Red Black Tree - Deletions

Before deletion find node to be deleted and copy to it the value from either the in order predecessor or successor. This successor or predecessor is now the new node to be deleted. If node to do be deleted is a leaf node. Don't worry, consider it the node to be deleted.

REMEMBER!: The standard case assumes the node to be deleted is a left child. If it is a right child, consider it a mirror case.

Case 0: Red Node or Red Child

In this case if the node to be deleted is red and has **only** null children, simply remove it. If it has a single child that is red, then recolour the child black and delete the node to be deleted.

This is **terminal case**.

Case 1: Red Brother

In this case the depth property is violated and the node to be deleted's brother is red. You must **recolour the brother black and parent red. Then apply left rotation on parent.**

This is not a teminal case and you must **continue to case 2, 3 or 4**.

Mirror

This applies if the node to be deleted is a right child. In this case perform the same recolouring and apply a **right** rotation on parent.

This is not a teminal case and you must **continue to case 2, 3 or 4**.

Case 2: All Black

In this case the depth property is violated and the brother of the node to be deleted is black and has two black children.

If parent also black: Make brother red and CONSIDER the parent as the new node to be deleted (**i.e. search for cases again from the top!**)
Note: If new (considered) node to be deleted is the root and black, you are finished.

If parent red: Exchange colours between parent and brother.
This is a **terminal case, you are done**.

Case 3: Black Brother, only! left nephew red

In this case the node to be deleted has a black brother whose left child is red. You must **recolour the left nephew black and the brother red. Then apply a right rotation to the brother.**

You must now **proceed with case 4**.

Mirror

If the node to be deleted is a right child this case is applicable when the brother is black and it's **right child** is red. You follow the same recolouring scheme but apply a **left rotation** to the brother instead.

You must now **proceed with case 4**.

Case 4: Black Brother, right or both nephews red

In this case the node to be deleted has a black brother which has atleast a red **right** child. (The other child or parent's colour is not important). You must **recolour the brother in the colour of the parent and recolour the parent and right nephew black**. Then perform a **left rotation** on the parent.

Case 4 is a **terminal case**.

Mirror

If the node to be delected is a right child this case is applicable when the brother is black and **at least the left nephew is red**. The recolouring procedure remains the same as in the standard case but instead we perfrom a **right rotation** on the parent.

Case 4 is a **terminal case**.