GRAPH THEORY - II

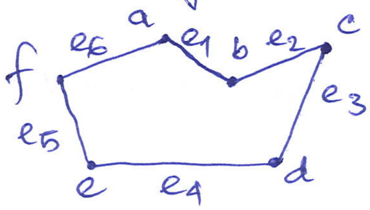
Connectivity in graphs!

Consider $n \in \mathbb{Z}^+ \cup \{0\}$

Path: A path of length n from vertex a to vertex b is a sequence of n edges $e_1, e_2, \ldots, e_n$ of the graph s.t.

e_1 is associated with $\{x_0, x_1\}$ $\quad x_0 = a$
e_2 " " " $\{x_1, x_2\}$ $\quad x_n = b$
$\vdots$
e_n " " " $\{x_{n-1}, x_n\}$

Simple path: A path is simple if it does not contain same edge more than once.



$a - b$ is a simple path

a, b, c, d
$e_1 \; e_2 \; e_3$

a, b, c, b is not a simple path.
$e_1 \; e_2 \; e_2$

a, b, c, d, e, f, a is circuit.
$e_1 \; e_2 \; e_3 \; e_4 \; e_5 \; e_6$

Circuit: A path is a circuit if $a = b$ and $n > 0$, i.e., a path of length greater than 0, s.t., the start and end vertices are the same. A circuit is also called a cycle in the graph.

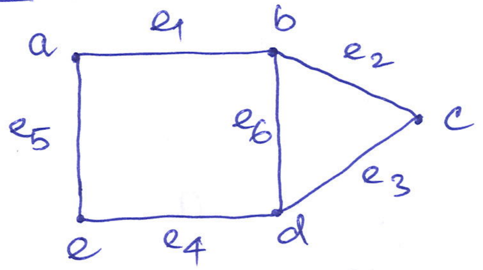
Simple circuit! A circuit is simple if it does not contain an edge more than once.

Walk! A walk is defined to be an alternating seq. of vertices and edges of a graph.

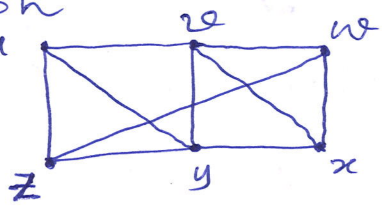
Closed walk: A walk that begins and ends at the same vertex.

<u>Trail</u>: A simple walk is called a trail.

<u>Ex</u>:



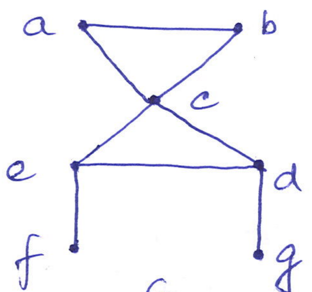
Find ~~dif~~ different properties of the graph



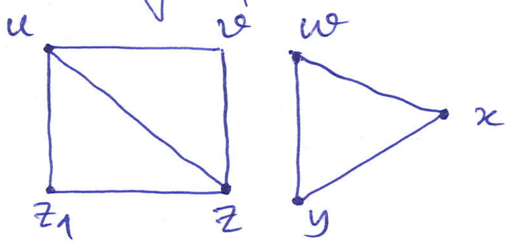
a, b, d, e, a is a circuit
a, b, c, d, e, a " "
a, e_1 b, e_2, c, e_3, d is a walk which is also a trail (simple walk).
b, e_2, c, e_3, d, e_6, b is a closed walk.

☒ For directed graphs, similar definitions will follow considering the directed edges.

<u>Connected graph</u>: An undirected graph is called connected if there is a path between every pair of distinct vertices of the graph.



G_1



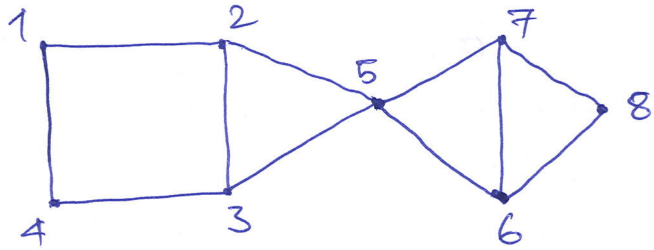
G_2

The graph G_1 is connected but not G_2.

<u>Theorem</u>: There is a simple path between every pair of distinct vertices of a connected undircted graph.

<u>Cut-set</u>: In a connected graph G, a cut-set is a set of edges whose removal from G leaves G disconnected, provided removal of no proper subset of these edges disconnects G.

Ex:

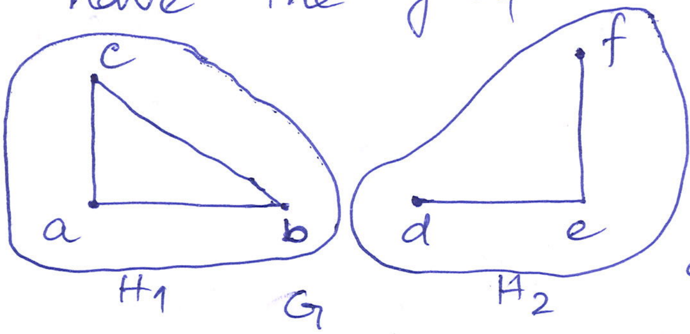


$\{(2,5),(3,5)\}$ is a cut-set.
$\{(1,2),(2,3),(3,5)\}$ is also a cut-set.
$\{(1,2),(2,3),(3,5),(2,5)\}$ is not a cut-set, since there is a proper subset $\{(1,2),(2,3),(3,5)\}$ whose removal disconnects the graph.

Connected components: A connected component of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G.

→ A connected component of a graph G is a maximal connected subgraph of G.

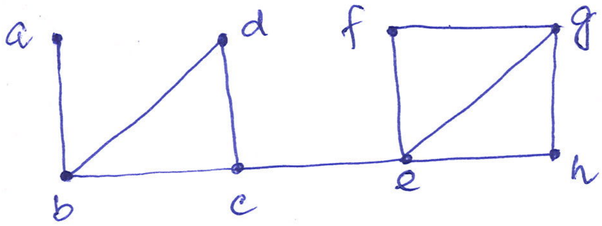
→ A graph that is not connected has two or more connected components that are disjoint and have the graph as their union.



H_1 and H_2 are connected components of G.
Also, H_1 and H_2 are disjoint and $H_1 \cup H_2 = G$.

☒ Sometimes removal of a vertex and all edges incident with it produces a subgraph with more connected components than in the original graph. These are called cut vertices.
So, the removal of a cut vertex from a connected graph produces a subgraph that is not connected.

Analogously, an edge whose removal produces a graph with more connected components than the original graph is called a cut edge or bridges.

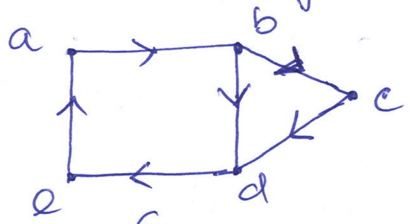


b, c, e are cut vertices, since removal of one of these vertices (and corresponding incident edges) disconnects the graph. The cut edges are $\{a, b\}$ and $\{c, e\}$. Removing either one of these edges disconnects the graph.

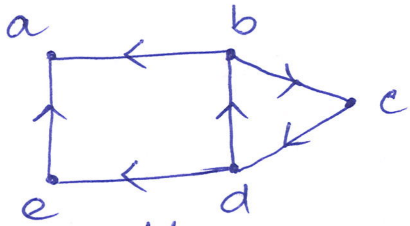
Connectedness in Directed graphs:

Strongly connected graphs: A directed graph is strongly connected if there is a path from a to b and from b to a whenever a and b are vertices in the graph.

Weakly connected graphs: A directed graph is weakly connected if there is a path between every two vertices in the corresponding undirected graph.



G

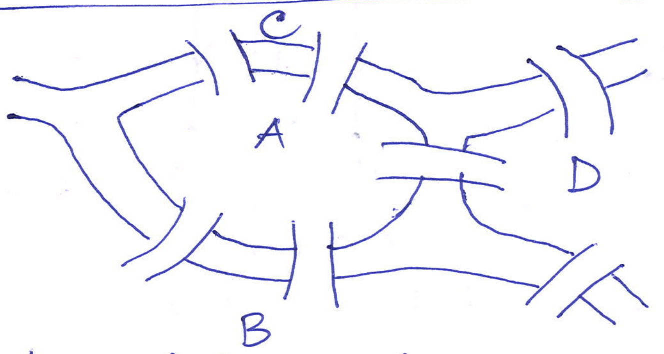


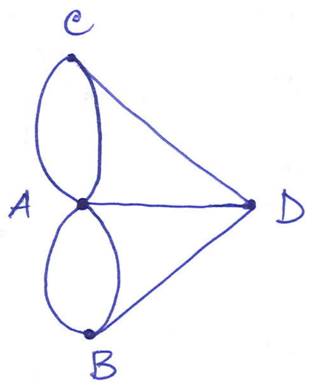
H

Verify that there is a path between any pair of vertices in G. Then G is strongly connected and hence weakly connected as well.

Note that the graph H is not strongly connected as from a to b path does not exist, whereas there is a from b to a, there is a path. But H will be weakly connected as the underlying undirected graph is connected.

Euler & Hamilton Paths





Pregel River (Königsberg, Russia)

whether it is possible to start at some location in town travel accross all the bridges without crossing any bridges twice and return to the starting point ??

OR

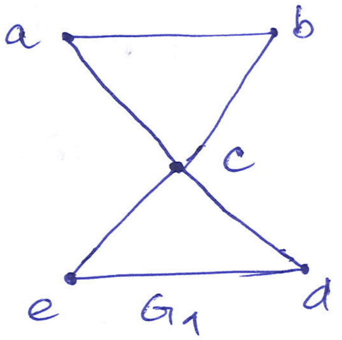
Is there a simple circuit in this multigraph that contains every edge ??

Euler circuit (E.C.) and Euler path (E.P.):

An E.C. in a graph G is a simple circuit containing every edge of G.

An E.P. in G is a simple path containing every edge of G.

Ex:

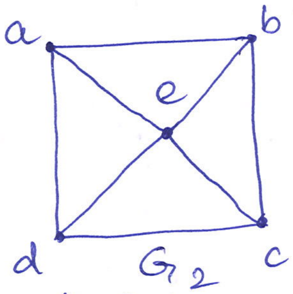


G_1

E.C.:- a, c, d, e, c, b, a

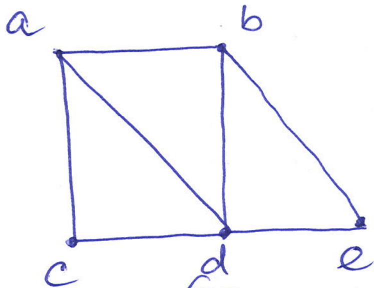
E.P.:- a, b, c, e, d, c, a

So, all E.P. are E.C. in this graph.



G_2

has no E.C. and E.P.



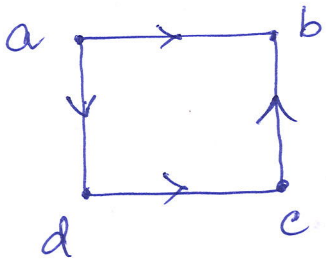
G_3

has no E.C.

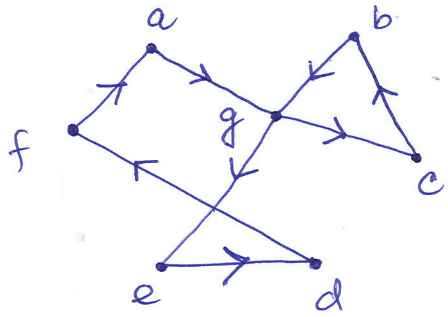
E.P.:- a, c, d, e, b, d, a, b.

Eulerian Graph: A graph that contains either an E.P. or an E.C. is called Eulerian Graph.

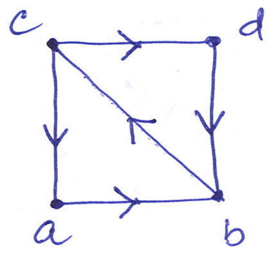
G_1 and G_3 are Eulerian graph but G_2 is not.



No. E.C.
and E.P.



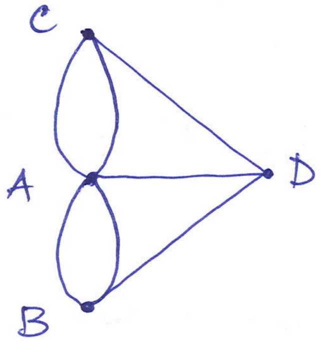
E.C.:- a, g, c, b, g
e, d, f, a



No E.C.
E.P.- c, a, b, c, d, b

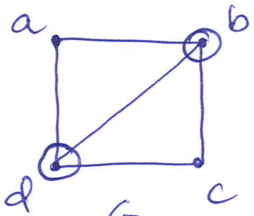
Theorem 1: A connected multigraph with at least 2 vertices has an E.C. iff (if and only if) each of its vertices has even deg.

Königsberg bridge problem

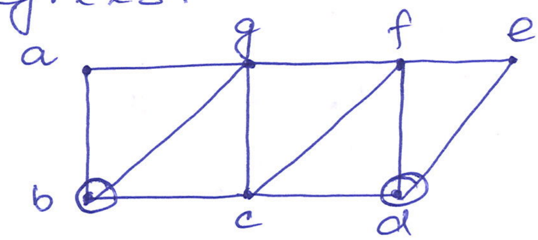


This graph has no E.C. as there is at least one vertex present in the graph with odd degree.

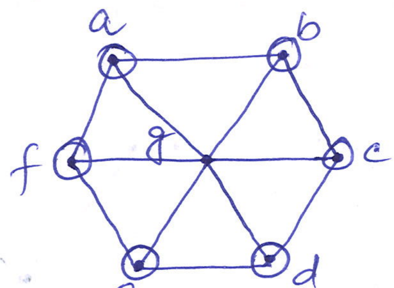
Theorem 2: A connected multigraph has an E.P. but not an E.C. iff it has exactly two vertices of odd degrees.



G_1



G_2



G_3

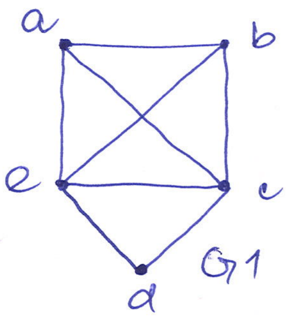
2 vertices b and d with odd degrees.
It has E.P.
d, a, b, c, d, b

2 vertices b and d with odd degress.
It has E.P.
b, a, g, f, e, d, c, g, b, c, f, d

6 vertices with odd degrees.
No E.P.

Hamilton Path ~~oo~~(H.P.) and Hamilton Circuit (H.C.):

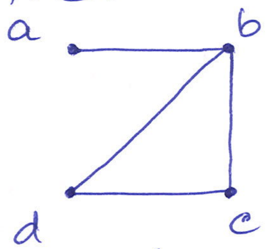
A simple path in a graph that passes through every vertex exactly once is called a H.P., and a simple circuit in the graph that passes through every vertex ~~ex~~actly once is called a H.C.



G1

H.C.:

a, b, c, d, e, a

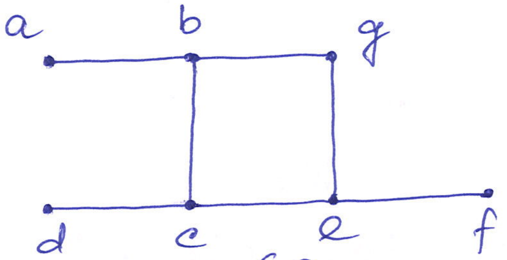


G2

NO H.C.

H.P.

a, b, c, d



G3

NO H.C. and H.P.

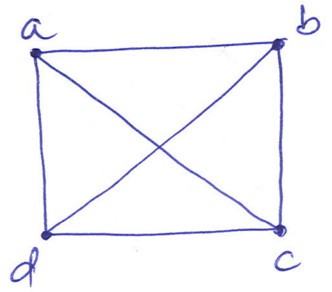
Hamiltonian Graph: A graph that contains either a H.P. or a H.C. is called a Hamiltonian graph.

Dirac's Theorem: If G is a simple graph with n (≥ 3) vertices such that the deg. of every vertex in G is at least $n/2$, the G has a H.C.

Ore's Theorem: If G is a simple graph with n (≥ 3) vertices such that $\deg(u) + \deg(v) \geq n$ for every pair of nonadjacent vertices u and v in G, then G has a H.C.

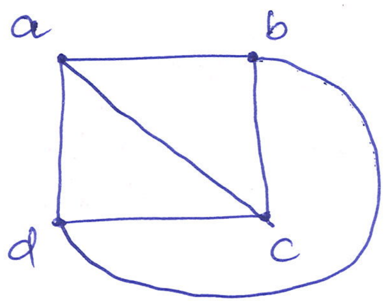
Note that, G_2 and G_3 both do not satisfy Dirac's Theorem and Ore's Theorem and hence do not contain a H.C.

Planner graph: A graph is planner if it can be drawn in the plane without any edges corossing. A graph which is not a planner is called non-planner graph



K_4

$\cong$



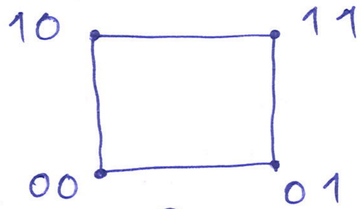
Planner graph.

K_4 has two edges corossing each other, but it can drawn without crossing.

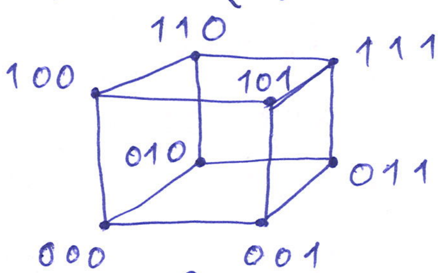
K_5, $K_{3,3}$ are not planner graphs.

Hypercube or n-cube: A graph that has vertices representing the 2^n bit strings of length n, is called Hypercube or n-cube Q_n.

Q_1

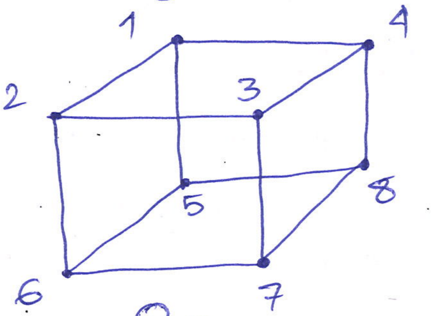


Q_2

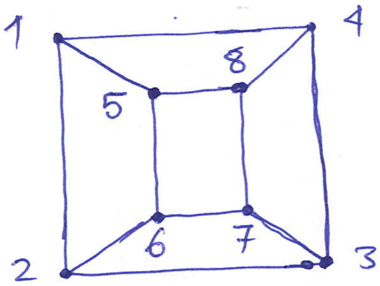


Q_3

☑ Is Q_3 planner graph ??



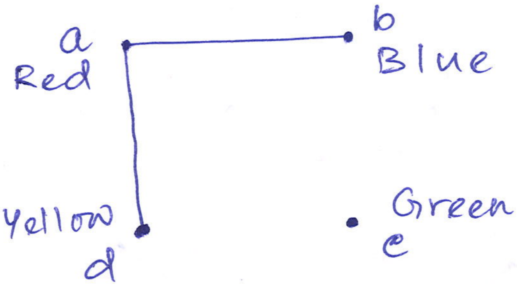
Q_3



Planner representation of Q_3.

Graph Colouring:

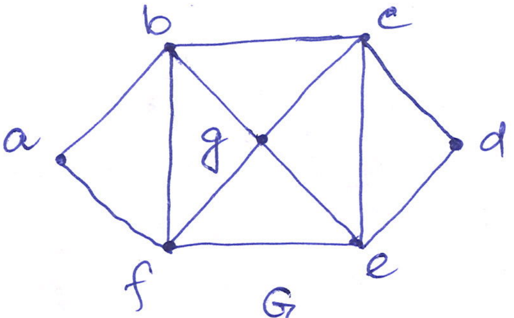
A colouring of a simple graph is the assignment of colours to each vertex of the graph so that no two adjacent vertices are assigned with the same colour.



A graph can be coloured by assigning different colours to each of its vertices. However, using least no. of colours to assign to the vertices is an interesting problem.

Chromatic Number:

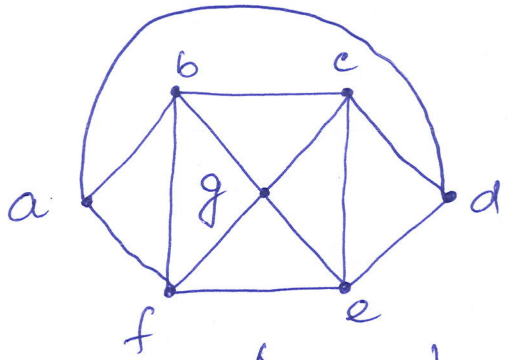
The chromatic no. of a graph is the least no. of colours needed for a colouring of the graph and it is denoted by $\chi(G)$.



G

Red - $\{a, g, d\}$
Blue - $\{b, e\}$
Green - $\{f, c\}$

$\chi(G) = 3$



Red - $\{a, g\}$
Blue - $\{b, e\}$
Green - $\{f, c\}$
Yellow - $\{d\}$

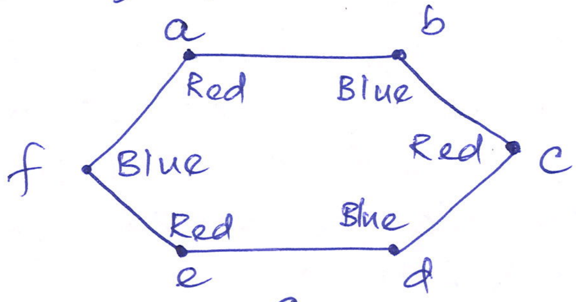
$\chi(G) = 4$

☒ What is the chromatic no. of K_n??

→ Since any pair of vertices in K_n is adjacent, so each vertex will have different colours. ~~See~~ Hence, $\chi(K_n) = n$.

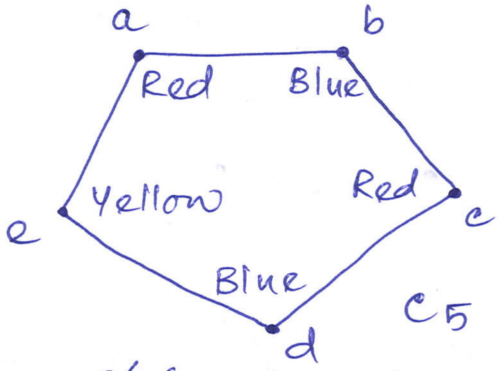
<u>Four Colour Theorem</u>: The chromatic no. of a planner graph is not greater than four.

→ Consider the cycle graph with 6 vertices C_6. Note that C_6 is planner graph and hence $\chi(C_6) \leq 4$.



C_6

$\chi(C_6) = 2$



C_5

$\chi(C_5) = 3$.

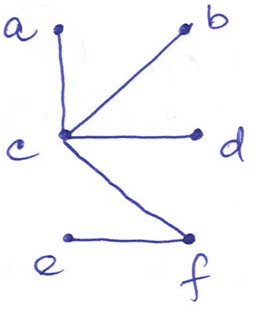
TREES

Trees are particularly useful in computer science, where they are employed in a wide range of algorithms. Trees can be used for data transmission, storage, study games and determine winning strategies. Constructing these models can help determine the computational complexity of algorithms based on a sequence of decisions, such as sorting algorithms.

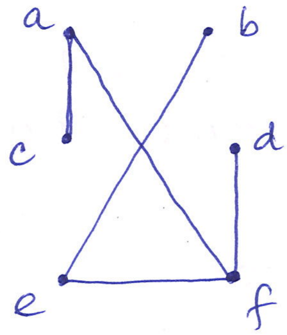
- A tree is a connected undirected graph with no simple circuits.

→ Becuse a tree cannot have a simple circuit, a tree cannot contain multiple edges or loops. Therefore any tree must be a simple graph.

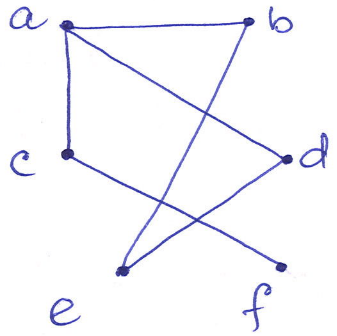
<u>Theorem</u>: An undirected graph is a tree iff (if and only if) there is a unique simple path between any two of its vertices.



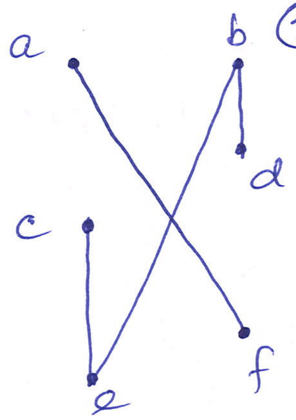
Tree



Tree

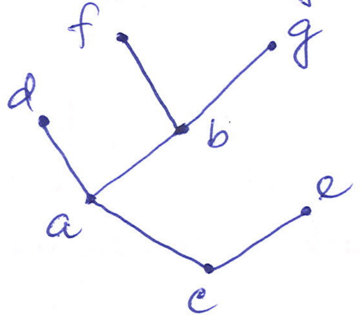


Not a tree
contains a simple circuit
a, b, e, d, a

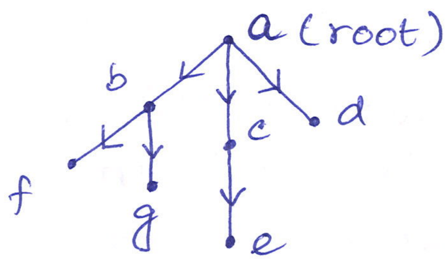


Not a tree
not connected

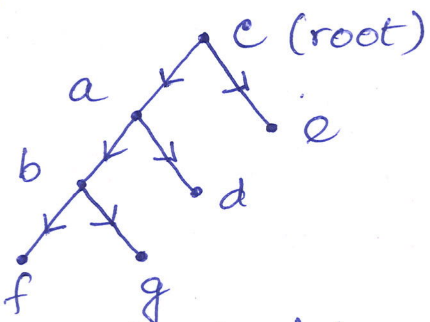
Rooted Tree: A rooted tree is a tree in which one vertex has been designated as the root and every edge is directed away from the root.



Tree

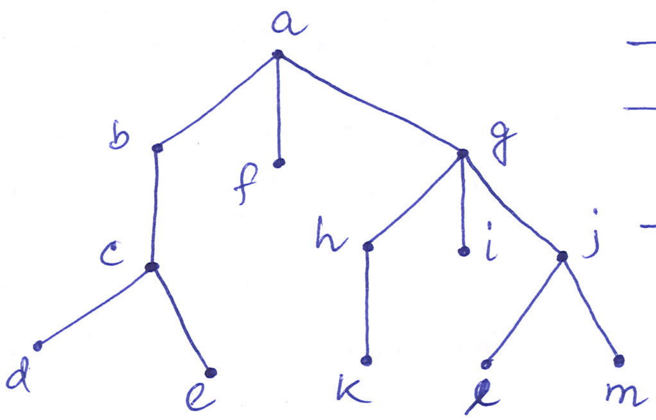


Rooted tree



Rooted tree

Ex:



— Parent of 'c' is 'b'.

— The children of 'g' are 'h', 'i', 'j'

— The ancestors of 'e' are 'c', 'b' & 'a'

— The decendents of 'b' are 'c', 'd', 'e'

The treminology for trees has botanical and genealogical origin.

Suppose, T is a rooted tree.

⟶ If v is a vertex in T other than the root, the <u>parent</u> of v is the unique vertex u, such that there is a directed edge from u to v.

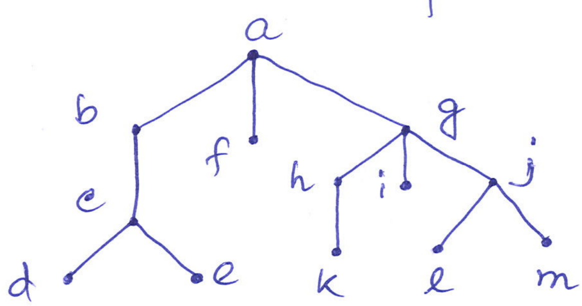
→ When u is the parent of v, v is a <u>child</u> of u. Vertices with the same parent are called <u>siblings</u>.

→ The <u>ancestors</u> of a vertex other than the root are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root (i.e., its parent, its parent's parent, and so on, until the root is reached).

→ The <u>descendants</u> of a vertex v are those vertices that have v as an ancestor.

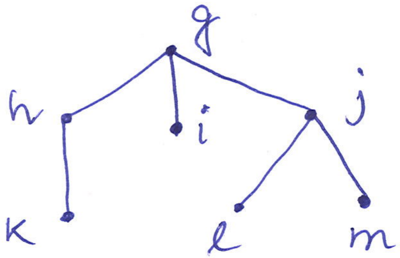
→ A vertex of a tree is called a <u>leaf</u> if it has no children.

→ Vertices that have children are called <u>internal vertices</u>. The root is an internal vertex unless it is the only vertex in the graph, in which it is a leaf.



– The internal vertices are 'a', 'b', 'c', 'g', 'h', 'j'

– The leaves are 'd', 'e', 'f', 'i', 'k', 'l', 'm'



The subtree rooted at g of the above tree is the subgraph of the tree consisting of g and its descendants and all edges incident to these descendants.

☒ A rooted tree is called an <u>m-ary tree</u> if every internal vertex has no more than m children.

☒ The tree is called a <u>full m-ary tree</u> if every internal vertex has exactly m children.

→ An m-ary tree with $m = 2$ is called a <u>binary tree</u>.

<u>Example</u>

— T_1 is a full 2-ary tree because each of its internal vertices has two children.

— T_2 is a full 3-ary tree.

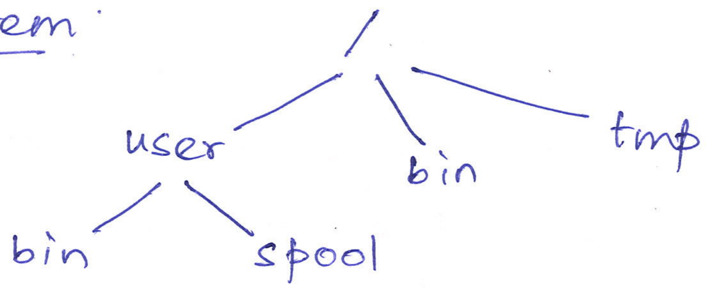
— T_3 is not a full m-ary tree as some internal vertices have 2 children and some have 3.

Trees as Models

1. <u>Saturated Hydrocarbons & Trees</u>

In 1857, Cayley discovered trees, while trying to enumerate the isomers of compounds of the form C_nH_{2n+2}.

2. <u>Computer file system</u>

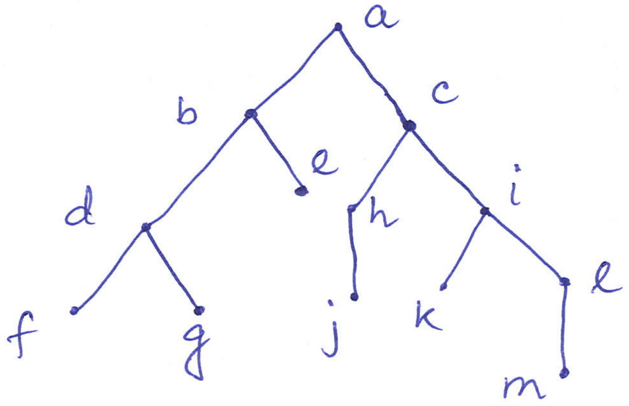


Ordered Tree: An ordered rooted tree is a rooted tree where the children of each internal vertex are in order from left to right.

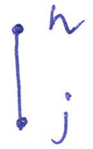
In an ordered binary tree, if an internal vertex has two children, the first child is called the left child and the second is right child.

The tree rooted at the left child of a vertex is the left subtree of the vertex.
Similarly, we can define the right subtree of the vertex.

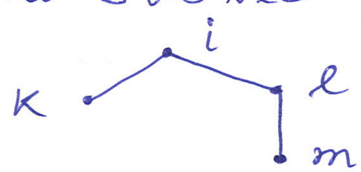
Ex:



- left child of d is f and right child of d is g.
- left subtree of c



- right subtree of c



Properties of Trees

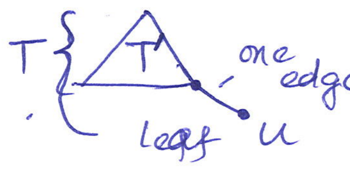
Theorem: A tree with n vertices has $n-1$ edges

Proof by induction

Base case $n=1$: A tree with one vertex has 0 edges. So the statement is true for $n=1$.

Suppose the statement is true for $n = k (\geq 0)$
i.e., Every tree with k vertices has $k-1$ edges

Now consider a tree T with $k+1$ vertices and a leaf u. If we remove u from T then T′ will be a tree with k vertices and hence will have $k-1$ edges.
Hence, T has $(k-1)+1 = k$ edges.



Theorem: A full m-ary tree with i internal vertices contains $n = mi + 1$ vertices.

Theorem: Let G be a graph with $n > 1$ vertices. Then the following are equivalent:

i) G is a tree,
ii) G is a cycle (circuit) - free and has $n-1$ edges,
iii) G is connected and has $n-1$ edges.

Theorem: A full m-ary tree with

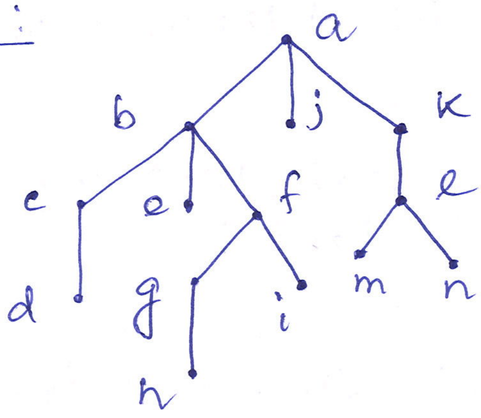
i) n vertices, has $i = \dfrac{n-1}{m}$ internal vertices and $l = [(m-1)n + 1]/m$ leaves.

ii) i internal vertices, has $n = mi + 1$ vertices and $l = (m-1)i + 1$ leaves.

iii) l leaves, has $n = (ml - 1)/(m-1)$ vertices and $i = (l+1)/(m-1)$ internal vertices.

The level of a vertex v in a rooted tree is the length of the unique path from the root to this vertex.
The level of the root is defined to be zero.
The height of a rooted tree is the maximum of the levels of the vertices, i.e, the height is the length of the longest path from the root to any vertex.

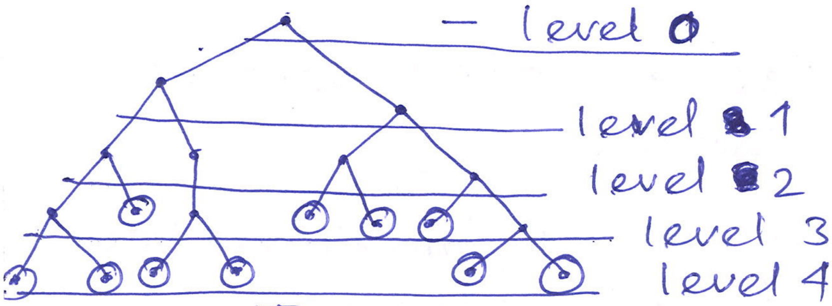
Ex :



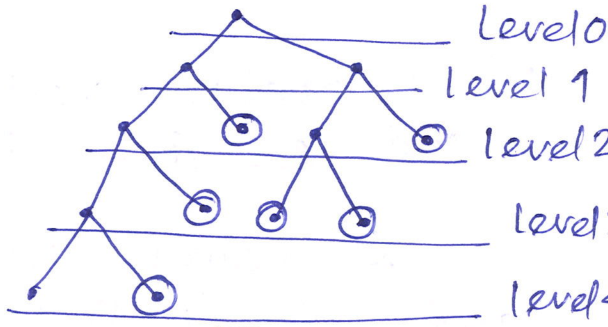
- root a is at level 0
- vertices b, j and k are at level 1
- c, e, f, l are at level 2.
- d, g, i, m, n are at level 3.
- vertex h is at level 4.

- The largest level of any vertex is 4, so the tree has height 4.

A rooted tree of height h is balanced if all leaves are at level h or at level h − 1.



T_1



T_2

T_1 is balanced.

all the leaves are at either level 3 or level 4.

T_2 is not balanced

leaves at level 2, 3 and 4.

<u>Theorem</u>: There are at most m^h leaves in an m-ary tree of height h.