Red Black Tree inserting: why make nodes red when inserted?

In Red Black Tree Properties doesn't contain any rule for insert node color, how ever always we are inserting red color nodes.

If we insert black color node does it violate any Properties(any rule out of 4)?

algorithm data-structures tree binary-search-tree red-black-tree

edited Mar 16 '13 at 16:20
templatetypedef
167k 37 412 656

asked Mar 16 '13 at 16:07
Indika Anu
33 5

2 Answers

Yep! Suppose you have a single node in your tree:

```
5 (black)
```

Now insert a new black node into the tree:

```
5 (black)
 \
  9 (black)
```

Now the invariant that every root-null path in the tree has the same number of black nodes is broken, since the path from the root left has one black node and the paths from the root right each have two.

Hope this helps!

answered Mar 16 '13 at 16:19
templatetypedef
167k 37 412 656

You seem to be asking two questions

1) Why make nodes red when inserted (in title)?

2) Does inserting as black violate any properies?

You also seem to be under the mistaken impression that a Yes answer for 2) is an automatic justification for 1).

It is not so! Inserting a node as red *also* can violate one of the RB Tree properties. For instance, if you make the red node a child of another red node, you have just violated the property that red nodes can only have black children.

The reason for making it red, is that it is 'easier' to fix up the child node properties (by rotations and repainting parent/grandparents), rather than trying to fix up the path length properties. Perhaps another reason is that, that is what the authors came up with.

It might also be possible to fix up the tree by inserting a black node and not repainting red. Perhaps no one has given thought to it yet.

answered Mar 16 '13 at 17:41
Knoothe
872 3 12