

## Insertion Algorithm for Red-Black Trees

### CLRS: Ch 13.3

To insert a new node  $z$  into a red-black tree:

Step 1. Insert  $z$  as a red node, using the regular bst INSERT procedure.

Step 2. Fix up the tree to restore the red-black tree properties.

Step 2 consists of repeatedly checking which of the following three mutually exclusive cases applies, and applying the appropriate action for each case, until we are done!.

**Simple Case.** Either  $z$  is the root or  $z$ 's parent is black. If  $z$  is the root then blacken  $z$  and we are done, otherwise we are already done!

**Pushup Case.** Both the parent & the uncle of  $z$  are red. We blacken the parent & uncle of  $z$ , redden the grandparent of  $z$ , and let  $z$  be the grandparent of the old  $z$ .

**Rearrange Case.** The parent of  $z$  is red but its uncle is black. Rearrange the (zig-zig or zig-zag) 2-up-path starting at  $z$  into a sibling path. Blacken the middle node and redden the two end nodes of the resulting sibling path. We are done!

[PUSHUP & REARRANGE PICTURES GO HERE]

### Note

Pushup Case is CLRS Case 1. Rearrange Case is CLRS Cases 2–3.

### Timing

The inserting algorithm takes  $O(\log n)$  time.

Step 1, the bst insert, takes  $O(\log n)$  time.

The total time spent in the Pushup Case is  $O(\log n)$  since we can push all the way to the root.

The total time spent in the Simple Case and the Rearrange Case is  $O(1)$ .