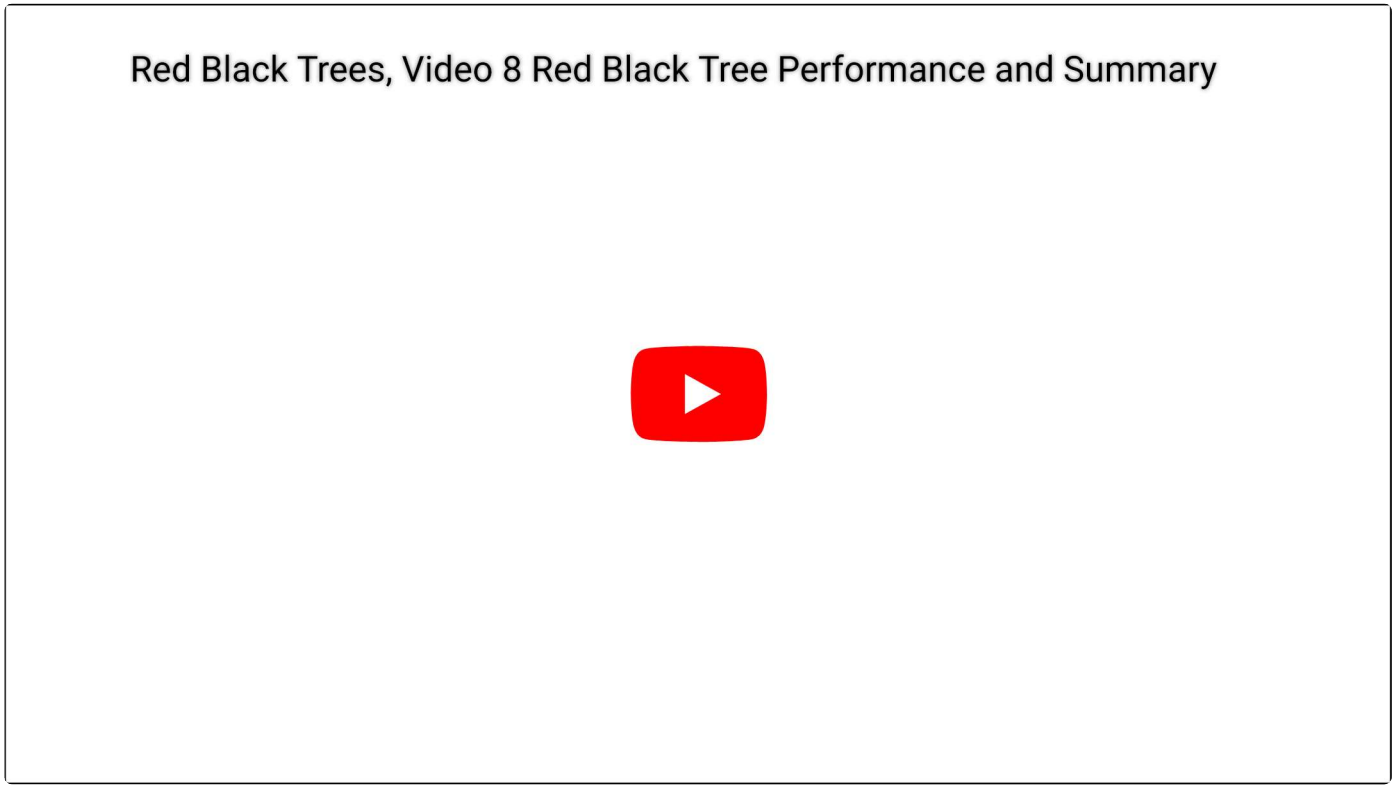
18.4 Runtime Analysis

Short and Sweet.

Because a left-leaning red-black tree has a 1-1 correspondence with a 2-3 tree and will always remain within 2x the height of its 2-3 tree, the runtimes of the operations will take $log(N)$time.



Here's the abstracted code for insertion into a LLRB:

```
private Node put(Node h, Key key, Value val) {
    if (h == null) { return new Node(key, val, RED); }

    int cmp = key.compareTo(h.key);
    if (cmp < 0)      { h.left  = put(h.left,  key, val); }
    else if (cmp > 0) { h.right = put(h.right, key, val); }
    else              { h.val   = val;                    }

    if (isRed(h.right) && !isRed(h.left))      { h = rotateLeft(h);  }
    if (isRed(h.left)  &&  isRed(h.left.left)) { h = rotateRight(h); }
    if (isRed(h.left)  &&  isRed(h.right))     { flipColors(h);      }

    return h;
}
```

Previous
18.3 Inserting LLRB Trees

Next
18.5 Summary

Last updated 1 year ago