Splay Tree

Shiv Shankar Dayal

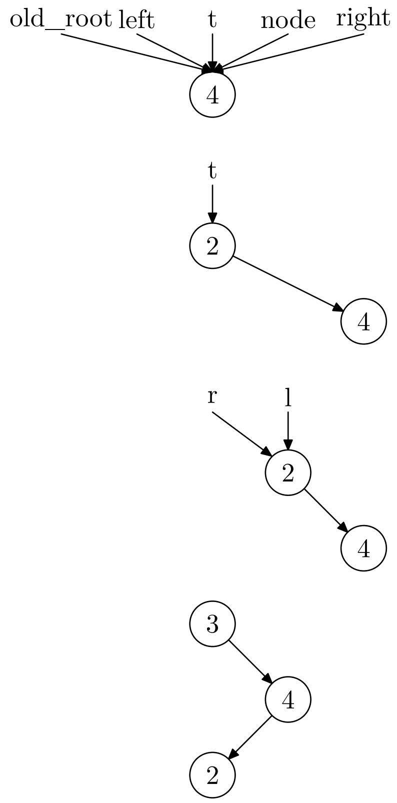
Insertion

```
splay(tree, data);
old_root = tree->root;

// Compare the current root with data:
if (old_root != NULL) {
  comp = (tree->comp)(data, old_root->data);
}

// If data is in the tree: overwrite it!
if (comp == 0) {
  old_data = tree->root->data;
  tree->root->data = data;
  return old_data;
}

// Otherwise insert a new node as a root and link the previous root to it:
tree->root = (sp_node *)malloc(sizeof(sp_node));
if (tree->root == NULL) {
  fprintf(stderr, "ERROR: Unable to allocate sp_node\n");
  tree->root = old_root;
} else {
  tree->root->data = data;
  if (comp > 0) {
    tree->root->left = old_root;
    tree->root->right = old_root->right;
    old_root->right = NULL;
  } else {
    tree->root->right = old_root;
    tree->root->left = old_root->left;
    old_root->left = NULL;
  }
}
```

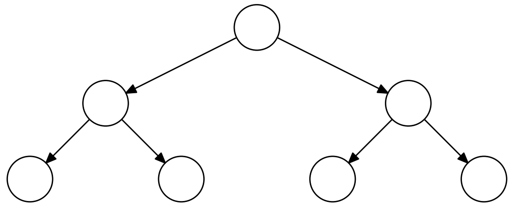


Splay

```
if (node->left == NULL) {
    break;
}

// Rotate right if needed:
if ((tree->comp)(data, node->left->data) < 0) {
    temp = node->left;
    node->left = temp->right;
    temp->right = node;
    node = temp;
    if (node->left == NULL) {
        break;
    }
}

// Link right:
right->left = node;
right = node;
node = node->left;
```



Splay Contd.

```
if (node->right == NULL) {
  break;
}

// Rotate left if needed:
if ((tree->comp)(data, node->right->data) > 0) {
  temp = node->right;
  node->right = temp->left;
  temp->left = node;
  node = temp;
  if (node->right == NULL) {
    break;
  }
}

// Link left:
left->right = node;
left = node;
node = node->right;
//------
// Assemble:
left->right = node->left;
right->left = node->right;
node->left = root.right;
node->right = root.left;
tree->root = node;
```