Data Structures
Binary Search Trees - Insert and Delete

Cain Susko

Queen's University
School of Computing

February 6, 2022

Insert

to insert a new node into our binary search tree (BST) we must place the new node so that the resulting tree is still a valid BST. The algorith for inserting is as follows: Given a new value v to insert

1. if the tree is empty allocate a new node and put v into it and return

2. start at the root of the tree if not empty

3. if v is equal to the current node's value then return

4. if v is less than the current node's value and there's a left child, go left

   (a) else, allocate a new node and put v into it. Set this new node as the left child of the current node and return.

5. if v is greater than the current node's value and there is a right child, go right

   (a) else, allocate a new node and put v into it. Set the new node as the right child of the current node and return

The big O of this algorithm is:

$$O\left(\log_2(n)\right).$$

Furthermore, the configuration of a BST can change its effectiveness. The more full a tree is, the faster it is at searching (cause it avoids long 1-child strings of ordered numbers). This is known as a **balanced** tree.

Delete

deleting a node from a binary tree requires an algorithm for properly relinking the remaining nodes. This algorithm is as follows.
Given a value to delete v:

1. find the value v in the tree with a slightly modified BST search algorithm. Search using a `current` pointer and a `parent` pointer

2. if the node was found, delete it from the tree while making sure to preserve the ordering (there are 3 cases to consider).

   (a) if the found node is a leaf

      i. if the found leaf is not the root node, remove the leaf and set the current node to None

      ii. if the found leaf is the rood node, remove the root node and set the tree value to None

   (b) if the found node has 1 child

      i. if the found node is not the root node, relink the parent node to the current nodes only child.

      ii. if the found node is the root node, set the current node's only child as the root.

   (c) if the found node has 2 children

      i. we can replace the value in the found node with the largest value in the left subtree or the right subtree's smallest value. Once the found value is replaced, use case (a) or (b) to remove the node the found value was replaced with.

Min & Max

the minimum value of a BST is located at the leftmost node of the tree and the maximum is located at the rightmost node.