Notes on Tree

Presentation · May 2017

DOI: 10.13140/RG.2.2.28456.37128

Gede Pramudya
Universiti Tun Hussein Onn Malaysia
44 PUBLICATIONS **290** CITATIONS

TREE

Curated by Gede Pramudya

December 12, 2013

Introduction to Tree

In this lesson we will focus on a particular type of graph called tree, so named because such graphs resemble trees. For example, family trees are graphs that represent genealogical charts. Family trees use vertices to represent the members of a family and edges to represent parent-child relationship.

Definition

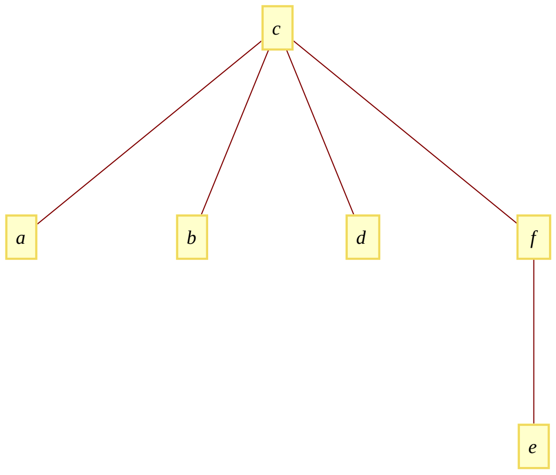
A tree is a connected undirected graph with no simple circuits. Because a tree cannot have a simple circuit, a tree cannot contain multiple edges or loops. Therefore, any tree must be a simple graph.

Theorem

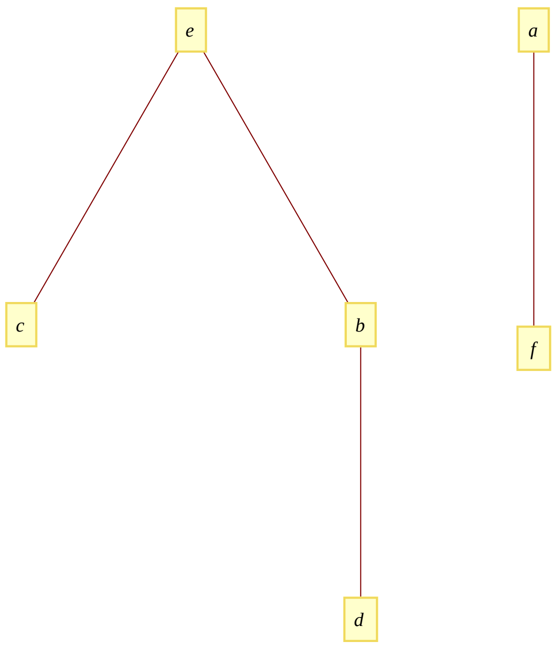
An undirected graph is a tree iff there is a unique simple path between any two of its vertices.

Examples

The following is an example of tree



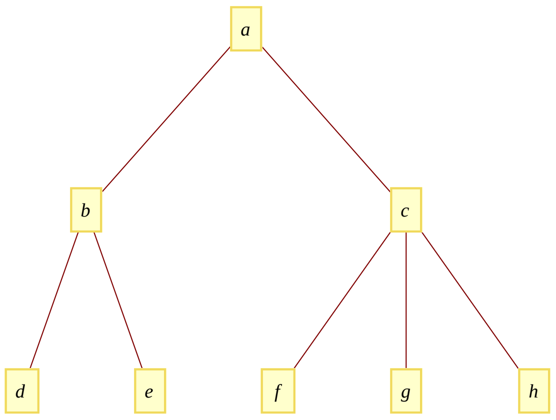
But, this not a tree.



Definition

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

Example

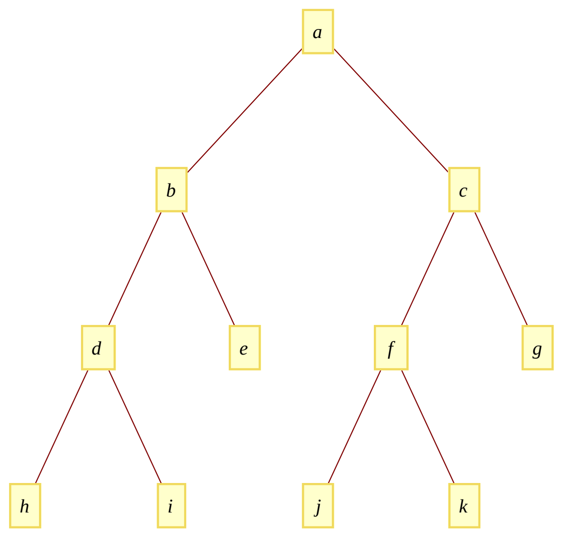


Definition

A rooted tree is called an m-ary tree if every internal vertex has no more than m children. The tree is called full m-ary tree if every internal vertex has exactly m children. An m-ary tree with m=2 is called a binary tree.

Example

The following tree is an example of binary trees.



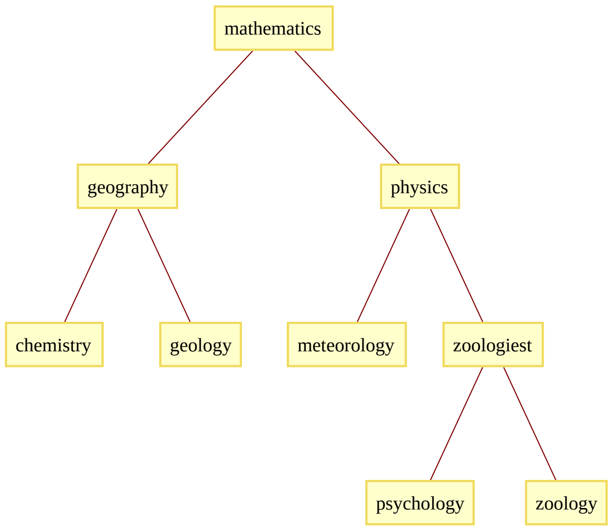
Application of Trees

Binary Search Tree

Searching for items in a list is one of the most important tasks that arises in computer science. Our primary goal is to implement a searching algorithm that finds items efficiently when the items are totally ordered. This can be accomplished through the use of a binary search tree, which is a binary tree in which each child of a vertex is designated as a right or left child, no vertex has more than one right child or left child, and each vertex is labeled with a key, which is one of the items. Furthermore,vertices are assigned keys so that the key of a vertex is both larger than the keys of all vertices in its left subtree and smaller than the keys of all the vertices in its right subtree.

Example

A binary tree for the words mathematics, physics, geography, zoology, meteorology, geology,psychology, zoologist, and chemistry (using alphabetical order).

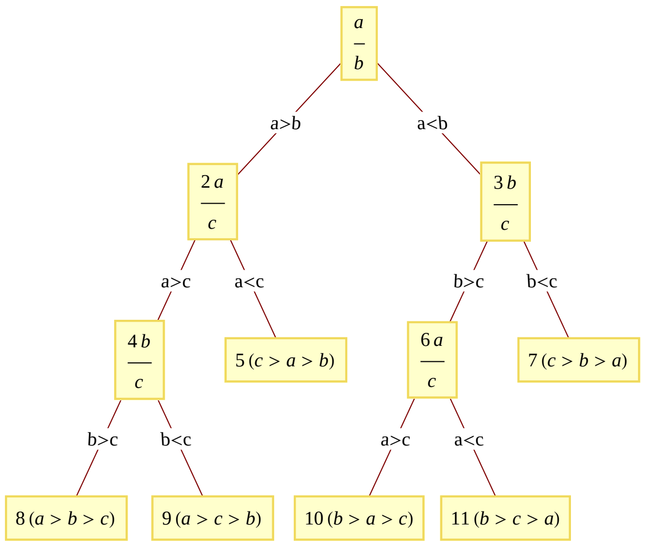


Decision Trees

Rooted trees can be used to model problems in which a series of decisions leads to a solution. For instance, a binary tree can be used to locate items based on a series of comparisons, where each comparison tell us whether we have located the item, or whether we should go right or left in a subtree. A rooted tree in which each internal vertex corresponds to a decision, with a subtree at these vertices for each possible outcome of the decision, is called a **decision tree**. The possible solution of the problem correspond to the paths to the leaves of this rooted tree.

Example

The following is a decision tree that order the elements of the list a, b, c.



Tree Traversal

Ordered rooted trees are often used to store information. We need procedures for visiting each vertex of an ordered rooted tree to access data. We will describe several important algorithm for visiting all vertices of an ordered rooted tree. Ordered rooted trees can also be used to represent various types of expressions, such as arithmetic expressions involving numbers, variables, and operations. The different listings of the vertices of ordered rooted trees used to represent expressions are useful in the evaluation of these expressions.

Traversal Algorithms

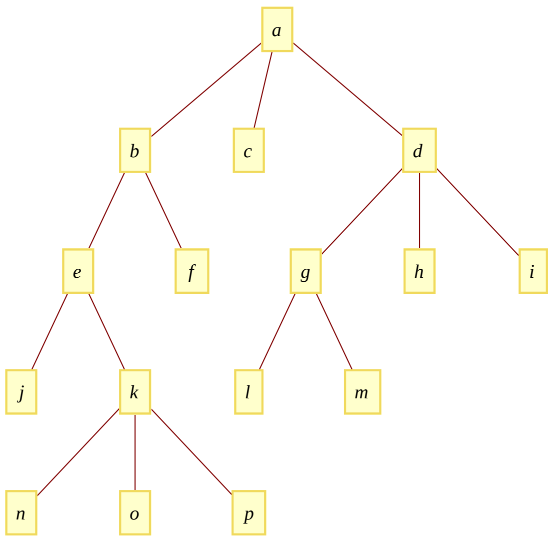
Procedures for systematically visiting every vertex of an ordered rooted tree are called traversal algorithms.

Definition: Preorder Tranversal

Let T be an oredered rooted tree with root r. If T consists only r, then r is the preorder traversal of T. Otherwise, suppose that $T_1, T2, \cdots, T_n$ are the subtrees at r from left to right in T. The preorder traversal begins by visiting r. It continues by traversing T_1 in preorder, then T_2 in preorder, and so on, until T_n is traversed in preorder.

Example

In which order does a preorder traversal visit the vertices in the ordered rooted tree T shown as belows.



The steps of the preorder traversal of T as follows. We traverse T in preorder by first listing the root a, followed by the preorder list of the subtree with root b, the preorder list of the subtree with root c, and the preorder list of subtree with root d. Consequently, the preorder traversal of T is $a, b, e, j, k, n, o, p, f, c, d, g, l, m, h, i$.

Definition: Inorder Tranversal

Let T be an ordered rooted tree with root r. If T consists only r, then r is the inorder traversal of T. Otherwise, suppose that $T_1, T2, \cdots, T_n$ are the subtrees at r from left to right in T. The inorder traversal begins by traversing T_1 in inorder, then visiting r. It continues by traversing T_1 in preorder, then T_2 in inorder, and so on, until T_n is traversed in inorder.

Example

The steps of an inorder traversal of the ordered rooted tree T in the previous example begins with an inorder traversal of the subtree with root b, the root a, the inorder listing of the subtree with root c, and the inorder listing of the subtree with root d. Therefore, the inorder listing of the ordered rooted tree is $j, e, n, k, o, p, b, f, a, c, l, g, m, d, h, i$.

Definition: Postorder Tranversal

Let T be an ordered rooted tree with root r. If T consists only r, then r is the postorder traversal of T. Otherwise, suppose that $T_1, T2, \cdots, T_n$ are the subtrees at r from left to right in T. The postorder traversal begins by traversing T_1 in postorder, then T_2 in postorder, ...,then T_n in postorder and ends by visiting r.

Example

The steps of postorder traversal of the ordered rooted tree T begins with the postorder traversal of the subtree with root b, the postorder traversal of the subtree with root c, the postorder traversal of subtree with root d, followed by the root a. Hence, the postorder traversal of tree T is $j, n, o, p, k, e, f, b, c, l, m, g, h, i, d, a$.

Infix,Prefix, and Postfix Notation

We can represent complicated expressions, such as compound propositions, combinations of sets, and arithmetic expressions using ordered rooted trees.

Example

What is the ordered rooted tree that represents the expression $((x + y) \uparrow 2) + ((x - 4)/3)$? The binary tree for this expression can be built from the bottom up. First, a subtree for the expression $x+y$ is constructed. Then this is incorporated as part of the larger subtree representing $(x+y) \uparrow 2$. Also, a subtree for $x - 4$ is constructed, and then this is incorporated into subtree representing $(x - 4)/3$. Finally, the subtrees representing $(x + y) \uparrow 2$ and $(x - 4)/3$ are combined to form the ordered rooted tree representing $((x + y) \uparrow 2) + ((x - 4)/3)$. The fully parenthesized expression obtain in this way is said to be in **infix form** (visit the ordered rooted binary tree by using inorder traversal algorithm). We can also find the **prefix form** of an expression when we traverse its binary tree in preorder. We obtain the **postfix form** of an expression by traversing its binary tree in postorder. Therefore, the prefix and postfix forms of expression $((x + y) \uparrow 2) + ((x - 4)/3)$ are $+ \uparrow +xy2/ - x43$ and $xy + 2 \uparrow x4 - 3/+$ consecutively.