# Control Flow Graph

Key:

Normal node: Starting node:

Finishing node: Defines X:

Reads X: X is constant: **X**

References another graph:

X\*

X’

#

1. getRoot()

**Root’.left**

1. RBTreeWithFaults()

**Root**\*

1. RBTreeWithFaults(RBNode root)

**Root**\*

root’

1. createInfinityNode()

createInfinityNode (rbnode)

1. createInfinityNode(RBNode leftchild)

Leftchild’

Node\*

1. createNullNode(RBNode parent)

parent’

Newnode\*

Newnode’

1. search(**int** k)

Currnode’

Currnode.key’

K’

SearchNode

**Root**’

K’

Currnode\*

Currnode’

1. insert(int k, string v)

newRBNode

SearchNode

y.key’

z.key’

Counter\*

y’

v’ y’

k’ z\*

**Root**’

K’

y\*

z.key’

y.key’

**Root.Left**\*

Z.black\*

**Root’**

z.parent\*

y.right\*

z’

y.left\*

z’

fixUpTree(RBNode))

Counter’

z’

Counter\*

1. leftChild(RBNode x,RBNode y)

x.left\* y’

y.Parent\* x’

Leftchild (RBNode,RBNode)

1. transplate(RBNode x, RBNode y)

x.Parent’

y’

x.Parent’

y’

x'

x.Parent.left’

Rightchild (RBNode,RBNode)

w\* x.Parent.Right’

w.Black’

1. deleteFixup(RBNode x)

x’

x.Black’

**root.Left’**

x.Black’

x.Parent.Right’

w\*

Counter\*

x.Parent’

leftRotate (RBNode)

w.Black\* x.Parent.Black\*

x\* x.Parent.Left’

Counter\*

w\*

x.Left’

w.Black’

W\* x.Parent.Right’

Counter\*

w.Left.Black’ w.Right.Black’

w.Black\* x.Parent.Black\*

x.Parent’

rightRotate (RBNode)

w.Right.Black’

x.Parent’

X\*

w.Black\*

Counter\*

counter’

x.Black\*

counter\*

w.Left.Black’ w.Right.Black’

w.Left.Black\*

w.Black\*

W\*

x.Parent.Left’

counter\*

W’

rightRotate (RBNode)

x.Parent.Black’

W.Black\*

x.Parent.Black\*

X\*

**Root.Left’**

Counter\*

x.parent’

rightRotate (RBNode)

w.Black\* x.Parent’

x\*

counter\*

w.Black\*

x.Parent.Black’

x.Parent.Black\*

w.Left.Black\*

w \* x.Parent.Left’

counter\*

w.Left.Black’

W’

LeftRotate (RBNode)

w.Black\* w.Right.Black\*

X\*

**Root.Left’**

Counter\*

x.parent’

LeftRotate (RBNode)