Binary Search Tree

Vivek Singh

Information Systems Decision Sciences (ISDS)
MUMA College of Business
University of South Florida
Tampa, Florida

2018

Introduction

A binary search tree is an important variation of binary trees.

- It relies on the property that keys that are less than the parent are found in the left subtree, and keys that are greater than the parent are found in the right subtree.
- We will call this the bst property.
- The property holds for each parent and child.
- All of the keys in the left subtree are less than the key in the root. All of the keys in the right subtree are greater than the root.

Essential Operations in BST

- Ability to create new empty map.
- Ability to Add a new key-value pair to the map. If the key is already in the map then replace the old value with the new value.
- Get the the value stored in the map for a given key or return None otherwise.
- Ability to delete a key-value pair from the map using a statement of the form del map[key].
- Get the size of the map.
- Ability to search for a key in the map.

Implementation in Python

For the implementation we will use the same class-oriented structure we used for linked lists, using a class for Tree Node and another for the Tree itself.

```
class BinarySearchTree:
    def __init__(self):
        self.root = None
        self.size = 0
    def length(self):
        return self.size
    def __len__(self):
        return self.size
    def __iter__(self):
        return self.root.__iter__()
```

Summary

- Binary Search Tree is an important variation of binary trees that has a great deal of importance in Computer science.
- The keys in the left sub-tree have a value less than the root key and the keys in the right sub-tree have greater values.
- Maps are based on the Binary search tree data structure.