# Red-black Tree Description

A **red-black tree** is a balanced binary search tree. All basic operations take O(log n) time in the worst case.

Properties:

1. Every node is either red or black
2. The root is black
3. Every leaf is black
4. If a node is red, then both its children are black
5. For each node, all simple paths from the node to descendant leaves contain the same number of black nodes.

Rotations must be done when inserting or deleting to preserve the properties of the tree.

Insertion: we insert a node into the tree like a binary search tree and color it red initially. Then we call an auxiliary procedure to recolor and rotate the nodes in the tree.

Deleting a node from a red-black tree is more complicated, and we need to define a transplant procedure that swaps two nodes. Deleting a node is structurally similar to deleting a node from a binary search tree. We need a dedicated fixup function to handle updating the coloring of the red-black tree.