МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 5**

по дисциплине:” Алгоритмы и структуры данных”

на тему:**”** **Построение 2-3-4 дерева”**

Вариант 1

Выполнил**:** студент группы 10701322

Бородкин Д.В.

Приняла**:** пр. Мисякова В.А.

Минск 2024

# Лабораторная работа №5. Построение 2-3-4 дерева.

# Цель работы: ****Закрепление теоретических знаний и получение практических навыков создания программ для построения 2-3-4 деревьев.****

# Задание:

* + 1. Построить 2-3-4 дерево из последовательности целых чисел (по варианту), внося их в дерево последовательно слева направо и изобразить это дерево.
    2. Изобразить 2-3-4 дерево после внесения каждого числа.

### Код программы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Lab2

{

public partial class Form1 : Form

{

TwoThreeFourTree tree = new TwoThreeFourTree();

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

int key = int.Parse(textBox1.Text);

tree.Insert(key);

textBox1.Clear();

}

private void button2\_Click(object sender, EventArgs e)

{

treeView1.Nodes.Clear();

TreeNode rootNode = new TreeNode("Root");

treeView1.Nodes.Add(rootNode);

tree.DisplayInTreeView(tree.root, treeView1.Nodes);

}

private void button4\_Click(object sender, EventArgs e)

{

treeView1.Nodes.Clear();

}

}

public class Node

{

public List<int> keys;

public List<Node> children;

public Node()

{

keys = new List<int>();

children = new List<Node>();

}

}

public class TwoThreeFourTree

{

public Node root;

public TwoThreeFourTree()

{

root = new Node();

}

public void Insert(int key)

{

if (root.keys.Count == 3)

{

Node newRoot = new Node();

newRoot.children.Add(root);

SplitChild(newRoot, 0);

root = newRoot;

}

InsertNonFull(root, key);

}

private void InsertNonFull(Node node, int key)

{

if (node.children.Count == 0)

{

node.keys.Add(key);

node.keys.Sort();

}

else

{

int i = node.keys.Count - 1;

while (i >= 0 && key < node.keys[i])

{

i--;

}

i++;

if (node.children[i].keys.Count == 3)

{

SplitChild(node, i);

if (key > node.keys[i])

{

i++;

}

}

InsertNonFull(node.children[i], key);

}

}

private void SplitChild(Node parent, int i)

{

Node node = parent.children[i];

Node newNode = new Node();

newNode.keys.Add(node.keys[2]);

node.keys.RemoveAt(2);

if (node.children.Count != 0)

{

newNode.children.Add(node.children[2]);

newNode.children.Add(node.children[3]);

node.children.RemoveRange(2, 2);

}

parent.keys.Insert(i, node.keys[1]);

node.keys.RemoveAt(1);

parent.children.Insert(i + 1, newNode);

}

public void DisplayInTreeView(Node node, TreeNodeCollection nodes)

{

TreeNode treeNode = new TreeNode();

foreach (int key in node.keys)

{

treeNode.Text += key.ToString() + " ";

}

nodes.Add(treeNode);

foreach (Node child in node.children)

{

DisplayInTreeView(child, treeNode.Nodes);

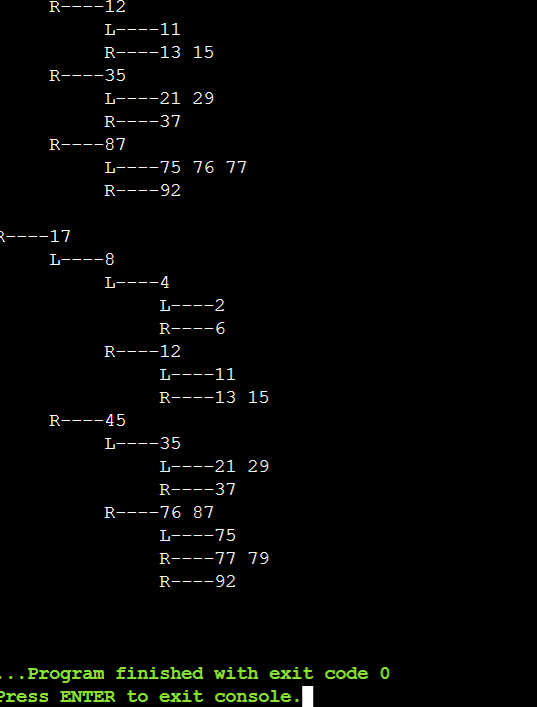
}

}

}

}

### Скриншот результата



## Контрольные вопросы

1. [2-3-4 дерево - это самобалансирующееся дерево, в котором каждый узел может иметь 2, 3 или 4 дочерних узла](https://www.geeksforgeeks.org/2-3-4-tree/). [Это также называется деревом B степени 4](https://www.geeksforgeeks.org/2-3-4-tree/).
2. [Узел 2-3-4 дерева может содержать 1, 2 или 3 элемента](https://www.geeksforgeeks.org/2-3-4-tree/). [Это зависит от типа узла: 2-узел содержит один элемент данных, 3-узел содержит два элемента данных, а 4-узел содержит три элемента данных](https://www.geeksforgeeks.org/2-3-4-tree/).
3. [Узел 2-3-4 дерева может иметь 2, 3 или 4 потомка](https://www.geeksforgeeks.org/2-3-4-tree/). [Это также зависит от типа узла: 2-узел имеет два дочерних узла, 3-узел имеет три дочерних узла, а 4-узел имеет четыре дочерних узла](https://www.geeksforgeeks.org/2-3-4-tree/).
4. [Ключи в узле 2-3-4 дерева определяют диапазон элементов, которые будут находиться в соответствующем сегменте](https://www.geeksforgeeks.org/2-3-4-tree/). [Элементы в каждом узле должны быть отсортированы от наименьшего к наибольшему](https://www.geeksforgeeks.org/2-3-4-tree/).
5. [Разбиение узлов в 2-3-4 дереве происходит при вставке нового элемента в 4-узел](https://www.geeksforgeeks.org/2-3-4-tree/). Если текущий узел является 4-узлом, этот узел разделяется. Среднее значение узла стирается и сохраняется. [Затем оставшийся 3-узел разделяется на два 2-узла](https://www.geeksforgeeks.org/2-3-4-tree/).
6. [Сложность поиска в B-дереве составляет O(log n)](https://www.geeksforgeeks.org/2-3-4-tree/). [Это обусловлено тем, что все внешние узлы находятся на одной и той же глубине, что гарантирует, что время выполнения операций, таких как вставка, удаление и поиск, всегда составляет O(log n)](https://www.geeksforgeeks.org/introduction-of-b-tree-2/).

## Вывод: Я изучил 2-3-4 деревья, их свойства, научился их строить.