Binary Search Trees

Binary Search Tree

Oct 27, 2020 Page 1
bool smap_contains_key(smap *m, const char *key) {
    smap_node *curr = m->root;
    while (curr != NULL && strcmp(key, curr->key) != 0) {
        if (strcmp(key, curr->key) < 0) {
            curr = curr->left;
        } else {
            curr = curr->right;
        }
    }
    return (curr != NULL);  
}
AVL Tree

add(s, 4VR)

properties:
1) same BST order property
2) all nodes have balance -1, 0, or 1

⇒ guarantees that h is $O(\log n)$
Single/Double Rotation

1. Identify the lowest unbalanced node.
2. Perform a single rotation to balance the tree.
3. If the tree is still unbalanced, perform a double rotation.

\[ \begin{align*}
T_1 < P < T_2 < P < T_3 \\
\end{align*} \]