COMP 251: Algorithms and Data Structures - Proofs Assignment

Liam Scalzulli
liam.scalzulli@mail.mcgill.ca

March 7, 2023

Complexity Proof

Claim: Inserting a node x into a red-black tree takes O(log n) time.

Presentation

Summary

Algorithm

Code explanation

Real world example

Correctness Proof

Claim: Red-Black Trees [CLRS 308]: A red-black tree with n internal nodes has height at most 2 log(n + 1).

Presentation

Proof. We start by showing that the subtree rooted at any node x contains at least $2^{bh(x)} - 1$ internal nodes. We prove this claim by induction on the height of x. If the height of x is 0, then x must be a leaf (T.nil), and the subtree rooted at x indeed contains at least $2^{bh(x)} - 1 = 2^0 - 1 = 0$ internal nodes. For the inductive step, consider a node x that has positive height and is an internal node with two children. Each child has a black-height of either $bh(x)$ or $bhx - 1$, depending on whether its color is red or black, respectively. Since the height of a child of x is less than the height of x itself, we can apply the inductive hypothesis to conclude that each child has at least $2^{bh(x)-1} - 1$ internal nodes. Thus, the subtree rooted at x contains at least $(2^{bh(x)-1} - 1) + (2^{bh(x)-1} - 1) + 1 = 2^{bh(x)} - 1$ internal nodes, which proves the claim. □

Source: CLRS

Summary

Algorithm

Code explanation

Real world example