Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный университет»

Кафедра технологий программирования

Алгоритмы и структуры данных

Отчет по лабораторной работе №4

Вариант 11

Выполнил — Ланцев Евгений Николаевич. 21-ИТ-1, ФИТ

Проверил — преподаватель Виноградова А.Д.

Полоцк
2022 г.

Лабораторная работа № 4

“Деревья. Помеченное дерево. Дерево выражений. Обход дерева.”

Цель работы: ознакомиться с понятиями «дерево», «помеченное дерево», «дерево выражение», «обход дерева» и основными алгоритмами их реализации и обработки, научиться применять полученные знания на практике.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ (ответы на контрольные вопросы):

1. Определение понятия дерево.

Дерево – это нелинейная иерархическая структура данных. Она состоит из узлов и ребер, которые соединяют узлы.

2. Из каких частей состоит дерево?

- Узел — это объект, в котором есть ключ или значение и указатели на дочерние узлы. Узлы, у которых нет дочерних узлов, называют листами или терминальными узлами. Узлы, у которых есть хотя бы один дочерний узел, называются внутренними.
- Ребро связывает два узла.
- Корень — это самый верхний узел дерева. Его ещё иногда называют корневым узлом.

3. Определение понятия лес.

Лес — множество непересекающихся деревьев. Например, если «срезать» корень, получится лес.

4. Определение понятия высота дерева.

Высота узла — это максимальная длина пути от этого узла к самому нижнему узлу (листу).

5. Какое дерево называется полным?

Полным бинарным **деревом называется** такое **дерево**, в котором каждая вершина имеет не более двух "сыновей"

6. Определение понятия обход дерева.

Чтобы выполнить какую-либо операцию с деревом, нужно добраться до определенного узла. Для этого и существуют алгоритмы обхода дерева. Они помогают «дойти» до необходимого узла.

7. Опишите алгоритм удаления вершины дерева.

Алгоритм удаления вершины с ключом равным X из случайного дерева поиска состоит в следующем. Сначала нужно найти вершину с ключом равными X. Если найденная вершина имеет не более одного поддерева, то её просто удаляем.

8. Перечислите способы обхода дерева в глубину.
    - Прямой (pre-order)
    - Симметричный или поперечный (in-order)
    - В обратном порядке (post-order)

Вариант 11

```
class Node {
  constructor(data, parent) {
    this.data = data;
    this.parent = parent;
    this.children = [];
  }
}
```

Рисунок 1 - Класс дерева

```
class Tree {
  constructor(data) {
    this.root = new Node(data, null);
  }
  add(data, currentNode, index) {
    if (index == null) {
      rl.question("\nCHOOSE ROOT ELEMENT → ", (index) ⇒ {
        if (index ≠ currentNode.data) {
          for (let i = 0; i < currentNode.children.length; i++) {
            this.add(data, currentNode.children[i], index);
          }
        } else {
          currentNode.children.push(new Node(data, currentNode));
          tasks();
        }
      });
    } else {
      if (index ≠ currentNode.data) {
        for (let i = 0; i < currentNode.children.length; i++) {
          this.add(data, currentNode.children[i], index);
        }
      } else {
        currentNode.children.push(new Node(data, currentNode));
        tasks();
      }
    }
  }
 display(currentNode) {
    console.log("-------");
    console.log("ROOT : " + currentNode.data);
    for (let i = 0; i < currentNode.children.length; i++) {
      console.log("CHILDREN : " + currentNode.children[i].data);
    }
    for (let i = 0; i < currentNode.children.length; i++) {
      this.display(currentNode.children[i]);
    }
 }
```

Рисунок 2 - Базовые методы для работы с деревом

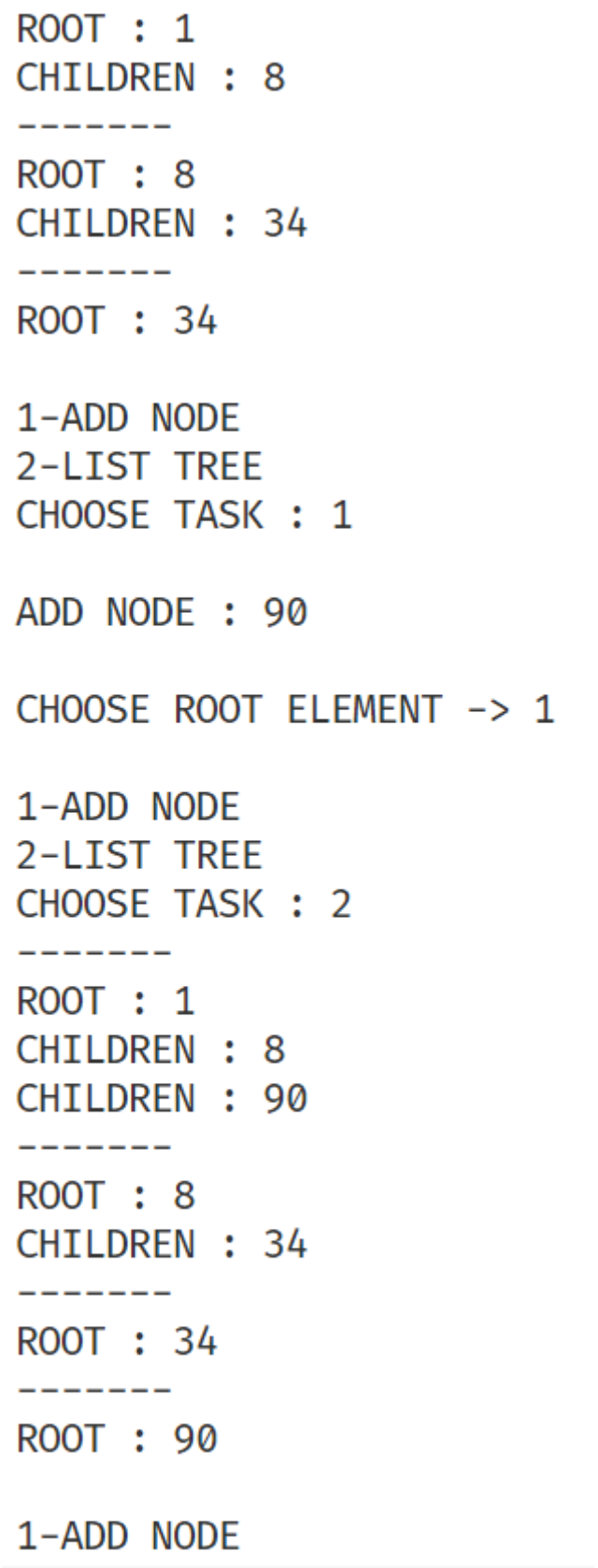

Рисунок 3 - Результат работы программы