**работа № 2. деревья. бинарные деревья**

***Цель работы*** – Изучить способы узлового представления деревьев и методы их прохождения, получить практические навыки программирования задач с использованием деревьев.

***Внимание! Для первых трех пунктов подготовки к работе необходимо разработать только алгоритмы (на уровне псевдокода), а не программы. Программная реализация требуется только для четвертого пункта.***

***При программной реализации в данной работе ЗАПРЕЩЕНО использовать имеющиеся в среде программирования классы (библиотеки) для работы с деревьями.***

Подготовка к работе

1. Реализовать узловое представление бинарного дерева.

2. Разработать алгоритмы, реализующие прямое, обратное, симметричное и горизонтальное прохождения бинарного дерева. Для первых трех прохождений использовать как рекурсивные, так и нерекурсивные варианты. Для реализации горизонтального прохождения требуется очередь.

3. Разработать алгоритм прохождения в горизонтальном порядке леса, представленного с помощью естественного соответствия бинарным деревом.

4. Разработать алгоритмы и программы решения задач в соответствии с заданными вариантами.

*Служебные алгоритмы и процедуры* (*в отчет по ЛР можно не включать*):

С1. Разработать процедуру ввода и формирования узлового представления бинарного дерева.

С2. Разработать процедуру вывода узлового представления бинарного дерева с целью визуального контроля структуры дерева.

Варианты заданий (разработать нерекурсивные алгоритмы и программы!)

Номер варианта *N* определяется по следующей формуле (*i* – номер студента по списку группы, *k* – число задач, **mod** – остаток целочисленного деления)

*N* = (*i* – 1) **mod** *k* + 1.

1. Копирование бинарного дерева.

2. Преобразование бинарного дерева в симметрично прошитое бинарное дерево.

3. Реализовать прохождение прошитого бинарного дерева в прямом порядке.

4. Реализовать прохождение прошитого бинарного дерева в симметричном порядке.

5. Создание нового бинарного дерева, являющегося зеркальным отражением исходного (все левые поддеревья становятся правыми поддеревьями и наоборот).

6. Определение числа вхождений заданного элемента в дерево.

7. Вычисление среднего арифметического всех элементов непустого дерева.

8. Поменять местами максимальный и минимальный элементы непустого дерева, все элементы которого различны.

9. Вывод элементов из всех листьев дерева.

10. Определение высоты непустого дерева, т. е. число ребер в самом длинном из путей от корня до листьев.

11. Подсчет числа вершин на *n*-ом уровне непустого дерева (корень считается вершиной нулевого уровня).

***Бонусные задачи* (+2 балла за задачу):**

Арифметическое выражение может быть представлено бинарным деревом следующим образом. Корень такого дерева содержит операцию, которая должна быть применена к результатам вычисления выражений, представленных левым и правым поддеревьями. Вершина, представляющая операцию, всегда имеет два непустых поддерева, а вершина, представляющая операнд, – два пустых поддерева. Таким образом, операции сопоставляются внутренним вершинам, а операнды – листьям. Следует обратить внимание на то, что не требуются скобки, поскольку порядок операций определяется структурой дерева. На рис. 1 показано представление бинарным деревом выражения

((*a* + *b*) × *c* – (*d* – *e*))↑(*f* + *g*).



Рис. 1. Бинарное дерево для выражения ((*a* + *b*) × *c* – (*d* – *e*))↑(*f* + *g*)

Прохождение такого дерева в прямом порядке дает префиксную форму выражения, прохождение в обратном порядке – постфиксную форму. При прохождении в симметричном порядке получается инфиксная форма выражения. Однако, поскольку дерево не содержит скобок, выражение, в инфиксной форме которого требуются скобки, не может быть восстановлено простым прохождением в симметричном порядке.

***Задачи***(можно ограничиться 4-мя арифметическими операциями +, –, \*, /):

Б.1. По заданному выражению в инфиксной форме построить соответствующее бинарное дерево. Используя соответствующие прохождения получить постфиксную и префиксную формы.

Б.2. По заданному выражению в постфиксной форме построить соответствующее бинарное дерево. Используя соответствующие прохождения получить инфиксную и префиксную формы.

Б.3. По заданному выражению в префиксной форме построить соответствующее бинарное дерево. Используя соответствующие прохождения получить инфиксную и постфиксную формы.

Б.4. По заданному выражению в инфиксной форме построить соответствующее бинарное дерево. Вычислить значение выражения, используя прямой и обратный порядки прохождения.

Б.5. По заданному выражению в постфиксной форме построить соответствующее бинарное дерево. Вычислить значение выражения, используя прямой и симметричный порядки прохождения.

Б.6. По заданному выражению в префиксной форме построить соответствующее бинарное дерево. Вычислить значение выражения, используя обратный и симметричный порядки прохождения.

Содержание отчета

1. Цель работы.

2. Информация в соответствии с подготовкой к работе.

3. Скриншот с результатами работы программы.

4. Выводы.