

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по практической работе №3
по дисциплине «Алгоритмы и структуры данных»
Тема: Бинарные деревья

Студент гр. 9303

Скворчевский Б.С.

Преподаватель

Филатов А.Ю.

Санкт-Петербург

2020

Цель работы.

Ознакомиться с понятием бинарного дерева, изучить его особенности и реализовать программу, решающую поставленную задачу с помощью бинарного дерева.

Основные теоретические положения.

Дерево – конечное множество T , состоящее из одного или более узлов, таких, что

а) имеется один специально обозначенный узел, называемый *корнем* данного дерева;

б) остальные узлы (исключая корень) содержатся в $m \geq 0$ попарно не пересекающихся множествах T_1, T_2, \dots, T_m , каждое из которых, в свою очередь, является деревом. Деревья T_1, T_2, \dots, T_m называются *поддеревьями* данного дерева.

При программировании и разработке вычислительных алгоритмов удобно использовать именно такое *рекурсивное* определение, поскольку рекурсивность является естественной характеристикой этой структуры данных.

Наиболее важным типом деревьев являются *бинарные деревья*. Удобно дать следующее формальное определение. *Бинарное дерево* – конечное множество узлов, которое либо пусто, либо состоит из корня и двух непересекающихся бинарных деревьев, называемых правым поддеревом и левым поддеревом. Так определенное бинарное дерево *не* является частным случаем дерева.

Постановка задачи.

Вариант 1д.

Задано бинарное дерево b типа BT с типом элементов $Elem$. Для введенной пользователем величины E (**var** $E: Elem$):

- определить, входит ли элемент E в дерево b ;
- определить число вхождений элемента E в дерево b ;
- найти в дереве b длину пути (число ветвей) от корня до ближайшего узла с элементом E (если E не входит в b , за ответ принять -1).

Выполнение работы.

Программа принимает 2 строки из файла и записывает их в переменные `tree` (дерево) и `E` (элемент для поиска). Далее она выводит их на экран. После происходит проверка синтаксиса. Далее вызывается метод `SetElem`, который устанавливает элемент для поиска, метод `read_tree`, который в свою очередь вызывает метод `read_node`, считывающий дерево из строки, метод `search_elem`, который выполняет поиск элемента в дереве. Далее на экран выводится число вхождений элемента E в дерево и длина минимального пути до него.

Выводы.

Были изучены основные понятия бинарного дерева и его особенности. Была реализована программа, решающая поставленную задачу с помощью бинарного дерева.

ПРИЛОЖЕНИЕ А

ТЕСТИРОВАНИЕ

Таблица 1 - Примеры тестовых случаев

| № | Входные данные | Выходные данные | Комментарий |
|---|-------------------------------------|---|------------------------------|
| 1 | (a(b(d)(e))(c(d)(e))(f(i)(k))) d | Введено бинарное дерево: (a(b(d)(e))(c(d)(e))(f(i)(k))) Введён элемент для поиска: d Элемент d встречается в бинарном дереве 2 раз. Длина кратчайшего пути равна 2. | Программа работает корректно |
| 2 | (a(b(d)(e))(c(d)(e))(f(i)(k))) c | Введено бинарное дерево: (a(b(d)(e))(c(d)(e))(f(i)(k))) Введён элемент для поиска: c Элемент c встречается в бинарном дереве 1 раз. Длина кратчайшего пути равна 1. | Программа работает корректно |
| 3 | (a(b(d)(e(g(o)))))) o | Введено бинарное дерево: (a(b(d)(e(g(o)))))) Введён элемент для поиска: o Элемент o встречается в бинарном дереве 1 раз. Длина кратчайшего пути равна 4. | Программа работает корректно |
| 4 | (j(d(x)(k))y(m(f))(j)) f | Введено бинарное дерево: (j(d(x)(k))y(m(f))(j)) Введён элемент для поиска: f | Программа работает корректно |

| | | | |
|---|---------------------------------|--|------------------------------|
| | | <p>Элемент f встречается в бинарном дереве 1 раз.</p> <p>Длина кратчайшего пути равна 2.</p> | |
| 5 | $(j(d(x)(k))y(m(f))(j))$ x | <p>Введено бинарное дерево: $(j(d(x)(k))y(m(f))(j))$</p> <p>Введён элемент для поиска: x</p> <p>Элемент x встречается в бинарном дереве 1 раз.</p> <p>Длина кратчайшего пути равна 2.</p> | Программа работает корректно |