Red Black Tree - Insertion

Initial Step
A new key is always inserted at the leaf node. Traverse through the BST and insert the given Node at the correct leaf position.

Case 0: Black Parent

If the parent of the new node is **Black** and the tree was a valid RB-tree before insertion, no condition can be violated and there is nothing to do.

This is a **terminal case**.

Case 1: Red Uncle

Red property violated by the new node and the **Uncle** of the new node is Red. the tree can be restored to a valid RB-tree by recoloring as follows:

1) Recolor **Parent** and **Uncle** in **Black**.

2) Recolor the **Grandparent** in **Red**.

3) **If the Grandparent's Parent is Red, there could be another violation of the red property. Therefore, propagate upwards in the tree (while treating the Grandparent as the newly inserted node, possibly until reaching the root)(in the end change root to black if necessary).**
Check all cases again

Case 2: Black Uncle (Triangle)

Red property violated and the **Uncle** of the new node is **Black**; Grandparent, Parent and new node form a **Triangle**.

1) Apply a Rotation on the **Parent** of the new node. The rotation should get Grandparent, Parent and new node into a **Line configuration**

- **Left Rotation** is applied when the Triangle points to the Left.
- **Right Rotation** is applied when the Triangle points to the Right.

2) Consider the former **Parent** as the newly inserted node.(Do not swap them, but just shift the status of "new" to the former Parent.)

3) **Continue with Case 3**

Case 3: Black Uncle (Line)

Red property violated and the **Uncle** of the new node is **Black** ; Grandparent, Parent and new node form a **Straight Line**.

1) Recolor the **Parent Black** and recolor the **Grandparent Red.**

2) Apply a Rotation on the **Grandparent** of the new node towards the opposite site the tree is currently leaning.

- **Left Rotation** is applied when the new node is a **Right child**.(MIRROR)
- **Right Rotation** is applied if the new node is a **Left child**.

This is a **terminal case**.