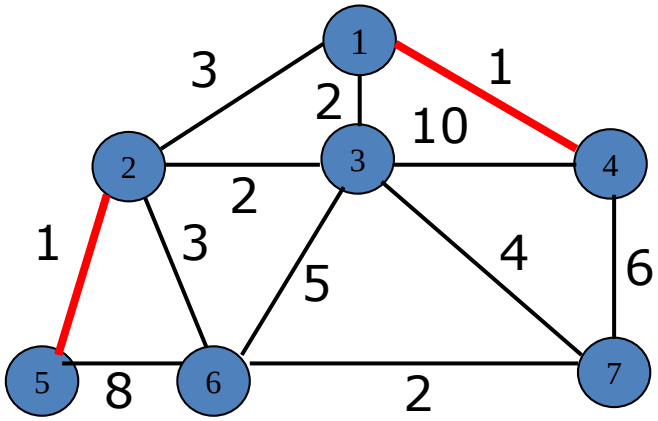
MINIMUM SPANNING TREE - MST

Example: let $G=(V,E)$ be the weighted graph below with $n=|V|=7$ and $m=|E|=12$.



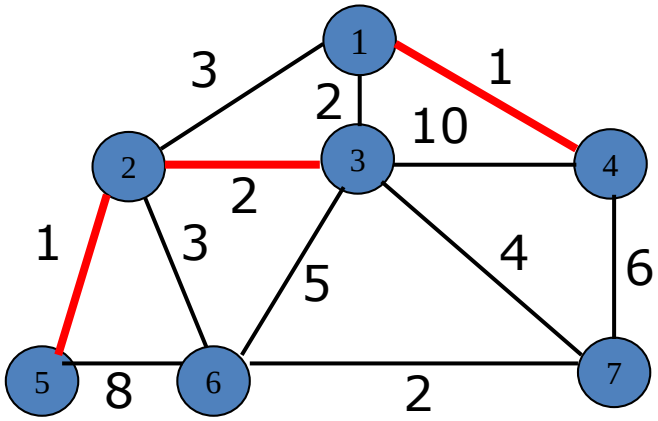
MINIMUM SPANNING TREE - MST

Example: let $G=(V,E)$ be the weighted graph below with $n=|V|=7$ and $m=|E|=12$.



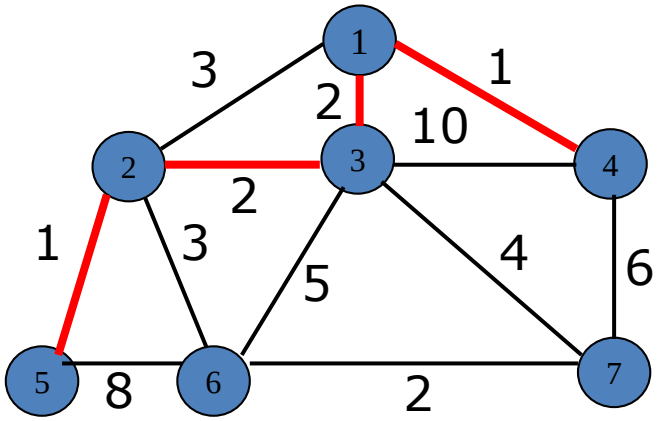
MINIMUM SPANNING TREE - MST

Example: let $G=(V,E)$ be the weighted graph below with $n=|V|=7$ and $m=|E|=12$.



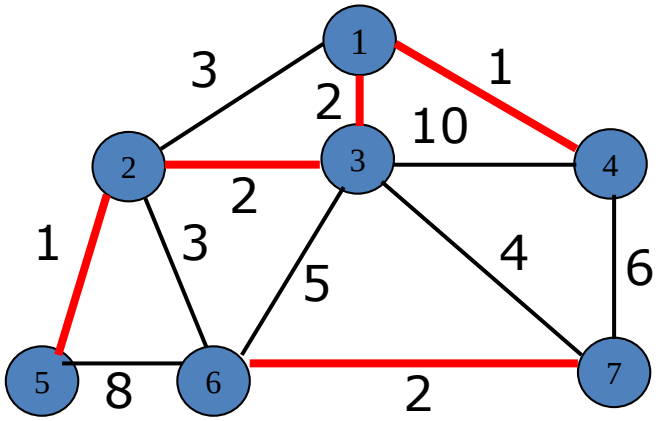
MINIMUM SPANNING TREE - MST

Example: let $G=(V,E)$ be the weighted graph below with $n=|V|=7$ and $m=|E|=12$.



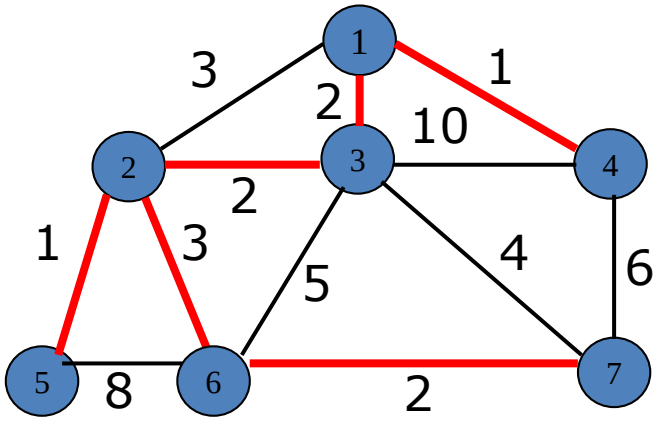
MINIMUM SPANNING TREE - MST

Example: let $G=(V,E)$ be the weighted graph below with $n=|V|=7$ and $m=|E|=12$.



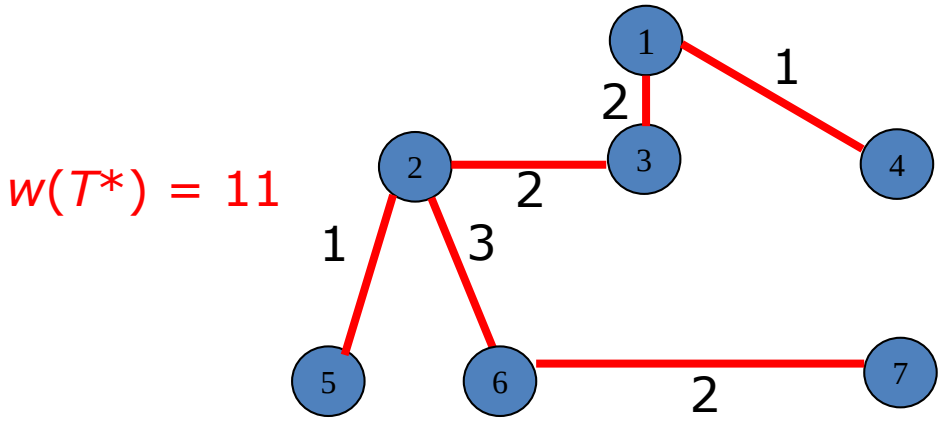
MINIMUM SPANNING TREE - MST

Example: let $G=(V,E)$ be the weighted graph below with $n=|V|=7$ and $m=|E|=12$.



MINIMUM SPANNING TREE - MST

Example: let $G=(V,E)$ be the weighted graph below with $n=|V|=7$ and $m=|E|=12$.



The MST may not be unique

MINIMUM SPANNING TREE - MST

Remarks:

1. The algorithm can be 'implemented' by following different strategies in Step 1. The different strategies characterise the way the edges are inserted into the set E_T.

2. The algorithm stops in two cases:
   i) either it finds the MST;
   ii) if it recognises that the tree T does not exist since the starting graph G is not connected

In case (ii), the algorithm finds a spanning forest of G.

By construction during the execution of the algorithm, cycles are NEVER generated.