//function to find the height of the bst

|  |  |  |
| --- | --- | --- |
| int height(node \* x){ |  |  |
| int c = 1; | C1 | 1 |
| int d = 1; | C2 | 1 |
| if (x->left != NULL) | C3 | 1 |
| c += height(x->left); | C4 | <=1 |
| if (x->right != NULL) | C5 | 1 |
| d += height(x->right); | C6 | <=1 |
| if (c > d) | C7 | 1 |
| return c; | C8 | <=1 |
| else return d; | C9 | <=1 |
| } |  |  |

Each operation in the height function will occur at most once. The height function will be called for every node in the binary search tree, so we obtain:

N (C1 + C2 + C3 + C5 + C7 ) + <=N (C4 + C6 + C8 + C9 )

height(node \* x) ε O(N)

If the initial call to height is a node that is not the root, N will be the number of nodes in its subtree, including the node from the initial call.