Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №6 з дисципліни

«Основи програмування»

Варіант 2

Виконав студент Боровков Іван Ігорович

( прізвище, ім'я, по батькові)

Перевірив викладач Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №6

ДЕРЕВА

Лабораторна робота №6

ДЕРЕВА



Умова задачі:



Код задачі:

Tree.hpp

#include <cstddef>

class Tree

{

private:

    struct TNode

    {

        float val = NULL;

        TNode \*left = NULL, \*right = NULL;

    };

    std::size\_t treeSize;

    TNode \*root;

public:

    Tree();

    void AddNode(double val);

    double FindAverage();

    ~Tree();

};

Tree.cpp

#include "Tree.h"

#include <queue>

#include <stack>

using namespace std;

Tree::Tree()

{

    root = new TNode;

    treeSize = 0;

}

void Tree::AddNode(double val)

{

    if (root->val == NULL)

    {

        root->val = val;

        treeSize++;

        return;

    }

    TNode \*currentNode;

    queue<TNode \*> nodeQueue;

    nodeQueue.push(root);

    for (size\_t i = 0; i < treeSize; i++)

    {

        currentNode = nodeQueue.front();

        nodeQueue.pop();

        if(currentNode->left == NULL)

        {

            currentNode->left = new TNode;

            currentNode->left->val = val;

            break;

        }

        if(currentNode->right == NULL)

        {

            currentNode->right = new TNode;

            currentNode->right->val = val;

            break;

        }

        nodeQueue.push(currentNode->left);

        nodeQueue.push(currentNode->right);

    }

    treeSize++;

}

double Tree::FindAverage()

{

    double sum = 0;

    TNode \*currentNode;

    stack<TNode \*>  nodeStack;

    nodeStack.push(root);

    for (size\_t i = 0; i < treeSize; i++)

    {

        currentNode = nodeStack.top();

        nodeStack.pop();

        sum += currentNode->val;

        if(currentNode->left != NULL)

            nodeStack.push(currentNode->left);

        if(currentNode->right != NULL)

            nodeStack.push(currentNode->right);

    }

    return sum / static\_cast<double>(treeSize);

}

Tree::~Tree()

{

    TNode \*currentNode;

    stack<TNode \*>  nodeStack;

    nodeStack.push(root);

    for (size\_t i = 0; i < treeSize; i++)

    {

        currentNode = nodeStack.top();

        nodeStack.pop();

        if(currentNode->left != NULL)

            nodeStack.push(currentNode->left);

        if(currentNode->right != NULL)

            nodeStack.push(currentNode->right);

        delete currentNode;

    }

}

main.cpp

#include "Tree.h"

#include <iostream>

using namespace std;

int main(int argc, char const \*argv[])

{

    cout << "How much values do you want to enter?\n";

    int size;

    cin >> size;

    int \*valueArr = new int[size];

    cout << "Enter values:\n";

    for (size\_t i = 0; i < size; i++)

        cin >> valueArr[i];

    Tree tree;

    for (size\_t i = 0; i < size; i++)

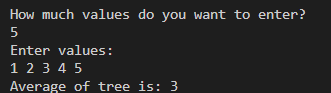
        tree.AddNode(valueArr[i]);

    cout << "Average of tree is: " << tree.FindAverage() << '\n';

    return 0;

}

Скріншоти виконання програми:



*Висновок*

В ході лабораторної роботи я вивчив особливості організації та обробки дерев.