MOHAMED BOUDIAF UNIVERSITY OF M'SILA

DEPARTMENT OF COMPUTER SCIENCE

The Foundations of Graph Theory

Chapter 1: Basic Concepts

Dr. SAID KADRI

Lecturer, Class A

Department of Computer Science, Faculty of Mathematics and Computer Science,

Mohamed Boudiaf University of M'sila

E-mail:kadri.said28@gmail.com

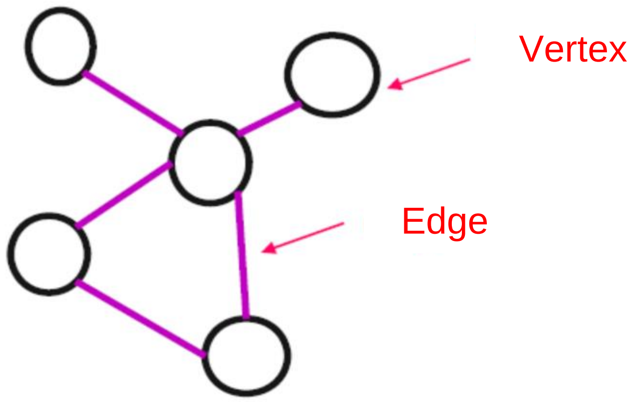
Website:https://kadrisaid28.wixsite.com/sgadri

2017 - 2018

Definition of a Graph

- **A Graph is a collection of vertices connected by edges or arcs.**

- **We call a graph the pair G(X, U) such that:**
- **X = {x1, x2, ….., xn} is the set of vertices of the graph.**
- **U={u1, u2, …., un} is the set of arcs of the graph**
- **U⊂XxX**



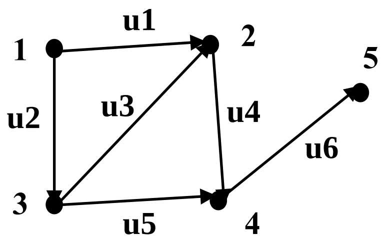
- **The term network:** is used to denote real systems (road network, electrical network, computer network, etc.).

- **The term Graph**: is used to denote a mathematical representation of a network.

But, generally:“**Network**” ≡ “**Graph**”

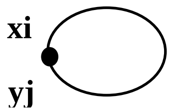
Properties of a graph:

- $n=|X|$ is called the order of the graph G.
- $m=|U|$ is the size of the graph (nb.arcs)
- a vertex x_i is represented by a point.
- An arc $U=(x_i, x_j)\in XxX$ is represented by an arrow or a line segment (depending on the type of graph, directed/undirected)



$n=|X|=5$ $m=|U|=6$

- If $x_i = x_j$ ==> u=(xi, xj) is represented by a loop (i.e. the two vertices are the same).



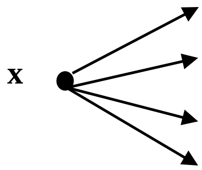
- The external half-degree of a vertex x is the number of arcs for which x is the initial endpoint or the number of outgoing arcs of x.

  We denote $d^+(x)$.
- The internal half-degree of the a vertex x is the number of arcs for which x is the terminal end or the number of arcs entering to x.

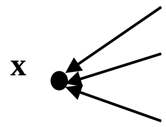
  We denote $d^-(x)$.

- The degree of the vertex x is the number of arcs for which x is the initial or the terminal end.

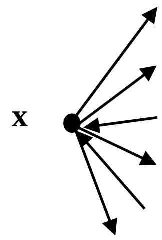
  We denote $d(x) = d^{+}(x) + d^{-}(x)$.



$d^{+}(x) = 4$

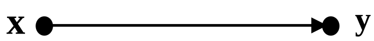


$d^{-}(x) = 3$

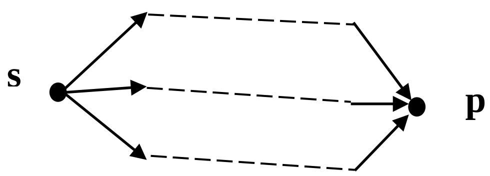


$d(x) = d^{+}(x) + d^{-}(x) = 7$

- If $(x, y) \in U$, then x is said to be the predecessor of y, y is the successor of x.



- $\Gamma^{+}(x)$: the set of successors of x.
- $\Gamma^{-}(x)$: the set of predecessors of x.
- A source vertex *s* is a vertex that has no predecessor ($\Gamma^{-}(s)=\phi$/ there are only outgoing arcs).
- A sink vertex *p* is a vertex that has no successor ($\Gamma^{+}(x) = \phi$) / there are only incoming arcs).
- An isolated vertex x is a vertex that has no neighbors (neither predecessor nor successor) (($\Gamma^{+}(x)$= ($\Gamma^{-}(x) = \phi$). So this is a source and sink vertex at the same time, it is also called an inaccessible vertex.



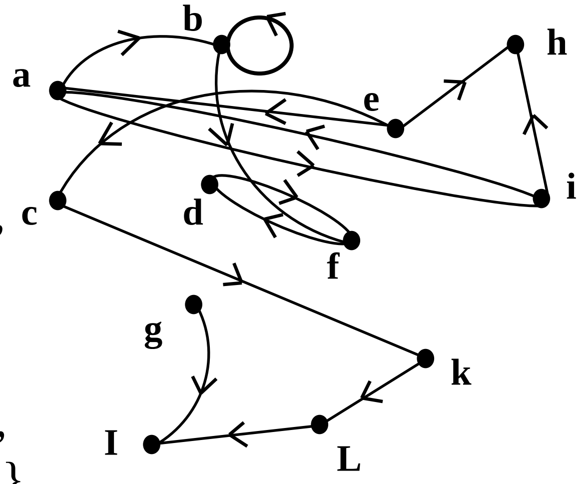
Example:

X={a, b, c, d, e, f, g, h, I, j, k, l}

U={(a, b), (a, i), (b, f), (f, d), (d, f),

(e, a), (e, h), (k, l), (e, c), (c, k), (b, b),

(g, j), (l, j)}



$\Gamma^+(f) = \{d\}, \Gamma^+(d) = \{f\}, \Gamma^+(b) = \{f, b\},$
$\Gamma^+(j) = \phi, \Gamma^+(e) = \{a, h, c\}, \Gamma^-(f) = \{b, d\},$
$\Gamma^-(e) = \phi, \Gamma^-(a) = \{i, e\}$

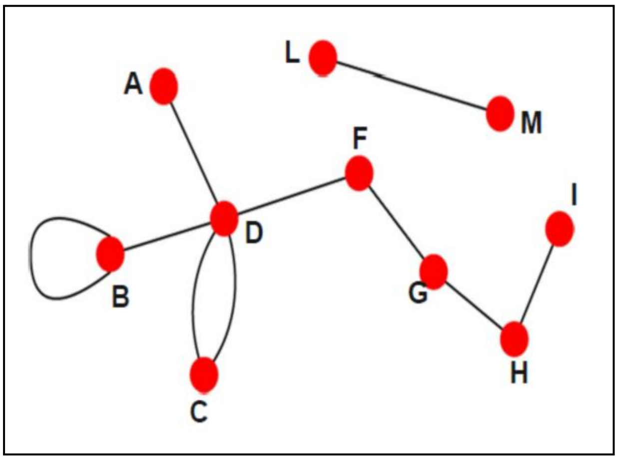
Examples of graphs:

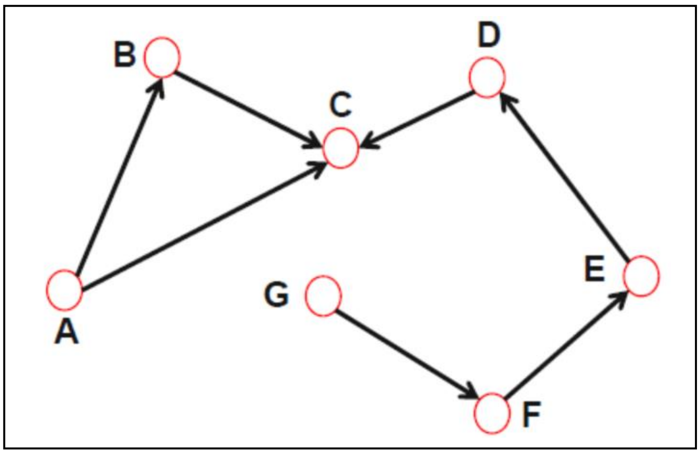
- Graph representing a road network, where vertices represent intersections and arcs represent roads.
- Drinking Water Distribution Network AEP in a city.
- Computer network in a compagny.
- The international information network Internet.

Directed graph and undirected graph

Undirected graph	Directed graph
Lines: unoriented (*symmetrical*)	Lines: oriented
➔edges	➔ *Arcs*

Undirected Graph **Di-Graph= Directed Graph**





Definition of a valued graph

A valued graph is a graph G(X, U, C) such that we associate with the graph G(X, U) a function F defined as follows:

F: U ⟶ R

F(u) = c is called the weight of the arc u, and we denote

c(x, y) Or c(u)

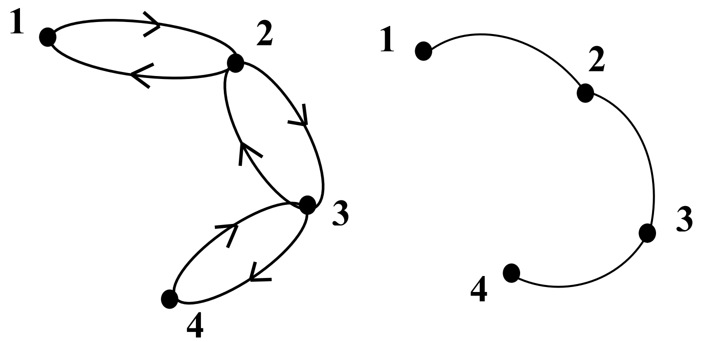
With:

- c(x, y): length of the road section (x, y).
- Capacity of the road section (x, y).
- Flow rate of a drinking water pipe.
- Travel price between cities x, y.

Types of graphs

1. Symmetric graph

A graph G(X, U) is said to be symmetric if ∀x, y∈X, if (x, y)∈U ===> (y, x) ∈U

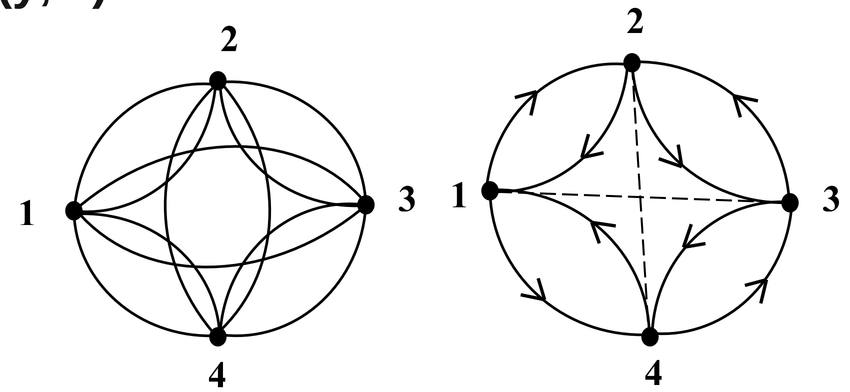


A symmetric graph is usually represented without orientation of the arcs. We talk about edges instead of arcs.

2.Complete graph

A graph G(X, U) is said to be complete if only if:

∀x, y∈X, (x, y)∈U and (y, x) ∈U



Complete graph **Graph not complete**

3. Simple graph

A graph G(X, U) is said to be simple if only if:

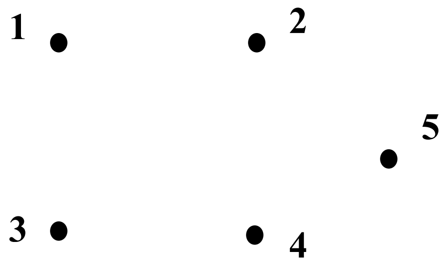
- **Does not contain any loops.**
- **$\forall x, y \in X, \exists$ at most $u=(x, y) \in U$**

4. Empty graph

A graph G(X, U) is said to be empty if there are no vertices, arcs or edges. ($X=\phi$, $U=\phi$).

5. Trivial graph

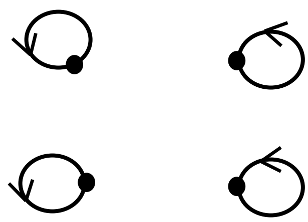
A graph G(X, U) is said to be trivial if there are vertices, but there are no arcs or edges. ($U=\phi$).



6. Reflexive graph

A graph G(X, U) is said to be reflexive, if there exists a loop at each vertex x of G.

- **$\forall x \in X \Longrightarrow (x, x) \in U$**



7. Anti-symmetric graph

A graph G(X, U) is said to be anti-symmetric if:

- **$\forall$x, y$\in$X, if (x, y)$\in$U ===> (y, x)$\notin$U**

8. Transitive graph

A graph G(X, U) is said to be transitive if:

- **$\forall$x, y, z$\in$X,if (x, y)$\in$U and (y, z)$\in$U ===> (x, z)$\in$U**

9. Inverse of a graph

The inverse of a graph G(X, U) is the graph G'(X, U') deduced from G by reversing the direction of its arcs.

OBS: the inverse of G' is G itself.

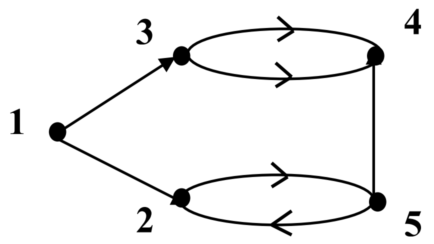
10. Complementary graph

The complementary graph of the graph G(X, U) is the graph G*(X, U*) such that:

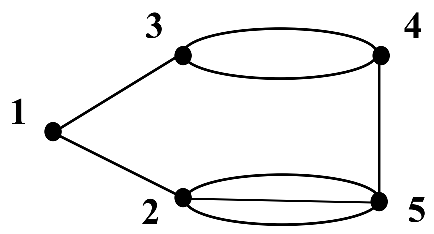
(x, y)$\in$U ===> (x, y)$\notin$U*

11. Multigraph

if G is a directed graph, x and y are two vertices of this graph, if x and y are linked by two arcs u1, u2 of the same direction, G is said to be a multigraph and the arc (x, y) is said to be a multiple arc.



a directed multiple graph



an undirected multiple graph

- **(3, 4) a multiple arc**
- **(2, 5) a symmetrical arc**

12. The bipartite graph

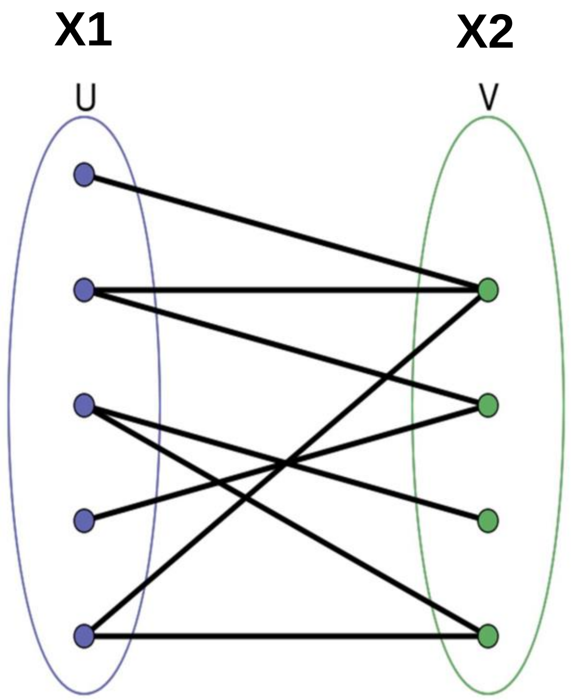
A bipartite graph (or bi-graph) is a graph whose set of vertices can be divided into two disjoint subsets of vertices X1 and X2 such that no two vertices in the same subset are adjacent (i.e., any arc of G has one endpoint in X1 and the other in X2).

And we note:

G(X, U) with: $X = X1 \cup X2$ and $X1 \cap X2 = \phi$

And if u=(x1, x2) and $x1 \in X1 \implies x2 \in X2$

OR $x1 \in X2 \implies x2 \in X1$

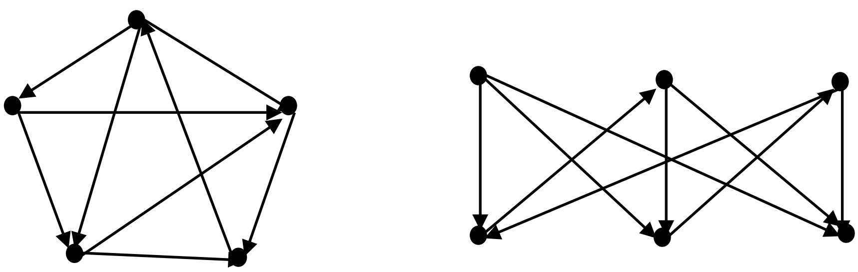


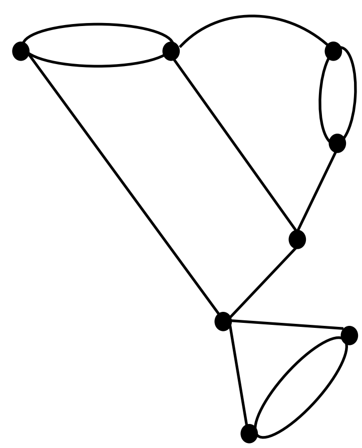
13. Planar graph

A graph G is said to be planar if it is possible to draw it in the plane in such a way that any two arcs do not intersect except at their ends.

This trace in the plane is called a planar representation..

Example:





G1, G2: two non-planar graphs **G3 a planar graph**

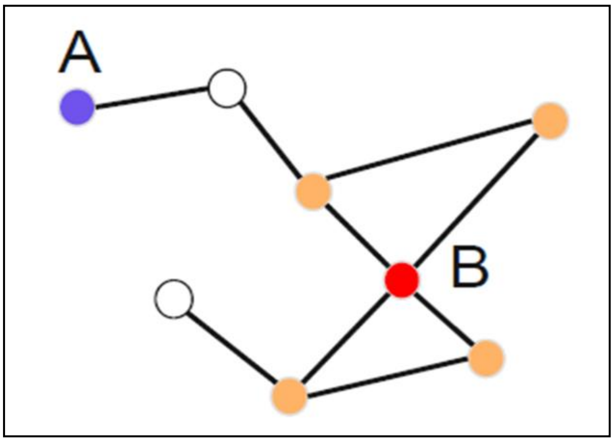
Degree of a vertex

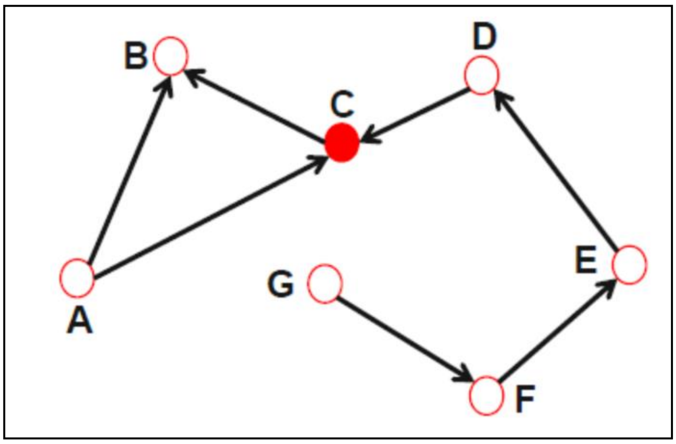
Undirected graph

Degree: the number of edges which connect vertices.

Directed/Oriented Graph

in a directed graph, we defines external degree and internal degree
The total degree of a vertex is the sum of the external degree and the internal degree.





Definition of a path

We call a path of G(X, U) the sequence of vertices

$a_1, a_2, \ldots, a_p$ (without cut) such that:

- $(a_i, a_{i+1}) \in U$ with i=1, 2, …, p-1.
- a_i is the initial endpoint of the path
- a_p is the terminal end of the path.

Definition of a simple path

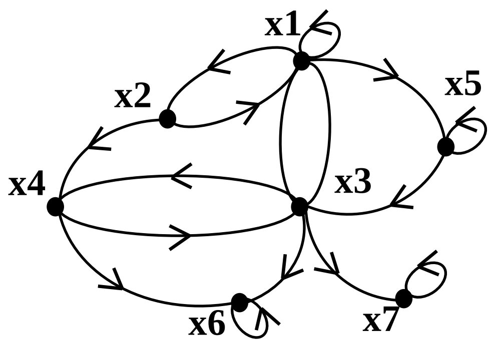
$a_1, a_2, \ldots, a_p$ is a simple path such that:

$\forall i \neq j, (a_i, a_{i+1}) \neq (a_j, a_{j+1})$ (i.e., an arc is taken only once).

Definition of an elementary path

is a path that only takes vertices once.

Examples:



x2x1x5x3x1x5x5x3x6 a path that is not simple, nor elementary.

x2x1x5x3x1x2x4 a simple path not elementary

x1x2x4x3x6 elementary path

Remark

If $a_1a_2a_3 \ldots .a_p$ is a path

- **a_1 is an ascendant of a_p**
- **a_p is a descendant of a_1**

Definition of a circuit

A circuit is a path $a_1a_2a_3 \ldots .a_p$ such that: $a_1=a_p$

i.e., the two initial and terminal ends are merged (a closed path).

Observations

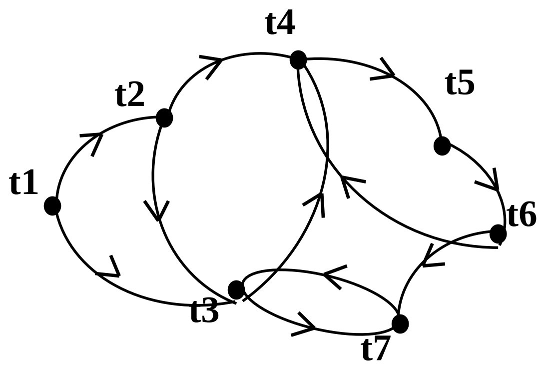
- **In the same way we can define a simple circuit, an elementary circuit.**
- **The notion of the circuit path takes into consideration the direction of the arcs in the graph.**
- **A simple circuit is a closed path that takes arcs only once.**
- **An elementary circuit is a closed path that takes vertices only once.**

Special paths and circuits

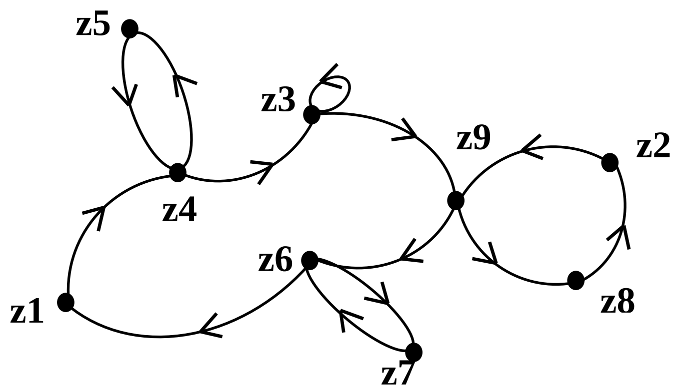
1. **A Hamiltonian path:** **is an elementary path that passes through all the vertices of the graph.**
2. **An Eulerian path:** **is a simple path that passes through all the arcs of the graph once and only once.**
3. **A Hamiltonian circuit:** **is an elementary circuit that passes through all the vertices of the graph.**
4. **An Eulerian circuit:** **is a simple, closed path that passes through all the arcs of the graph once and only once.**

Examples:

$t_1t_2t_3t_4t_5t_6t_7$ is a Hamiltonian path



$z_1z_4z_5z_4z_3z_3z_9z_8z_2z_9z_6z_7z_6z_1$ is an Eulerian circuit

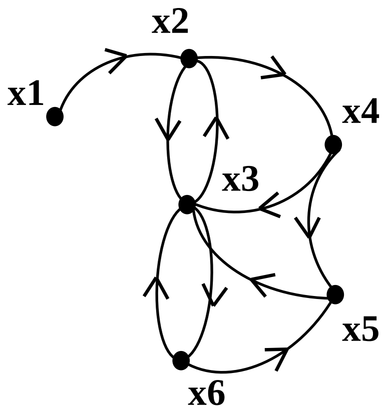


Definition of a chain

A chain of an undirected graph (or directed but the orientation is not taken into consideration) G(X, U) is a sequence of vertices $a_1, a_2, \ldots, a_p$ (without cut) such that:

$(a_i, a_{i+1}) \in U$ or $(a_{i+1}, a_i) \in U$ with $i=1, 2, \ldots, p-1$.

Example:



$x_1x_2x_4x_3x_6x_5x_3x_2$ is a chain (a path)

$x_1x_2x_5$ is not a chain

Definition of a simple chain

It is a chain where the (arcs/edges) are taken only once.

Example:

x1x2x3x2x4x3x5x6x3 is a simple chain.

Definition of an elementary chain

It is a chain whose all vertices of the graph are taken only once.

Example:

$x_1x_2x_4x_3x_6$ is an elementary chain.

Definition of a cycle

A cycle is a chain $a_1a_2a_3$ ….a_p, such that: $a_1=a_p$

i.e., the initial end and the terminal end are merged

Example:

$x_2x_4x_3x_6x_5x_3x_2$ is a cycle.

Observations

- **In the same way we can define a simple cycle, an elementary cycle.**
- **In defining a chain or cycle the orientation of the graph is not important.**

Special chains and cycles

1. **A Hamiltonian chain: is an elementary chain that passes through all the vertices of the graph.**
2. **A Eulerian chain: is a chain that passes through all the arcs of the graph once and only once.**
3. **A Hamiltonian cycle: is an elementary cycle that passes through all the vertices of the graph.**
4. **An Eulerian cycle: is a simple, closed chain that passes through all the arcs of the graph once and only once.**

Definition of a subgraph

Let G(X, U) be a graph

G'(X', U') is a subgraph of G if X'⊂X

U'={u∈U/ both ends of u belong to X'}

(i.e., we delete vertices and arcs from the original graph)

Definition of a partial graph

Let G(X, U) be a graph, we call partial graph of G, the graph G'(X', U') such that: X'=X, U'⊂U

(it is a graph having the same set X of vertices, but a different set of arcs U'⊂U).

Definition of a partial subgraph

Let G(X, U) be a graph, we call a partial subgraph of G, a partial graph of a subgraph of G.

(i.e. we delete vertices and arcs)

Example:

Let the graph G(X, U) be:

X={a, b, c, d, e, f, g}; U={(a, b), (a, c), (c, d), (b, f), (f, g),

(e, a), (e, f), (e, d), (f, d)}

1. G'(X', U') is a subgraph (Fig 1)
   X'={a, e, f, d}; U'={(e, a), (e, f), (d, e), (f, d)}

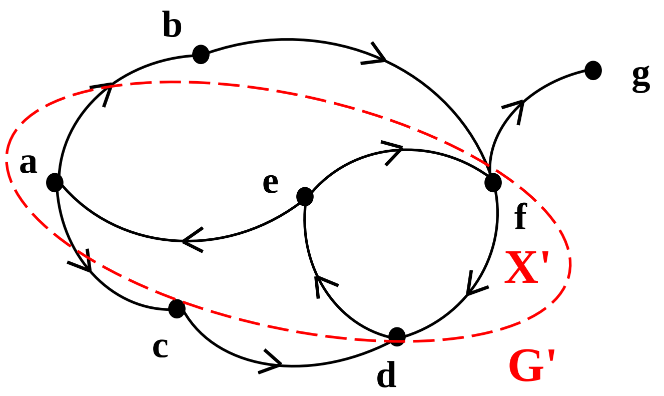


Fig 1

2. G"(X, U") is a partial graph(Fig 2)

   U"={(a, b), (a, c), (c, d), (d, e), (e, f), (f, d)}

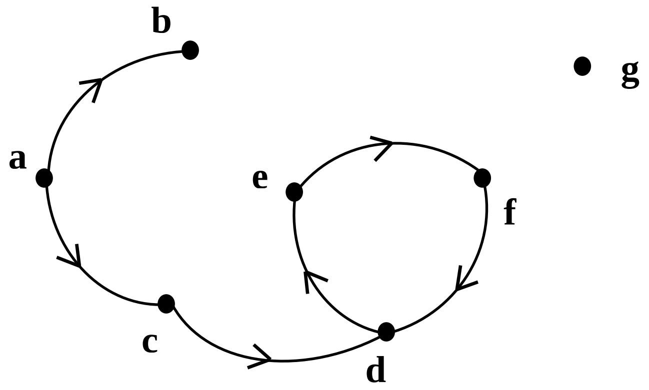


Fig 2

3. G"'(X', U"') is a partial subgraph(Fig 3)

   X'={a, e, f, d}; U"'={(a, e), (e, f), (f, d)}

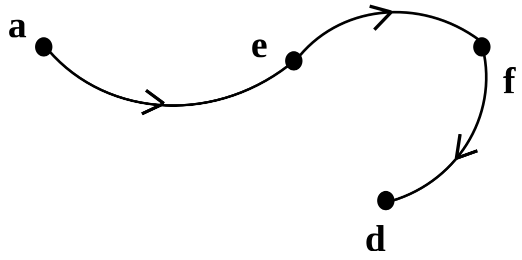


Fig 3

Practical example:

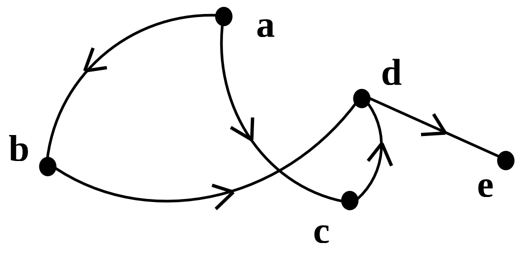
For a country's road network

1. **All the roads in a wilaya are a subgraph of the road network.**
2. **The set of national roads is a partial graph of the road network.**
3. **All the national roads in a wilaya are a partial subgraph of the road network.**

Definition of a connected graph

A graph G(X, U) is said to be connected if:

- **∀x, y∈X, there is a chain between x and y (the direction of the arcs is not important)**



A connected graph; but is not strongly connected

Definition of a maximal connected component

Is a connected subgraph and maximal for this property. That is, if we add a vertex to this component we destroy the connectivity.

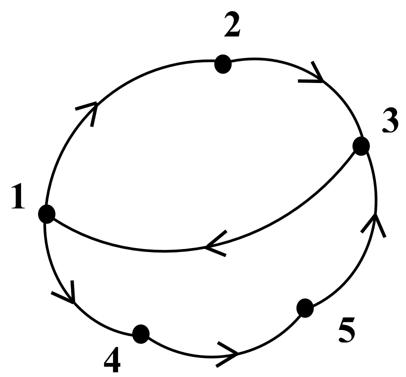
OBS: if G is a connected graph, it itself constitutes the only maximal connected component.

Definition of a strongly connected graph

A graph G(X, U) is said to be strongly connected if:

∀x, y∈X, there is a path from x to y and another from y to x

Example:



(1, 2)∃a path (1, 2)
(2, 1)∃a path (2, 3), (3, 1)
(2, 3)∃a path (2, 3)
(3, 2)∃a path (3, 1), (1, 2)
(3, 1)∃a path (3, 1)
(1, 3)∃a path (1, 2), (2, 3)
(1, 4), (4, 5), (5, 3)
(3, 4)∃a path (3, 1), (1, 4)
(4, 3)∃a path (4, 5), (5, 3)
(4, 5)∃a path (4, 5)
(5, 4)∃a path (5, 3), (3, 1), (1, 4) **G is strongly connected**
(5, 3)∃a path (5, 3)
(3, 5)∃a path (3, 1), (1, 4), (4, 5)
(1, 4)∃a path (1, 4)
(4, 1)∃a path (4, 5), (5, 3), (3, 1)
(1, 5)∃a path (1, 4), (4, 5)
(5, 1)∃a path (5, 3), (3, 1)
(2, 4)∃a path (2, 3), (3, 1), (1, 4)
(4, 2)∃a path (4, 5), (5, 3), (3, 1), (1, 2)
(2, 5)∃a path (2, 3), (3, 1), (1, 4), (4, 5)
(5, 2)∃a path (5, 3), (3, 1), (1, 2)

We note that ∀x, y∈X We can find a path from x to y and another from y to x ===> the graph G(X, U) is a strongly connected graph.

Definition of a strongly connected component

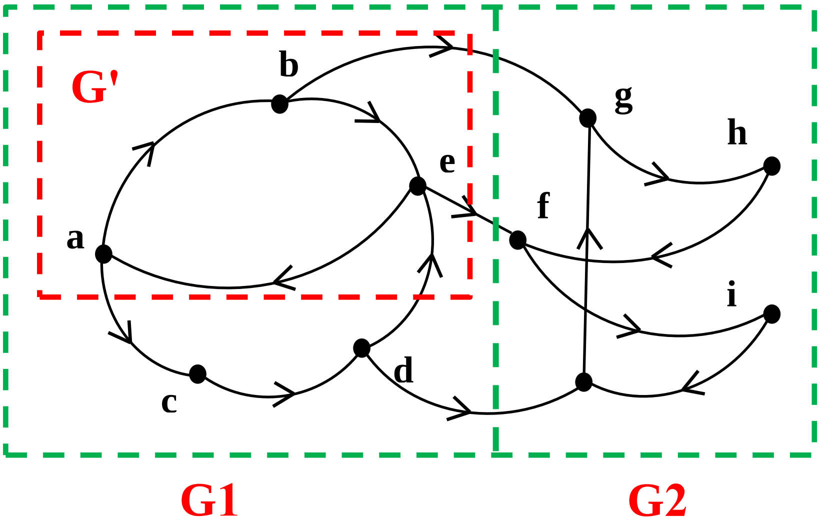
A strongly connected component of a graph G(X, U) is called a strongly connected subgraph G'(X', U').

Example:

Let G(X, U) be the following graph such that:

X={a, b, c, d, e, f, g, h, I, j}

U={(a, c), (c, d), (d, e), (e, f), (d, j), (b, g), (g, h), (h, f) , (f, i), (I, j), (j, g)}



G(X, U) is not strongly connected.

G'(X', U') is a subgraph of G.

X'={a, b, e}; U' = {(a, b), (b, e), (e, a)}

G' represents a strongly connected component (not maximal)

G1(X1, U1), G2(X2, U2) are two maximal strongly connected components.

Observations:

G(X, U) a graph, if ∀x, y∈X, there exists a chain between x and y we say that G(X, U) is simply connected.

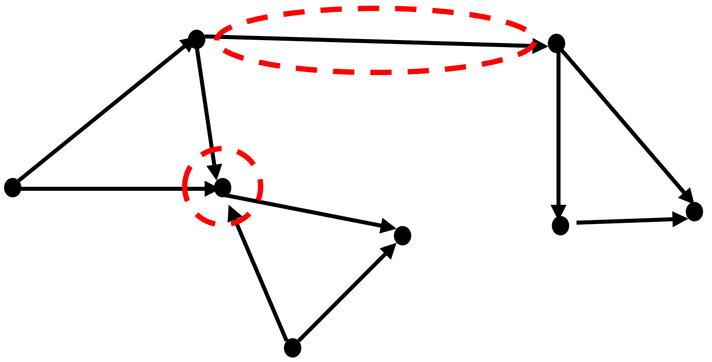
G(X, U) is a connected graph, if after eliminating more than k edges, G becomes unconnected we say that G is k-edge connected.

Definition of an articulation point

An articulation point of a graph G is a vertex whose deletion increases the number of connected components.

Definition of an isthmus:

Is an arc whose deletion increases the number of CCs.



Algorithm for finding a Simply Connected Component SCC of a vertex S

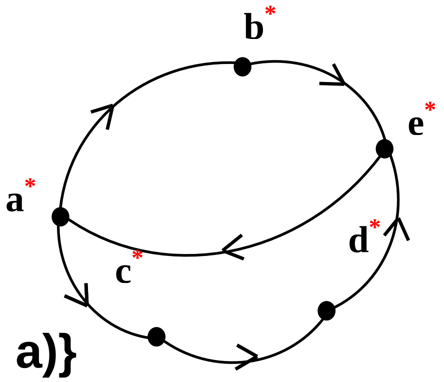
Let G(X, U) be a graph

(1) Mark the vertex S (by *)
(2) Mark any adjacent (successor/predecessor) vertex of an already marked vertex (by *)
(3) Repeat (2) until no more vertices can be marked.
(4) The vertices marked with (*) form the simply connected component of S

Example:

Let G(X, U) be a graph with:

X={a, b, c, d, e}



U = {(a, b), (a, c), (b, e), (c, d), (d, e), (e, a)}

Let's build the CSC of b

(1) Mark b (b*)
(2) Mark the adjacent vertices of b (successors and predecessors of b)
(3) No vertex remained unmarked.

{a, b, c, d, e} constitutes a simply connected component, i.e., G forms a SCC

Algorithm for finding a strongly connected component of a vertex S

Let G(X, U) be a graph

(1) Mark the vertex S with (+ and -)
(2) (a) Mark with (+) any successor (not yet marked +) from a vertex already marked (+)

(b) Mark with (-) any predecessor (not yet marked -) of a vertex already marked (-)

(3) Vertices marked with both + and – form a Str.CC containing S.

Example:

in the graph opposite

CFC(b)={b, a, e}