Assignment 1:3 – Binary Tree

Introduction

Unit 1:3 assigns students to present the definition of a relatively complex term in three ways—parenthetical, sentence and expanded forms—within their chosen discipline or profession. The ability to effectively write definitions that allow understanding from different audiences is an important component of technical writing. Written communication will differ between colleagues, friends, family, and unknown audiences. The usage of colloquial, jargon and slang terminologies may limit our ability to cross-communicate between different groups. Thus, it is important that we able to cater towards different audiences within the given context of the information being communicated.

In this assignment, I have chosen to present the definition of 'binary tree' towards novice Computer Science students—who will encounter this data structure in their studies—in the aforementioned three forms.

Parenthetical Definition

Binary trees (trees that only divide into one or two branches at each joint) are examples common data structures found in the field of Computer Science ("Definition").

Sentence Definition

A binary tree is a type of data that is structured into a tree-like form. An example of a binary tree would include simple family trees that only have up to two children per parent-pair. In Computer Science, it is usually portrayed as an inverted tree. A joint, which is also called a 'node', will contain information that is relevant to data that are stored adjacent joints. Moreover, the nodes located at the tips of a binary tree are called 'leaves', whereas the origin is called the 'root'. Lastly, it is important to note that this type of data structure can only expand in up to two branches at each node ("Organizing" 3)**.**

Extended Definition

History of Binary Trees

In 1951, David Huffman was writing a term paper that was presented with a problem of finding the most efficient way of expressing binary code ("Huffman"). Huffman was able to solve this problem by forming a frequency-based binary tree. From this success, he published called "A Method for the Construction of Minimum-Redundancy Codes" which became led into the principles of a binary tree (Huffman *et al.*).

What are Binary Trees?

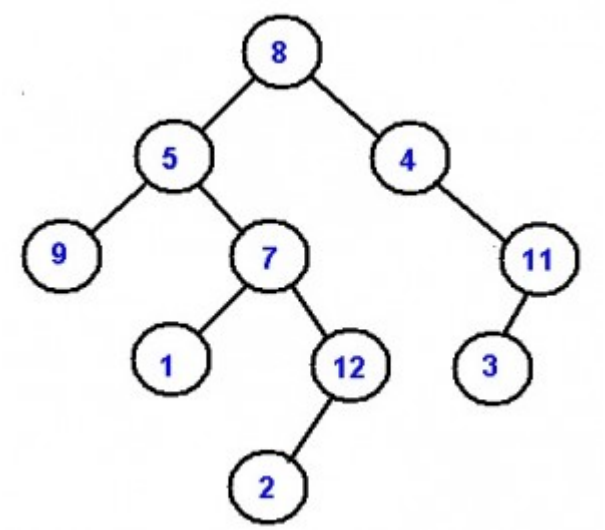

Figure 1. Binary Search Tree (Source: “Binary”).

In Computer Science, binary trees are used to store data in a hierarchal structure that usually portrayed as an inverted tree, as seen in ‘Figure 1’. The ‘root’ is the node that acts as the origin or the beginning of a binary tree. Each node is restricted to have up to two ‘children’ nodes. A node that has no additional nodes connected to it is called a ‘leaf’. The most common type of binary tree is called the ‘Binary Search Tree’ – which is also known as a ‘BST’ (“Binary”). A BST usually involves invariant criteria for expansion of its nodes. For example, Figure 1 shows a BST with nodes that have larger or smaller values than the parent node. Nodes with smaller numbers expand to the right of the parent node. On the other hand, nodes with larger values expand to the left of the parent node.

What examples of non-Binary Trees?

It is important to distinguish a binary tree from a *multi-ary* tree. By definition, a binary tree will have nodes that can only expand with up to two ‘children’ nodes. On the other hand, a *multi-ary* tree can have infinite ‘children’ nodes upon expansion. Most binary trees also have invariant criteria that is similar to the aforementioned BST example. However, most *multi-ary* trees do not have very specific invariant criteria. Thus, we can distinguish a binary tree from a *multi-ary* tree with these properties (Garnier *et al.*).

What are the applications?

An example of its application is observed in folder and file systems in computers. The main folder can be seen as the ‘root’. Any subfolders that are present in the main folder are called ‘nodes’. This pattern may continue until actual files are present. These files would be considered as the ‘leaves’, since they represent the tip of a tree. It is important to note that we are using this example within the context of a binary tree. Thus, each subfolder would either contain two subfolders, two files, or have one of each.

Why is this important to you?

Knowing the principles of a binary tree will allow better understanding in data structures. This is especially important in the field of Computer Science. It is seen in programming languages such as Java (“Binary Tree”). However, there are limitations of a binary tree in actual practice. In fact, *multi-ary* trees are more commonly used in data storage in servers or computers (Garnier *et al.*). Ultimately, it is important to acquire this understanding of binary trees, along with knowledge of other principles in Computer Science, in order to create better programs.

Works Cited:

“Definition of Binary Tree in English:.” *Binary Tree*. N.p., n.d. Web. 25 Jan. 2016. <http://www.oxforddictionaries.com/definition/english/binary-tree>.

“Organizing Data: Trees and Graphs.” *Organizing Data: Trees and Graphs* (n.d.): 1-16. *Organizing Data: Trees and Graphs*. Carnegie Mellon University, 20. Web. 25 Jan. 2016.

“Huffman Coding”. Wikimedia Foundation, n.d. Web. 25 Jan. 2016.

Huffman, David A. “Incremental Calculation of Minimum-Redundancy Length-Restricted Codes.” *IEEE Transactions on Communications IEEE Trans. Commun.* 55.2 (2007): 380. Web. 25 Jan. 2016.

“Binary Trees.” *Binary Trees*. N.p., n.d. Web. 25 Jan. 2016. <http://cslibrary.stanford.edu/110/BinaryTrees.html>.

Garnier, Rowan, and John Taylor. *Discrete Mathematics: Proofs, Structures and Applications*. Third Edition ed. Boca Raton: CRC, 2009. Print.

“*Binary Fractal Tree”*. N.p., n.d. Web. 25 Jan. 2016. <http://benice-equation.blogspot.ca/2013/01/binary-fractal-tree.html>.