Лабораторная работа № 9

**Тема:** Деревья

**Цель работы:** получить навыки работы с деревьями, изучить алгоритмы поиска в деревьях.

Краткие теоретические сведения

Дерево — это один из частных случаев графа. Древовидная модель может быть весьма эффективна в случае представления динамических данных, особенно тогда, когда у разработчика стоит цель быстрого поиска информации, в тех же базах данных, к примеру. Еще древом называют структуру данных, которая представляет собой совокупность элементов, а также отношений между этими элементами, что вместе образует иерархическую древовидную структуру.

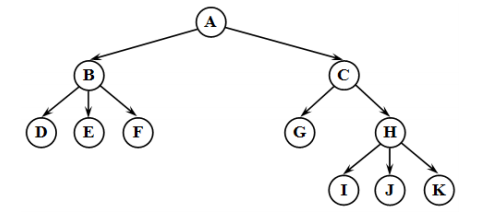


Рисунок 1 – пример дерева

Каждый элемент — это вершина или **узел** дерева. Узлы, соединенные направленными дугами, называются **ветвями**. Начальный узел — это **корень** дерева (корневой узел). **Листья**— это узлы, в которые входит 1 ветвь, причем не выходит ни одной.

С общей терминологией можно ознакомиться на рисунке 2:

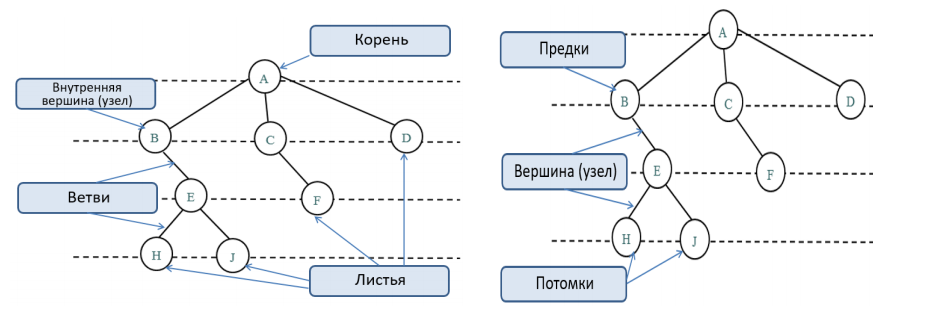


Рисунок 2 – обозначение частей дерева

Какие **свойства** есть у каждого древа:

— существует узел, в который не входит ни одна ветвь;

— в каждый узел, кроме корневого узла, входит 1 ветвь.

На практике деревья нередко применяют, изображая различные иерархии. Очень популярны, к примеру, генеалогические древа — они хорошо известны. Все узлы с ветвями, исходящими из единой общей вершины, являются потомками, а сама вершина называется предком (родительским узлом). Корневой узел не имеет предков, а листья не имеют потомков.

Также у дерева есть **высота (глубина)**. Она определяется числом уровней, на которых располагаются узлы дерева. Глубина пустого древа равняется нулю, а если  речь идет о дереве из одного корня, тогда единице. В данном случае на нулевом уровне может быть лишь одна вершина – корень, на 1-м – потомки корня, на 2-м – потомки потомков корня и т. д.

**Лабораторная работа** рассчитана на 4 часов аудиторных занятий и состоит в изучении теоретического материала и получении практических навыков при работе с деревьями в python. Сдача лабораторной работы заключается в ответах на контрольные вопросы.

**Содержание отчета (в электронном виде):**

1. Название и цель работы
2. Ответы на контрольные вопросы
3. Скрипты выполненных заданий в Python.

**Задание:**

Выполнить индивидуальный вариант задания по работе с деревьями.

Индивидуальные варианты:

1. Реализуйте алгоритм обхода в глубину на дереве.
2. Реализуйте алгоритм обхода в ширину на дереве.
3. Создайте бинарное дерево поиска из отсортированного массива.
4. Напишите функцию, которая проверяет, является ли дерево бинарным деревом поиска.
5. Реализуйте функцию, которая возвращает все пути от корня до листьев.
6. Реализуйте алгоритм поиска в бинарном дереве.
7. Напишите функцию, которая находит минимальное и максимальное значение в бинарном дереве поиска.
8. Реализуйте функцию, которая добавляет элемент в бинарное дерево поиска.
9. Реализуйте функцию, которая удаляет элемент из бинарного дерева поиска.
10. Напишите функцию, которая находит k-й по величине элемент в бинарном дереве поиска.
11. Реализуйте функцию, которая находит общего предка двух узлов в бинарном дереве.
12. Напишите функцию, которая определяет, является ли одно дерево поддеревом другого.
13. Напишите функцию, которая возвращает глубину дерева.
14. Реализуйте функцию, которая возвращает ширину дерева.
15. Напишите функцию, которая возвращает диаметр дерева.
16. Реализуйте функцию, которая возвращает количество узлов на каждом уровне дерева.
17. Напишите функцию, которая возвращает количество узлов на каждом уровне в бинарном дереве поиска.
18. Реализуйте функцию, которая определяет, является ли дерево сбалансированным.
19. Напишите функцию, которая определяет, является ли дерево симметричным.
20. Реализуйте функцию, которая возвращает зеркальное отображение дерева.

**Контрольные вопросы**

1. Какие основные преимущества использования деревьев поиска в структурах данных?
2. Можете ли вы объяснить, как деревья используются для поиска информации в базах данных?
3. Какие типы деревьев поиска существуют и в чем их особенности?