Trees and other Special Classes of Graphs

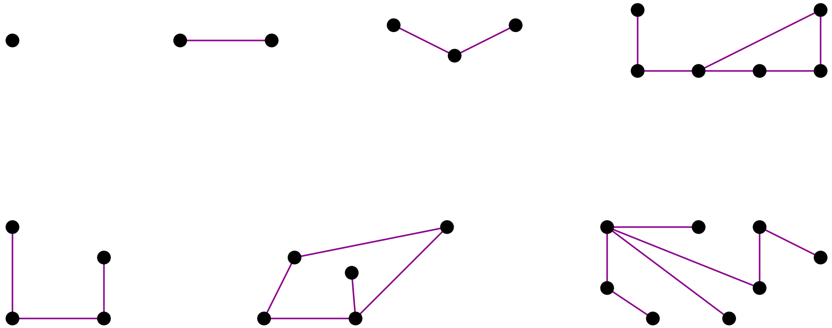
- Special Classes of Graphs
  - Complete Graphs, Path, Cycle, Star, Wheel, n-Cubes
- Bipartite Graphs
- Trees
  - Characterization of Trees
  - Minimum Spanning Tree
  - Rooted Trees

Imdad ullah Khan

Trees

A **tree** is a **connected** graph **with no cycles**

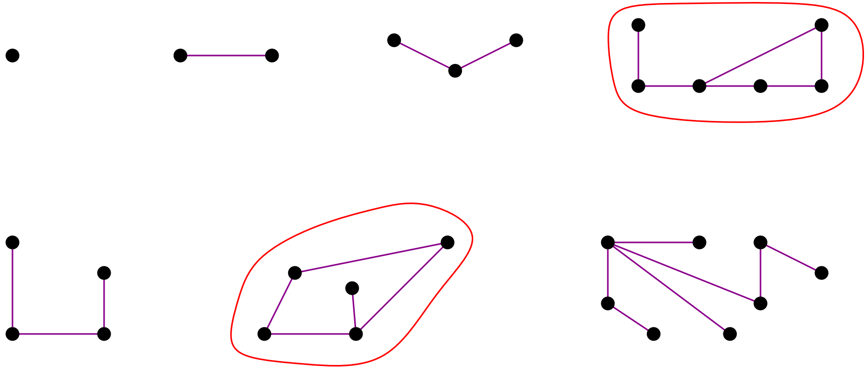
ICP 15-09 Which one of the following is not a tree?



Trees

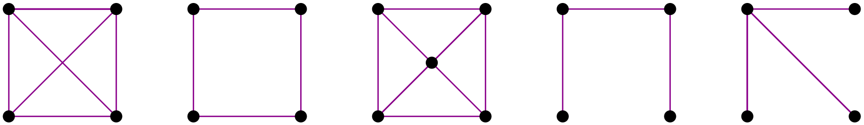
A **tree** is a **connected** graph **with no cycles**

ICP 15-09 Which one of the following is not a tree?



A **tree** is a **connected** graph **with no cycles**

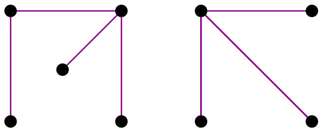
$\boxed{\textbf{ICP 15-10}}$ Which one of the following is a tree?



K_4 C_4 W_4 P_4 S_4

Forest

A **forest** is a ~~**connected**~~ graph **with no cycles**



$\boxed{\textbf{ICP 15-11}}$ How many components are there in a forest ?

$\boxed{\textbf{ICP 15-12}}$ Can a connected component of a forest have cycles ?

$\boxed{\textbf{ICP 15-13}}$ Each connected component of a forest is a tree. ▷ True/False

Characterizations of Trees

A **tree** is a **connected** graph **with no cycles**

Theorem

A graph is a tree if and only if there is a unique path between any two vertices

Proof: G is a tree $\implies$ unique path b/w any u and v

Let u and v have two "different" paths b/w them

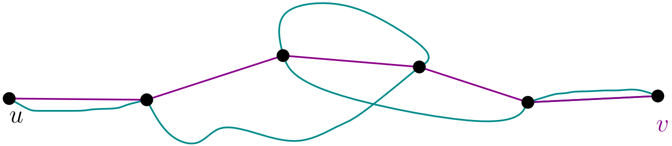
A **tree** is a **connected** graph **with no cycles**

Theorem

A graph is a tree if and only if there is a unique path between any two vertices

Proof: G is a tree $\implies$ unique path b/w any u and v

Let u and v have two “different” paths b/w them



This creates a cycle

Characterizations of Trees

A **tree** is a **connected** graph **with no cycles**

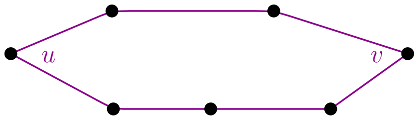
Theorem

A graph is a tree if and only if there is a unique path between any two vertices

Proof: unique path b/w any u and v $\implies$ G is a tree

The graph is connected, as (unique) paths exist

It cannot have cycles. If cycle exists, then



Characterizations of Trees

A **tree** is a **connected** graph **with no cycles**

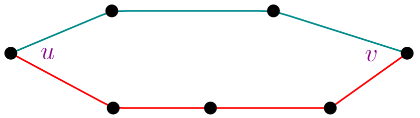
Theorem

A graph is a tree if and only if there is a unique path between any two vertices

Proof: unique path b/w any u and v $\implies$ G is a tree

The graph is connected, as (unique) paths exist

It cannot have cycles. If cycle exists, then



we get at least two paths

A **tree** is a **connected** graph **with no cycles**

Theorem

A graph is a tree if and only if there is a unique path between any two vertices

ICP 15-14 Give a formal proof of this theorem.

Characterizations of Trees

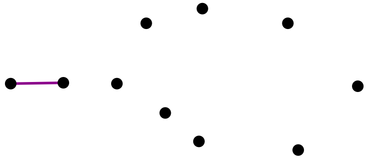
A **leaf** in a graph is a vertex with degree 1

Theorem

Any tree has at least one leaf

Proof: If there is no leaf i.e. every vertex has degree ≥ 2

Start a walk from any vertex. In each step take an unvisited edge



Characterizations of Trees

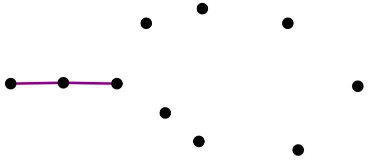
A **leaf** in a graph is a vertex with degree 1

Theorem

Any tree has at least one leaf

Proof: If there is no leaf i.e. every vertex has degree ≥ 2

Start a walk from any vertex. In each step take an unvisited edge



Characterizations of Trees

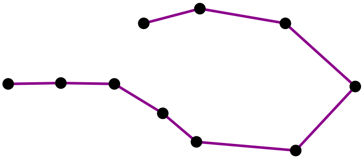
A **leaf** in a graph is a vertex with degree 1

Theorem

Any tree has at least one leaf

Proof: If there is no leaf i.e. every vertex has degree ≥ 2

Start a walk from any vertex. In each step take an unvisited edge



Characterizations of Trees

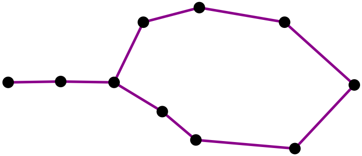
A **leaf** in a graph is a vertex with degree 1

Theorem

Any tree has at least one leaf

Proof: If there is no leaf i.e. every vertex has degree ≥ 2

Start a walk from any vertex. In each step take an unvisited edge



cannot get stuck unless a cycle exists

Characterizations of Trees

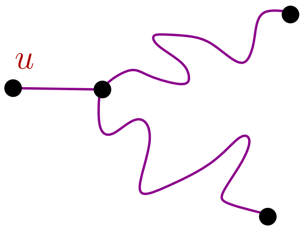
A **leaf** in a graph is a vertex with degree 1

Any tree has at least one leaf.

Theorem

Any tree on n vertices has $n-1$ edges

Inductive Proof: Remove a leaf u



Characterizations of Trees

A **leaf** in a graph is a vertex with degree 1

Any tree has at least one leaf.

Theorem

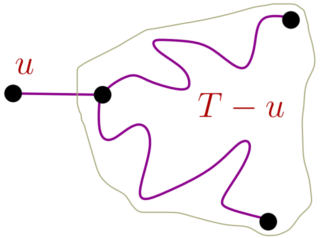
Any tree on n vertices has $n - 1$ edges

Inductive Proof: Remove a leaf u

$T - u$ **is a tree**

$T - u$ has $n - 2$ edges

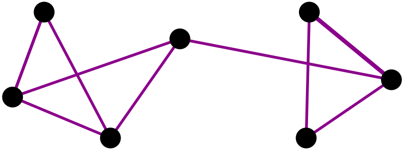
So, T has $n - 1$ edges



Characterizations of Trees

An edge is **cut edge** if removing it disconnects the graph

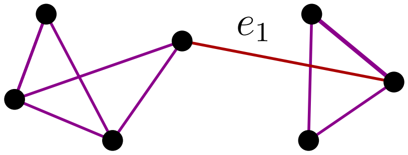
Recall definition of cut vertices and k-connected graphs



Characterizations of Trees

An edge is cut edge if removing it disconnects the graph

Recall definition of cut vertices and k-connected graphs



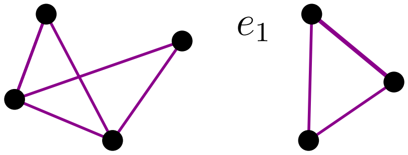
e_1 is a cut edge

Removing e_1 disconnects the graph

Characterizations of Trees

An edge is cut edge if removing it disconnects the graph

Recall definition of cut vertices and k-connected graphs



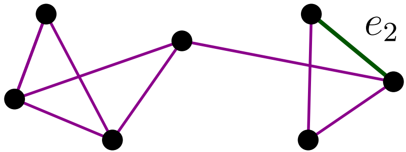
e_1 is a cut edge

Removing e_1 disconnects the graph

Characterizations of Trees

An edge is cut edge if removing it disconnects the graph

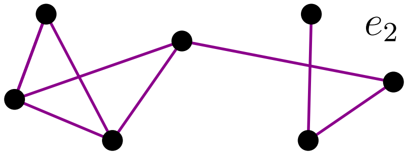
Recall definition of cut vertices and k-connected graphs



e_2 is not a cut edge

An edge is cut edge if removing it disconnects the graph

Recall definition of cut vertices and k-connected graphs



e_2 is not a cut edge

Graph is connected even after removing e_2

Characterizations of Trees

Definition

A connected graph is 2-edge connected iff it has no cut edge

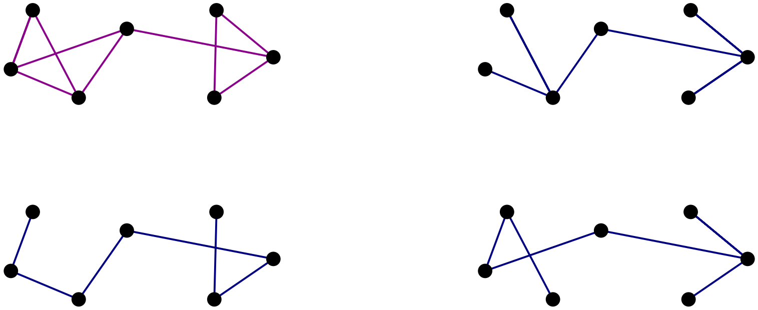
Theorem

An edge is not a cut-edge iff it is on a cycle

A tree is a connected graph with every edge a cut-edge

Spanning Tree

A **spanning tree** of a graph is a spanning subgraph that is a tree



There can be many spanning trees

Spanning Tree

Theorem

A graph G is connected iff G has a spanning tree

Proof: T is a spanning tree of G $\implies$ G is connected

T contains every vertex of G $\triangleright$ T is spanning

T is connected $\triangleright$ T is a tree

So a path between every u and v exists in T

The same path also exists in G $\triangleright$ $T \subseteq G$

Hence, G is connected

Spanning Tree

Theorem

A graph G is connected iff G has a spanning tree

Proof: G is connected $\implies$ there is a spanning tree T of G

If G is a tree, we are done

else G must have cycle

take one such cycle

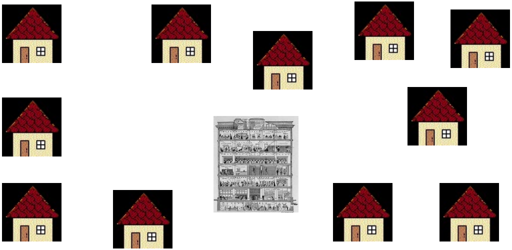
remove an edge from it

repeat until no cycle left

Remaining graph is acyclic $\because$ we deleted all cycles

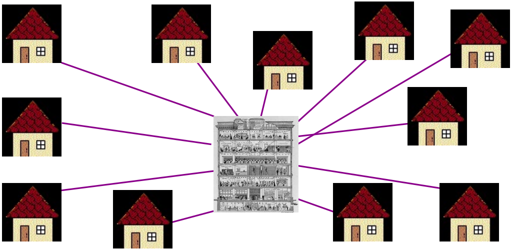
It is connected, if u, v are connected through a removed edge there was an alternative path

Optimal connection between points



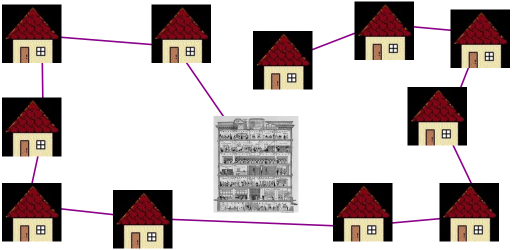
- Layout a telephone network so as every user is connected to all others
- Goal is to use as little wire as possible
- The network should not have any cycle (should be a tree)

Optimal connection between points



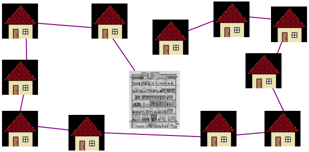
- Layout a telephone network so as every user is connected to all others
- Goal is to use as little wire as possible
- The network should not have any cycle (should be a tree)
- Naive approach (star network) may use a lot of wires

Optimal connection between points



- Layout a telephone network so as every user is connected to all others
- Goal is to use as little wire as possible
- The network should not have any cycle (should be a tree)
- Naive approach (star network) may use a lot of wires
- Many possible solutions

Minimum Spanning Tree



- Layout a telephone network so as every user is connected to all others
- Goal is to use as little wire as possible
- In this problem all pairwise connections are possible
- The underlying graph is a complete graph
- Weight of edges are lengths of physical paths between nodes

- There could be restrictions on possible edges (not complete graphs)
- Weight of edges could be arbitrary

Minimum Spanning Tree

Given a weighted graph $G = (V, E, c)$: c : cost/weight function: $c : E \to \mathbb{R}$

Find a spanning tree, such that its total weight is minimum (among all the spanning trees)

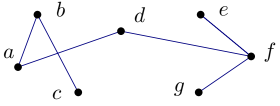
Weight of a tree is the sum of weights of all edges $\sum_{e \in T} c(e)$

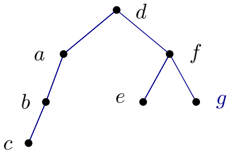
Prim's algorithm, Kruskal algorithm

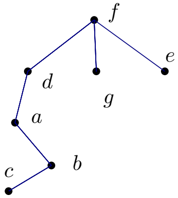
Rooted Tree

Designate one vertex as root of the tree

Assign direction to each edge away from the root





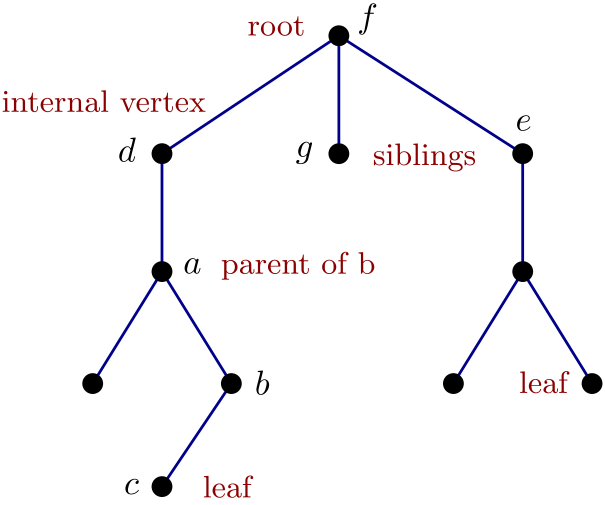


Consider edges as strings of unit lengths connecting balls (vertices)

Rooting a tree means pulling one ball (root) up - other balls will hang below

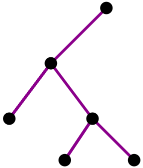
Different roots produce different rooted trees

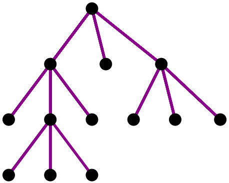
Rooted Tree: Terminology

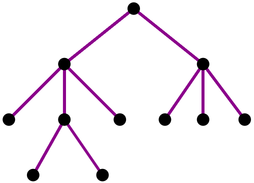


m-ary trees

A rooted tree is called an m-ary tree if every internal vertex has no more than m children.



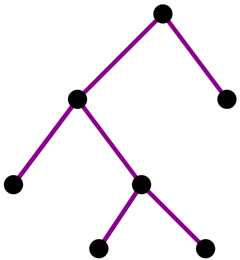


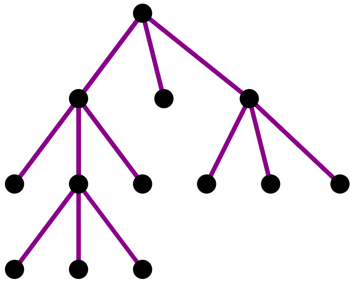


An m-ary tree with $m = 2$ is called a binary tree

m-ary trees

A rooted tree is called a full m-ary tree if every internal vertex has exactly m children





m-ary trees

Theorem

A full m-ary tree with k internal vertices has $n = mk + 1$ vertices

Proof: Two types of vertices in a tree

Type-1: Children of someone

Type-2: Not children of anyone (only root)

Every internal vertex has exactly m children, so total type-1 are mk

m-ary trees

Theorem

A full m-ary tree with

1. n *vertices has* $i = (n-1)/m$ *internal vertices and* $\ell = [(m-1)n+1]/m$ *leaves*
2. i *internal vertices has* $n = mi + 1$ *vertices and* $\ell = (m-1)i + 1$ *leaves*
3. ℓ *leaves has* $n = (ml - 1)/(m-1)$ *vertices and* $i = (\ell - 1)/(m-1)$ *internal vertices*

Proof: Left as exercise. Please try it yourself, it is in the book

First restate and prove for $m = 2$, then for general m