Question 6

Why does the method remove(x) in the RedBlackTree implementation perform the assignment u:parent = w:parent? Shouldn’t this already be done by the call to splice(w)?

Answer:

In a red-black tree, every node must have 2 children. If a node does not have 2 children with data, its left and right pointer will point to two black nil nodes. Also, it has only one child node with data, the other pointer will point to a black nil node. However, when the function splice(u) is called, if u is a leaf node, it will not update the parent of u’s black nil child nodes. This violates the properties of the red-black tree. Therefore, we use the assignment statement “u:parent = w:parent” to create a link from the black nil node to its new parent.