

Problem 8_1 c)

Consider a red-black tree formed by inserting n nodes with the algorithm described in the lecture slides. Prove that if $n > 1$, the tree contains at least one red node.

There are strict rules that make a RED-BLACK Tree perform specific “actions” in logarithmic time. And one of these rules are:

- Every node is RED or BLACK
- The root is always BLACK

Considering that we have n (nodes) > 1 , and following the rules it is clear that there will always be a red node.

Why?

When we grow or shrink, the RED-BLACK Tree becomes tricky and hard to follow. It is hard to not break the rules. Therefore we decide that the inserted node to be RED (so we do not violate any rule). *Inserting a node and immediately coloring it RED it is much easier to identify and subsequently fix any violations.*

With that do be said for any $n > 1$ there will always be a RED node.