

## I. RB Tree Definition

A Red-Black tree is a kind of enhanced binary tree. It has the following characteristics:

- A. *Every node is either **RED** or **BLACK***
- B. *The **ROOT** is always **BLACK***
- C. *If a node is **RED**, it's **CHILDREN** must be **BLACK***
- D. ***BLACK HEIGHT** must be the same for every root-leaf path (including null children)*

## II. Transformations

The following transformations are allowed to maintain red-black correctness:

### A. *Rotation*

Left or Right child becomes parent, parent becomes opposite-side child of the new parent.  
An inside grandchild (G-F-C) becomes an opposite-side inside grandchild (F-G-C) upon rotation.

Alternately, An inside S-grandchild becomes an inside S'-grandchild upon rotation

### B. *Flip*

BRR → RBB

### C. *Toggle node color*

Any node can be changed from Red to Black or Black to Red as needed.

### III. RB Tree Algorithm

Insertion is similar to insertion in a regular binary tree; you traverse the tree, comparing key to each successive node until you find the right location.

In order to maintain a correct RBTree structure, there are several additional steps:

**A. color-flips during descent**

For any node-children on descent, if B-RR then flip to R-BB

Note that this doesn't change black-height for the children. Why not?

Rationale: will change R leaf nodes to B without changing black height; then inserting a new R as a child will not cause a violation of Rule 3 (Red nodes can have only Black children)

**B. rotations during descent**

If node X is color-flipped during descent, and X's parent P is already red, the result is a violation of rule 3 (R-R), we must fix the violation using rotations.

**1. (P is RED), and X is an outside grandchild of G**

Three steps required

- a) toggle G
- b) toggle P
- c) rotate G to raise X

**2. (P is RED), and X is an inside grandchild of G**

Four steps required

- a) toggle G
- b) toggle X
- c) rotate P to raise X
- d) rotate G to raise X

**C. post-insertion rotation(s)**

There are three possibilities for G-P-X:

**1. P is BLACK**

In this case, nothing further is required

**2. P is RED, and X is an outside grandchild of G**

Three steps required

- a) toggle G
- b) toggle P
- c) rotate G to raise X

**3. P is RED, and X is an inside grandchild of G**

Four steps required

- a) toggle G
- b) toggle X
- c) rotate P to raise X
- d) rotate G to raise X