Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторної роботи № 7

з дисципліни Алгоритми та структури даних

Варіант 18

Виконав студент групи СА-32

Маленчак Валентин Богданович

Прийняв доцент Щербак С.С.

*Львів 2017*

**Тема:** Дерева. Бінарні дерева. Пошук.

**Мета роботи:** набуття навичок програмування дерев.

**Завдання на роботу:** розробити засоби динамічного збереження дерев та виконання дій над ними згідно варіанту.

**Індивідуальне завдання:** вивести на друк всі чорні вершини червоно-чорного дерева.

# Хід роботи

Текст програми на мові С++:

#pragma once

#include <iostream>

#include <queue>

#include "UInterface.h"

using namespace std;

template <class T>

class RedBlackTree

{

private:

struct myNode

{

T data;

myNode \*left;

myNode \*right;

bool isBlack;

myNode(T element, myNode\* l = NULL, myNode\* r = NULL, bool black = true)

{

data = element;

left = l;

right = r;

isBlack = black;

}

};

UInterface myInterface;

int nodeCount(myNode\* node)

{

if (!node)

return 0;

if (node->left == NULL && node->right == NULL)

return 1;

int leftc, rightc;

if (node->left != NULL)

leftc = nodeCount(node->left);

else

leftc = 0;

if (node->right != NULL)

rightc = nodeCount(node->right);

else

rightc = 0;

return leftc + rightc + 1;

}

public:

myNode\* root;

RedBlackTree()

{

root = NULL;

}

~RedBlackTree()

{

deleteSubtree(root);

}

int getHeight(myNode\* node)

{

return node == NULL ? 0 : 1 +

max(getHeight(node->left), getHeight(node->right));

}

void deleteSubtree(myNode\* node)

{

if (node)

{

deleteSubtree(node->left);

deleteSubtree(node->right);

delete node;

}

}

void add(T element, myNode\*& node, bool color = true)

{

if (node == NULL)

{

node = new myNode(element);

if (!color)

node->isBlack = false;

}

else if (getHeight(node->left) < getHeight(node->right))

add(element, node->left, color);

else if (getHeight(node->left) > getHeight(node->right))

add(element, node->right, color);

else

if(nodeCount(node->left) > nodeCount(node->right))

add(element, node->right, color);

else

add(element, node->left, color);

}

void print(myNode\* node)

{

if (node != NULL)

{

print(node->left);

myInterface.outputData(node->data);

myInterface.outputColor(node->isBlack);

if (node->left != NULL)

myInterface.outputLeftSonData(node->left->data);

else

myInterface.outputLeftSonEmpty();

if (node->right != NULL)

myInterface.outputRightSonData(node->right->data);

else

myInterface.outputRightSonEmpty();

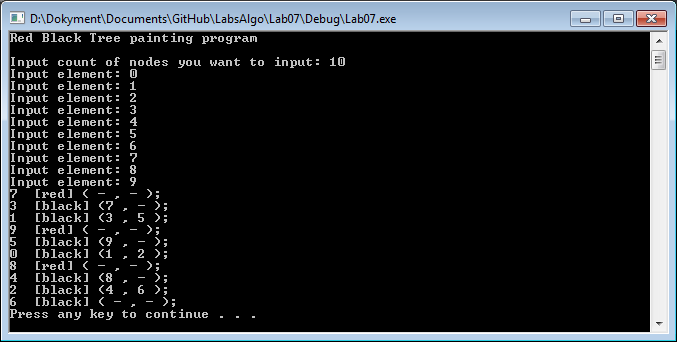
print(node->right);

}

}

};

Результати комп'ютерної реалізації:



**Висновок:** я набув практичних навичок програмування бінарних дерев.