Lab 10: Red-Black Tree

Compile with GCC 5.2.1@Ubuntu Gnome – VM workstation

With C89 ‘Standard’

**10.**

Use structer Tree{ right, left, value, parent, color }

Set enum COLOR{RED, BLACK};

Functions

sibling(return sibling of node)

removeNode(find a node have value, remove node using removeUtil)

removeUtil1,2,3,4,5,6

{

1: if(n->parent != NULL) call 2(n)

2: if(sibling(n)->color == RED) set n->parent->color RED, s->color BLACK

if(n==n->parent>left(right) rotateLeft(Right)(n->parent), call 3(n)

3: if(n->parent->color == BLACK) && sibling(n)->color == BLACK &&

s->left and right->color == BLACK then s->color = RED, call 1(n->parent)

else call 4(n)

4: if(n->parent->color == RED) && sibling(n)->color == BLACK &&

s->left and right->color == BLACK then s->color = RED,

n->parent->color=BLACK else call 5(n)

5: if(n==n->parent->left && s->right->color == BLACK &&

s->left->color == RED) then s->color = RED, s->left->color = BLACK, rotateRight(s); else if(n==n->parent->right && s->right->color == BLACK && s->left->color == RED) then s->color = RED, s->right->color = BLACK, rotateLeft(s); call 6(n);

6: s->color = n->parent->color, n->parent->color = BLACK;

If(n==n->parent->left(right)) then

s->right(left)->color = BLACK, rotateLeft(Right)(n->parent);

}

replaceNode(tree, node, node)

print inorder after end of input