Property of Binary Search Tree(BST)

Yumo Bai

September 2022

1 Introduction

During the class, we introduced an extremely useful data structure Binary Search Tree. In this supplemental material, I will prove few properties of binary search tree as well as provide its formal definition.

2 Definition of BST

A binary search tree is organized in a binary tree. We can represent the Binary Search Tree by a linked data structure in which each node is an object. In addition to a key and satellite data, each node contains attributes left, right, and p that point to the nodes corresponding to its left child,its right child, and its parent, respectively. If a child or the parent is missing, the appropriate attribute contains the value NIL(null). The root node is the only node in the tree whose parent is NIL(null).

The keys in a binary search tree are always stored in such a way as to satisfy the **binary-search-tree property:**
Let x be a node in a binary search tree. If y is a node in the left sub-tree of x, then $y.key \leq x.key$ If y is a node in the right sub-tree of x, then $y.key \geq x.key$. [1]

3 Property of Binary Search Tree

3.1 Property 1: At level i of Binary Search Tree there is at most 2^{i-1} nodes

This property could prove by induction.

Proof. When i=1. There is only root node. $2^{i-1} = 2^{1-1} = 1$ Therefore, the statement is true when i=1.

Assume $\forall j \in \mathbb{N} \quad i > j \geq 1$ the statement is true, consider j+1.

By inductive hypothesis we have that for level j there is at most 2^{j-1} nodes. According to the definition of binary search tree, each node have at most degree of 2. Therefore, for j+1 level it will have at most $2 * 2^{j-1}$ nodes which is 2^j. $\square$

3.2 Property 2: The Binary Search Tree with depth k will have at most $2^k - 1$ nodes

Proof. According to the property one, we have that for ith level there is at most 2^{i-1} nodes. Therefore, we have:

$$\sum_{i=1}^{k} 2^{i-1} = 2^0 + 2^1 \dots 2^{k-2} + 2^{k-1}$$

It is easily to find that the above series is geometric series that the sum of it could be calculated as:

$$S_k = \frac{a_1(1 - r^k)}{1 - r}$$

Applying the formula we have:

$$S_k = \frac{1(1 - 2^k)}{1 - 2}$$

$$S_k = \frac{(1 - 2^k)}{-1}$$

$$S_k = 2^k - 1$$

$\square$

3.3 Property 3: For a Complete Binary Search Tree with N nodes, the height of this Tree is $\lfloor log_2(n) \rfloor + 1$

Proof. Let the height of this Complete Binary Search Tree be $h \in \mathbb{N}$

By the Property 2 and the definition of Complete Binary Search Tree we have:

$$2^{h-1} - 1 < N \leq 2^h - 1$$

Because $N \in \mathbb{N}$ we have:

$$2^{h-1} \leq N < 2^h$$

Take Log on the inequality we have:

$$h - 1 \leq log_2(N) < h$$

Therefore $h = \lfloor log_2(n) \rfloor + 1$ $\square$

3.4 Property 4: For any Binary Search Tree $\mathbb{T}$, if it has n_0 leafs and n_2 nodes with degree of 2. Then $n_0 = n_2 + 1$

Proof. Suppose there are n_1 nodes with degree of 1, n nodes, and e edges. Then by the definition of Binary Search Tree:

$$n = n_0 + n_1 + n_2$$

$$e = 2n_2 + n_1$$

Based on these two equations, we have:

$$2n_2 + n_1 = n_0 + n_1 + n_2 - 1$$

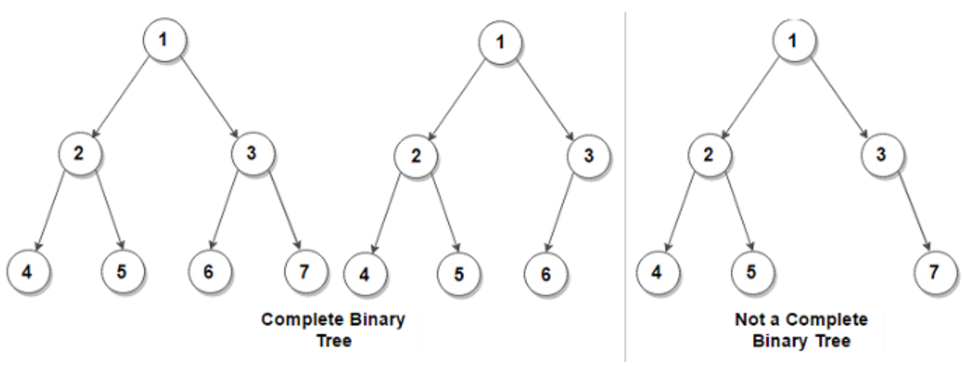
This is because besides of root node every node will correspond with one edge.

$$n_2 = n_0 - 1 \Rightarrow n_0 = n_2 + 1$$

□

4 Challenge Questions

If there is a Complete Binary Search Tree has N nodes and we number its nodes as following diagram:



Then for all nodes i ($1 \leq i \leq N$) we have following properties:

1. If $i = 1$, then the node is the root. If $i > 1$ its parents is $\lfloor i/2 \rfloor$
2. If $2i > N$ then the node have no left child; otherwise, its left child is $2i$.
3. If $2i + 1 > N$ then the node have no right child; otherwise, its right child is $2i + 1$.

References

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition.* The MIT Press, 3rd edition, 2009.