

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего  
образования  
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий  
Кафедра информатики и систем управления

ОТЧЕТ  
по лабораторной работе  
по дисциплине:  
Алгоритмы и структуры данных

РУКОВОДИТЕЛЬ:  
Капранов С.Н.

\_\_\_\_\_  
(подпись)  
(фамилия, и.,о.)

СТУДЕНТ:  
Еричев Д.А.

\_\_\_\_\_  
(подпись)  
(фамилия, и.,о.)

\_\_\_\_\_  
(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_  
С оценкой \_\_\_\_\_

### Задача 10.

Дано N-дерево. Найти в дереве самые длинные пути без ветвлений.

#### 1) Программный код.

##### Класс Ntree.java:

```
package Laba4;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.*;
import java.util.Scanner;

public class Ntree {
    static int max = -1000;
    public static class TreeNode{
        int val;
        List<TreeNode> children = new LinkedList<>();
        TreeNode(int data){
            val = data;
        }
        TreeNode(int data, List<TreeNode> child){
            val = data;
            children = child;
        }
        TreeNode() {}
    }

    private static void printNAryTree(TreeNode root){
        System.out.println("N-Дерево:");
        if(root == null) {
            System.out.println("Дерево пустое!");
            return;
        }else{
            Queue<TreeNode> queue = new LinkedList<>();
            queue.offer(root);
            while(!queue.isEmpty()) {
                int len = queue.size();
                for(int i=0;i<len;i++) {
                    TreeNode node = queue.poll();
                    if(node != null){
                        System.out.print(node.val + " ");
                        for (int j=0;j<node.children.size();j++) {
                            queue.offer(node.children.get(j));
                        }
                    }
                }
                System.out.println();
            }
        }
    }

    private static void searchMethod(TreeNode root, int counter){
        if(root == null) {
            System.out.println("Дерево пустое!");
        }else{
            // ... (the rest of the search method code is missing in the original image)
        }
    }
}
```

```

Queue<TreeNode> queue = new LinkedList<>();
queue.offer(root);
while(!queue.isEmpty()) {
    int len = queue.size();
    for(int i=0;i<len;i++) {
        TreeNode node = queue.poll();
        if(node != null) {
            if(len == 1 || len == 2 ||
node.children.size() == 1) {
                counter++;
            }
            if(counter>max) {
                max = counter;
            }
            for (int j=0;j<node.children.size();j++) {
                queue.offer(node.children.get(j));
            }
        }
    }
}
Queue<TreeNode> queueMax = new LinkedList<>();
queueMax.offer(root);
while(!queueMax.isEmpty()) {
    if(max > 3) {
        int len = queueMax.size();
        for(int i=0;i<len;i++) {
            TreeNode node = queueMax.poll();
            if(node.children.size() == 1 || len == 1
|| len == 2) {    //len == 1, когда добавляем последнее число в
дерево, либо первое,
//node.children.size() == 1 -
количество в стеке 1 ребенок
            if(len == 1) {
                System.out.print(" ");
            }
            System.out.print(node.val+" ");
        }
        for (int j=0;j<node.children.size();j++) {
            queueMax.offer(node.children.get(j));
        }
    }
    max--;
    System.out.println();
}
}
}

public static void main(String[] args) throws
FileNotFoundException {
    TreeNode root = null;
    root =
read(root,"/home/denis/JavaPrograms/src/Laba4/file.txt");
    printNaryTree(root);
}

```

```

        System.out.println("Самые длинные пути без ветвлений:");
        searchMethod(root, 0);
    }
    static TreeNode read(TreeNode root, String path) throws
FileNotFoundException {
        Scanner sc = new Scanner(new File(path));
        root = new TreeNode(sc.nextInt());
        while (sc.hasNextInt()) {
            root.children.add(new TreeNode(sc.nextInt()));
            root.children.add(new TreeNode(sc.nextInt()));
            root.children.add(new TreeNode(sc.nextInt()));
            root.children.add(new TreeNode(sc.nextInt()));
            root.children.get(0).children.add(new
TreeNode(sc.nextInt()));
            root.children.get(0).children.get(0).children.add(new
TreeNode(sc.nextInt()));

            root.children.get(0).children.get(0).children.add(
new TreeNode(sc.nextInt()));
            root.children.get(1).children.add(new
TreeNode(sc.nextInt()));
            root.children.get(1).children.add(new
TreeNode(sc.nextInt()));
            root.children.get(1).children.add(new
TreeNode(sc.nextInt()));
            root.children.get(2).children.add(new
TreeNode(sc.nextInt()));
            root.children.get(2).children.get(0).children.add(new
TreeNode(sc.nextInt()));

            root.children.get(2).children.get(0).children.add(
new TreeNode(sc.nextInt()));

            root.children.get(2).children.get(0).children.get(
0).children.add(new TreeNode(sc.nextInt()));
            root.children.get(3).children.add(new
TreeNode(sc.nextInt()));
            root.children.get(3).children.add(new
TreeNode(sc.nextInt()));
            root.children.get(3).children.add(new
TreeNode(sc.nextInt()));
        }
        return root;
    }
}

```

## 2) Описание функций:

private static void printNaryTree(TreeNode root): Вывод дерева

private static void searchMethod(TreeNode root, int counter):поиск длинных путей без ветвлений

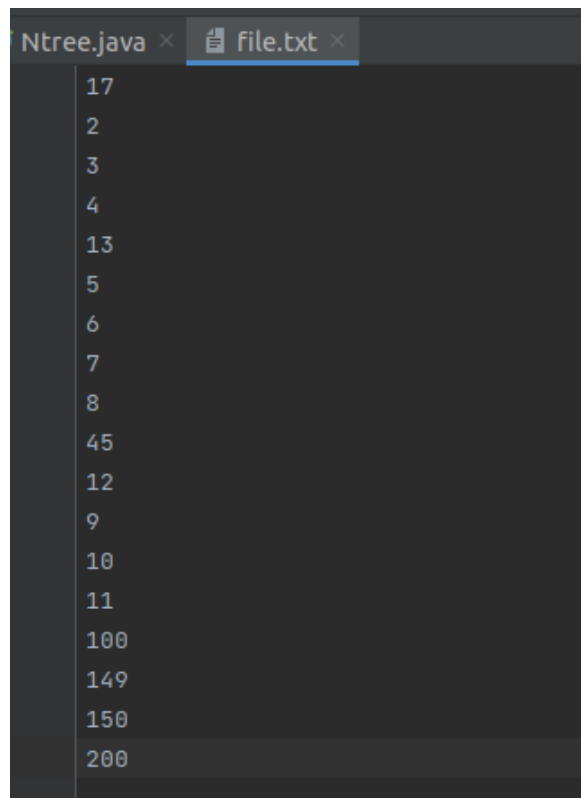
static TreeNode read(TreeNode root, String path) : считывание из файла

### 3)Результаты работы программы.

Вывод в консоль:

```
N-Дерево:
17
2 3 4 13
5 8 45 12 9 149 150 200
6 10
7 11
100
Самые длинные пути без ветвлений:
17
2 4
5 9
6 10
7 11
100
Process finished with exit code 0
```

Считывание из файла:



The screenshot shows an IDE window with two tabs: 'Ntree.java' and 'file.txt'. The 'file.txt' tab is active and displays a list of numbers, each on a new line. The numbers are: 17, 2, 3, 4, 13, 5, 6, 7, 8, 45, 12, 9, 10, 11, 100, 149, 150, and 200.