Brian Moye

CS 255-01

Determine how the height of a binary search tree (BST) is related to the number of nodes in the BST.

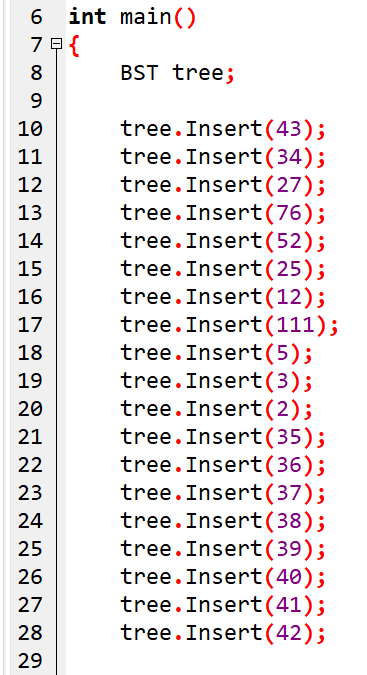
Due: 6 Dec. 2021

A binary search tree (BST) is a binary tree consisting of nodes with a left and right subtree. The left subtree would have items smaller than the root of that subtree, while the right would have items larger. Each subtree must also be a BST for the entire object to be considered a BST. The question at hand is how the height of a binary search tree (BST) is related to the number of nodes in the BST.

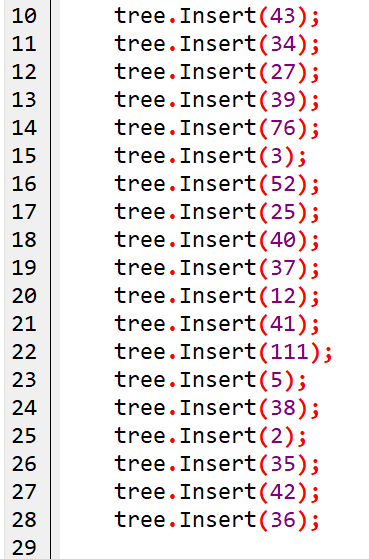
Hypothesis: The height of a BST is equal to the corresponding power of two that accommodates that number of nodes in the tree, minus one.

Methodology

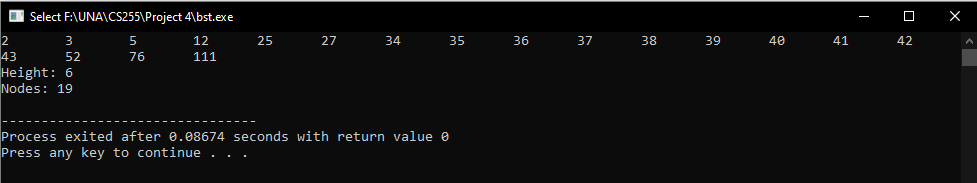
For my experiment, I decided to just pick numbers to fill a BST. I input the following in this order: 43, 34, 27, 76, 52, 25, 12, 111, 5, 3, 2, 35, 36, 37, 38, 39, 40, 41, 42



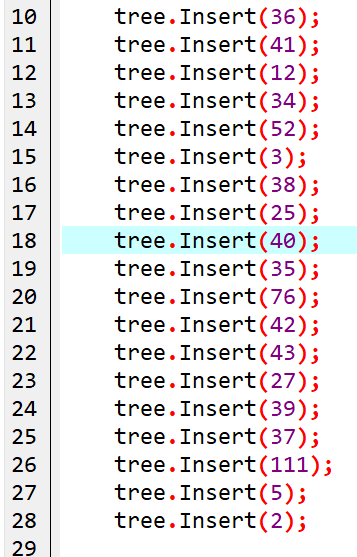
Based on my hypothesis, the height of this BST with 19 nodes should be 4, as 2^4 is 16 and 2^5 is 32, thus 5-1. However, the actual height was 9 as the order in which the items are inserted affected the height of the BST. Next, I tried inserting the same numbers in the following order: 43, 34, 27, 39, 76, 3, 52, 25, 40, 37, 12, 41, 111, 5, 38, 2, 35, 42, 36



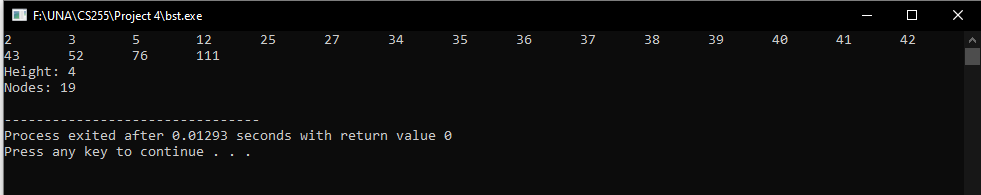
This input order resulted in a lower height of 6, as compared to the previous 9.



Next, I tried the following input order: 36, 41, 12, 34, 52, 3, 38, 25, 40, 35, 76, 42, 43, 27, 39, 37, 111, 5, 2



Which did yield the desired result of a height of 4.



Results:

From this I conclude that for an ideal BST, where the numbers are input in an optimal manner, the height of the BST does match the corresponding power of two that accommodates that number of nodes in the tree, minus one. In all other form of BST’s this would likely not hold true.