COMP108
Data Structures and Algorithms

Trees (Part I)

Professor Prudence Wong

pwong@liverpool.ac.uk

2022-23

Outline

Trees

- Basic terminologies
- Binary trees and traversals

Learning outcome:

- Be able to tell what a tree is
- Be able to describe different algorithms to traverse a binary tree

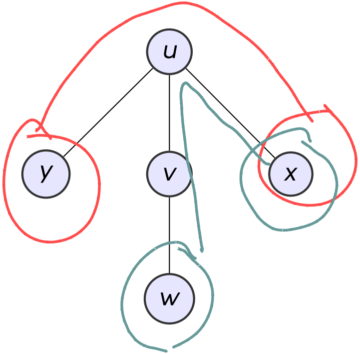
Tree

- Linked list is data structure where elements are arranged in a linear manner
- What if there are branches?
- Linked list is a special type of tree

Tree

Definition

A tree $T = (V, E)$ consists of a set of vertices V and a set of edges E such that for any pair of vertices $u, v \in V$, there is exactly one path between u and v.

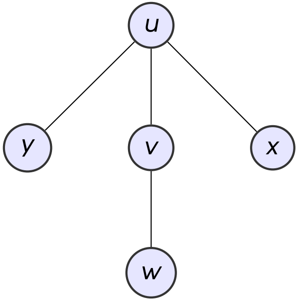


A tree

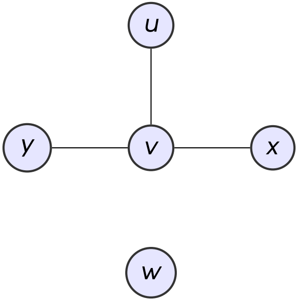
Tree

Definition

A tree $T = (V, E)$ consists of a set of vertices V and a set of edges E such that for any pair of vertices $u, v \in V$, there is exactly one path between u and v.



A tree



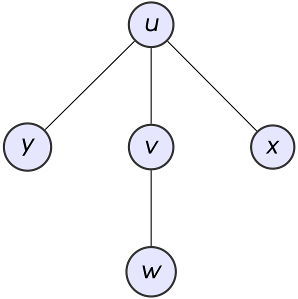
Not a tree

Tree

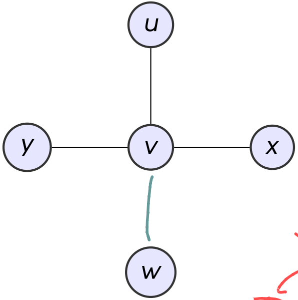
a tree is that there is no cycle (WRONG)

Definition

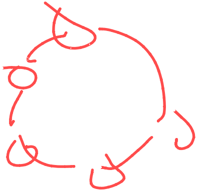
A tree $T = (V, E)$ consists of a set of vertices V and a set of edges E such that for any pair of vertices $u, v \in V$, there is exactly one path between u and v.

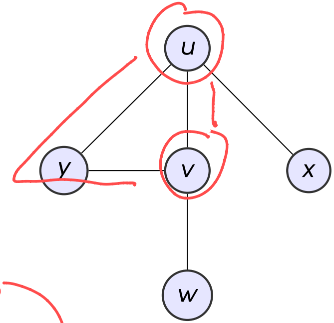


A tree



Not a tree





Not a tree

Tree

Equivalent statements

1. There is exactly one path between any two vertices in T
2. T is connected (there is at least one path between any two vertices in T) and there is no cycle (acyclic) in T
3. T is connected and removal of one edge disconnects T
4. T is acyclic and adding one edge creates a cycle
5. T is connected and $m = n - 1$ (where $n \equiv |V|$, $m \equiv |E|$)

Tree

Equivalent statements

1. There is exactly one path between any two vertices in T
2. T is connected (there is at least one path between any two vertices in T) and there is no cycle (acyclic) in T

   connected: at least one path; no cycle: at most one path
3. T is connected and removal of one edge disconnects T
4. T is acyclic and adding one edge creates a cycle
5. T is connected and $m = n - 1$ (where $n \equiv |V|$, $m \equiv |E|$)

Tree

Equivalent statements

1. There is exactly one path between any two vertices in T

2. T is connected (there is at least one path between any two vertices in T) and there is no cycle (acyclic) in T

   connected: at least one path; no cycle: at most one path

3. T is connected and removal of one edge disconnects T

   removal of an edge (u, v) disconnects at least u & v due to (2)

4. T is acyclic and adding one edge creates a cycle

5. T is connected and $m = n - 1$ (where $n \equiv |V|$, $m \equiv |E|$)

Tree

Equivalent statements

1. There is exactly one path between any two vertices in T

2. T is connected (there is at least one path between any two vertices in T) and there is no cycle (acyclic) in T

   connected: at least one path; no cycle: at most one path

3. T is connected and removal of one edge disconnects T

   removal of an edge (u, v) disconnects at least u & v due to (2)

4. T is acyclic and adding one edge creates a cycle

   adding (u, v) creates one more path between u & v $\implies$ a cycle

5. T is connected and $m = n - 1$ (where $n \equiv |V|$, $m \equiv |E|$)

Tree

Equivalent statements

1. There is exactly one path between any two vertices in T

2. T is connected (there is at least one path between any two vertices in T) and there is no cycle (acyclic) in T

   connected: at least one path; no cycle: at most one path

3. T is connected and removal of one edge disconnects T

   removal of an edge (u, v) disconnects at least u & v due to (2)

4. T is acyclic and adding one edge creates a cycle

   adding (u, v) creates one more path between u & v $\implies$ a cycle

5. T is connected and $m = n - 1$ (where $n \equiv |V|$, $m \equiv |E|$)

   proof by induction

Tree

Lemma

$P(n)$: *If a tree T has n vertices and m edges, then $m = n - 1$.*

Proof.

By induction on the number of vertices.

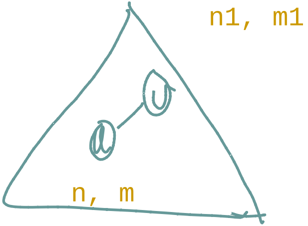
Base case: A tree with single vertex does not have an edge. $m=0,\ n=1 \Rightarrow m=n-1$

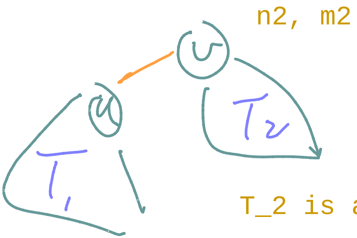
Induction step: $P(n-1) \implies P(n)$ for $n > 1$?

```
hypothesis:
m1 = n1 - 1
m2 = n2 - 1

n = n1 + n2
m = m1 + m2 + 1

m = m1 + m2 + 1
  = (n1-1) + (n2-1) + 1
  = n1 + n2 - 2 + 1 = n1+n2-1 = n-1
```





T_2 is a tree

T_1 is a tree

Tree

Lemma

$P(n)$: If a tree T has n vertices and m edges, then $m = n - 1$.

Proof.

By induction on the number of vertices.

Base case: A tree with single vertex does not have an edge.

Induction step: $P(n-1) \implies P(n)$ for $n > 1$?

- Remove an edge from the tree T. By (3), T becomes disconnected. Two connected components T_1 and T_2 are obtained, neither contains a cycle (otherwise the cycle is also present in T).

Tree

Lemma

$P(n)$: *If a tree T has n vertices and m edges, then $m = n - 1$.*

Proof.

By induction on the number of vertices.

Base case: A tree with single vertex does not have an edge.

Induction step: $P(n-1) \implies P(n)$ for $n > 1$?

- Remove an edge from the tree T. By (3), T becomes disconnected. Two connected components T_1 and T_2 are obtained, neither contains a cycle (otherwise the cycle is also present in T).
- Therefore, both T_1 and T_2 are trees.
  Let n_1 and n_2 be the number of vertices in T_1 and T_2.
  $\implies n_1 + n_2 = n$

Tree

Lemma

$P(n)$: *If a tree T has n vertices and m edges, then $m = n - 1$.*

Proof.

By induction on the number of vertices.

Base case: A tree with single vertex does not have an edge.

Induction step: $P(n-1) \implies P(n)$ for $n > 1$?

- Remove an edge from the tree T. By (3), T becomes disconnected. Two connected components T_1 and T_2 are obtained, neither contains a cycle (otherwise the cycle is also present in T).
- Therefore, both T_1 and T_2 are trees.
  Let n_1 and n_2 be the number of vertices in T_1 and T_2.
  $\implies n_1 + n_2 = n$
- By the induction hypothesis, T_1 and T_2 contains $n_1 - 1$ and $n_2 - 1$ edges, respectively.

Tree

Lemma

$P(n)$: If a tree T has n vertices and m edges, then $m = n - 1$.

Proof.

By induction on the number of vertices.

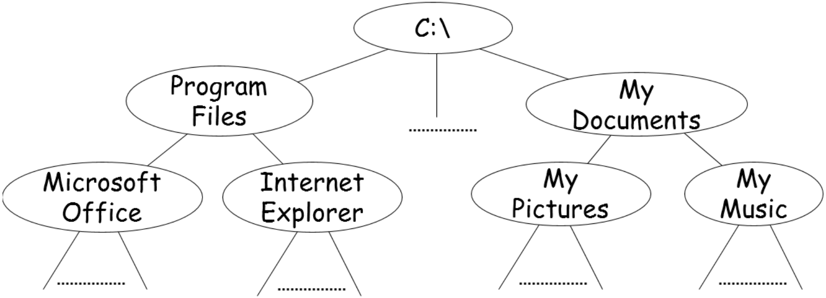
Base case: A tree with single vertex does not have an edge.

Induction step: $P(n-1) \implies P(n)$ for $n > 1$?

- Remove an edge from the tree T. By (3), T becomes disconnected. Two connected components T_1 and T_2 are obtained, neither contains a cycle (otherwise the cycle is also present in T).
- Therefore, both T_1 and T_2 are trees.
  Let n_1 and n_2 be the number of vertices in T_1 and T_2.
  $\implies n_1 + n_2 = n$
- By the induction hypothesis, T_1 and T_2 contains $n_1 - 1$ and $n_2 - 1$ edges, respectively.
- Hence, T contains $(n_1 - 1) + (n_2 - 1) + 1 = n - 1$ edges.

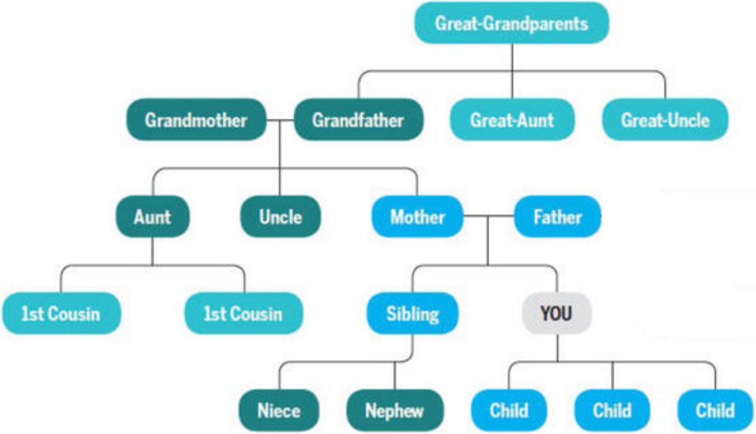
Rooted trees

Tree with hierarchical structure, e.g., folder structure of file system

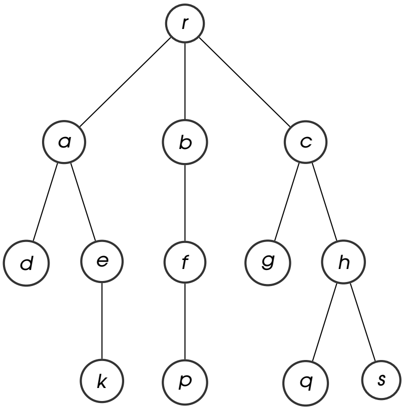


Rooted trees - Family trees

Credit: cdn health

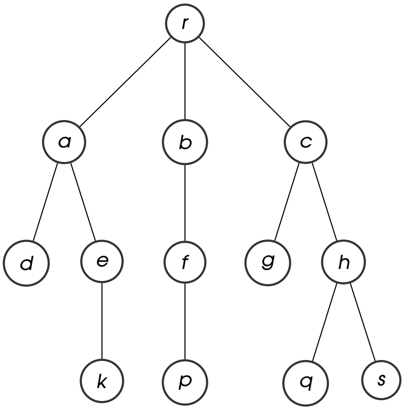


Rooted trees - Terminologies



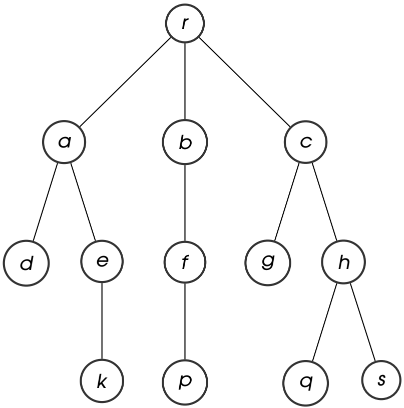
- Topmost vertex is called the root

Rooted trees - Terminologies



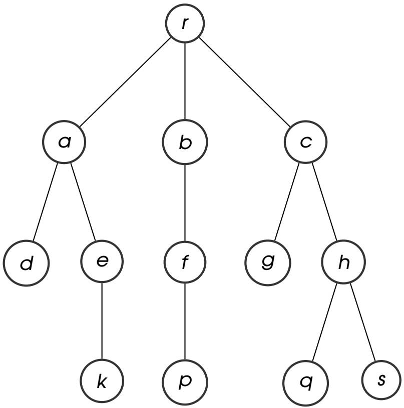
- Topmost vertex is called the root
- A vertex u may have some children below it, u is called the parent of its children.

Rooted trees - Terminologies



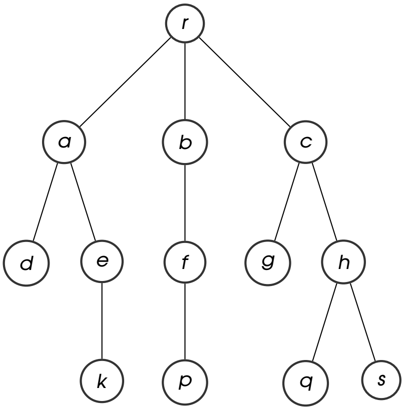
- Topmost vertex is called the root
- A vertex u may have some children below it, u is called the parent of its children.
- Degree of a vertex is the no. of children it has
- Degree of a tree is the max degree of all vertices

Rooted trees - Terminologies



- Topmost vertex is called the root
- A vertex u may have some children below it, u is called the parent of its children.
- Degree of a vertex is the no. of children it has
- Degree of a tree is the max degree of all vertices
- A vertex with no child (degree-0) is called a leaf
- Vertices other than leaves/root are called internal vertices

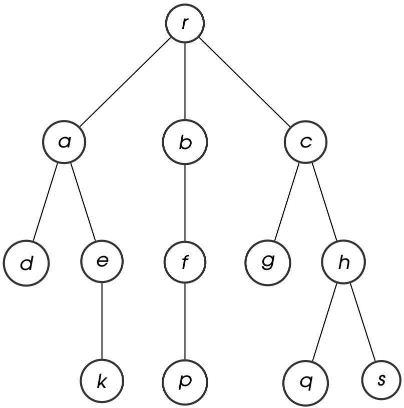
Rooted trees - Terminologies



r is parent of children a, b, c

- Topmost vertex is called the root
- A vertex u may have some children below it, u is called the parent of its children.
- Degree of a vertex is the no. of children it has
- Degree of a tree is the max degree of all vertices
- A vertex with no child (degree-0) is called a leaf
- Vertices other than leaves/root are called internal vertices

Rooted trees - Terminologies

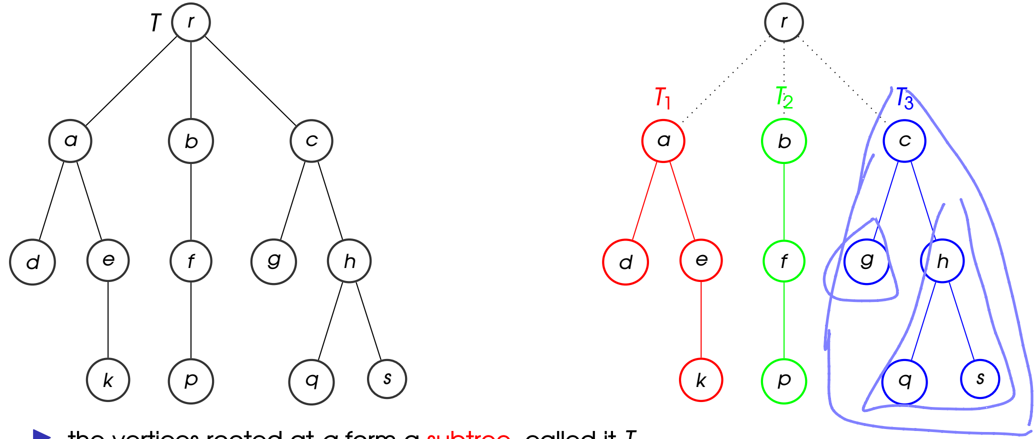


r is parent of children a, b, c

- Topmost vertex is called the root
- A vertex u may have some children below it, u is called the parent of its children.
- Degree of a vertex is the no. of children it has
- Degree of a tree is the max degree of all vertices
- A vertex with no child (degree-0) is called a leaf
- Vertices other than leaves/root are called internal vertices

deg-0: d, k, p, g, q, s (leaves)
deg-1: b, e, f; deg-2: a, c, h
deg-3: r

Rooted trees - More terminologies



- the vertices rooted at a form a subtree, called it T_1
- similarly, subtree T_2 is rooted at b, subtree T_3 is rooted at c
- we say that the tree T has three subtrees T_1, T_2 and T_3

Summary

Summary: Trees - basic terminologies

Next: Binary trees

For note taking