

Практическое занятие №2

Густов Владимир Владимирович
gutstuf@gmail.com

Цитата дня: Средства не виноваты в том, что их безграмотно используют.

(с) Эдсгер Вибе Дейкстра

Структуры данных

массив

строка

очередь и стек

вектор

список

таблица (словарь, хеш-таблица)

множество

дерево

граф*

Дерево

Нелинейная структура данных, являющаяся связным ациклическим (без циклов) графом.

Узел – элемент дерева;

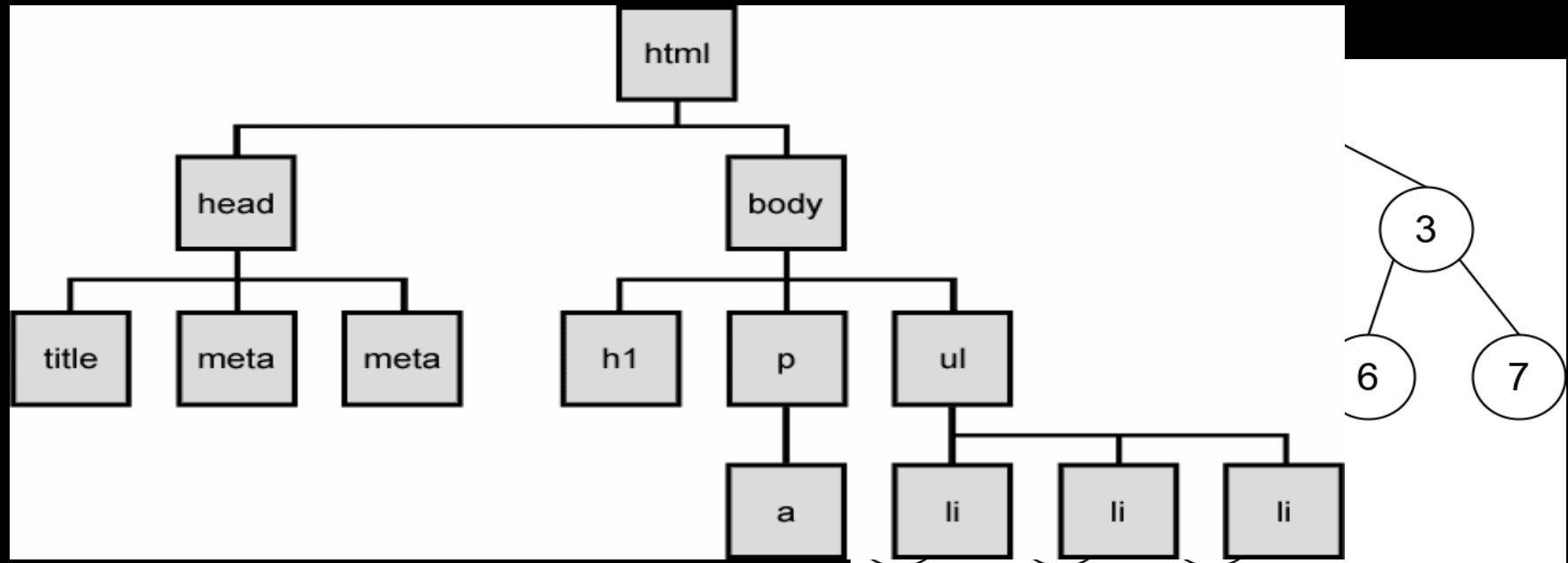
Вершина – узел, имеющий дочерние элементы;

Ветвь – связь между узлами;

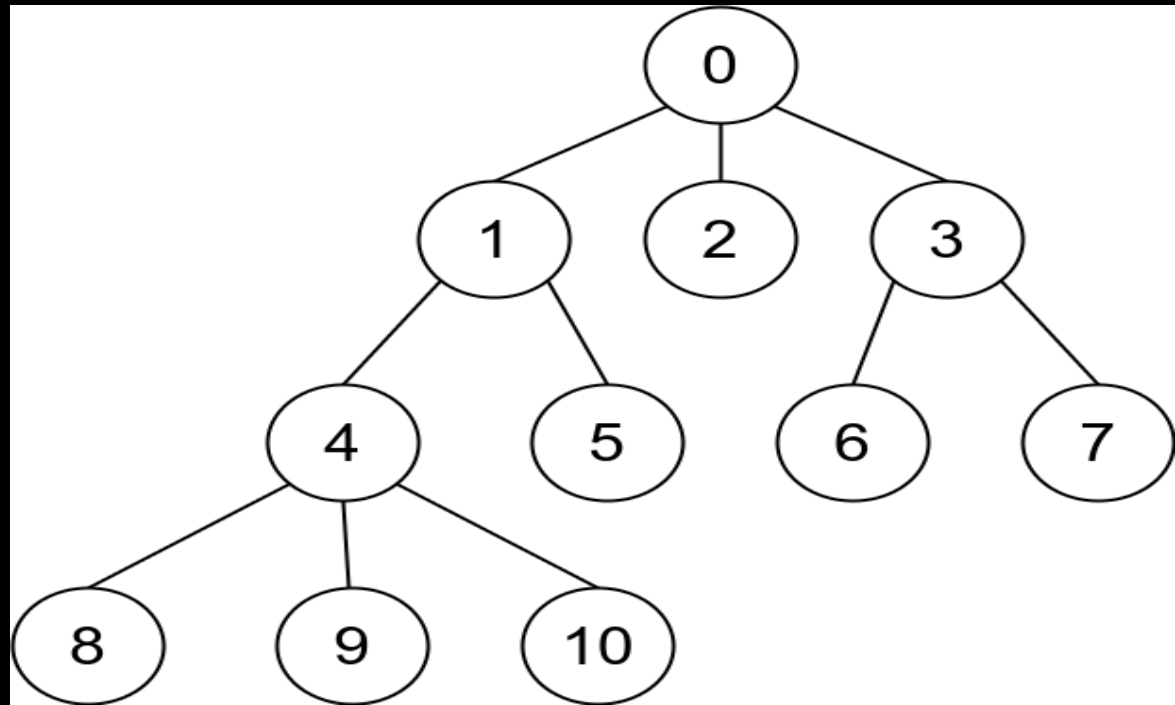
Корень - самый верхний узел дерева;

Лист – узел, не имеющий дочерних элементов;

Лес – упорядоченное множество деревьев;



- **Уровень** – «поколение» узла. Если корневой узел находится на уровне 0, то его следующий дочерний узел находится на уровне 1, его внук - на уровне 2, и так далее.
- **Ширина** – максимальное кол-во узлов расположенных на одной высоте;
- **Высота** – длина самого дальнего пути к листу;
- **Глубина узла** – длина пути от текущего узла к корню;
- **Степень** – количество потомков у узла.

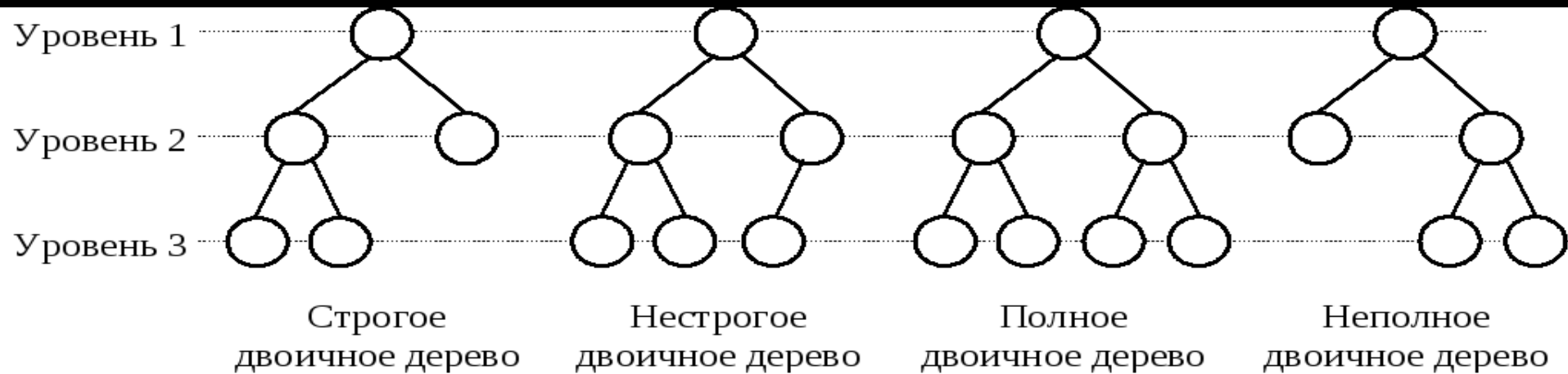


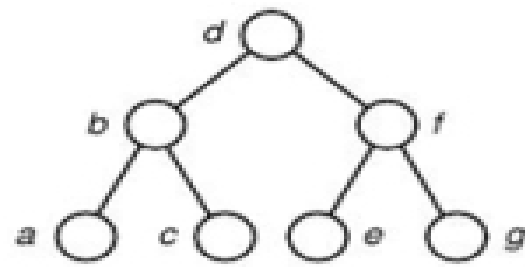
•BSP-дерево	Дерево со штрафами
•Абстрактное синтаксическое дерево	Расширяющееся дерево
•Двоичное дерево поиска (Binary search tree)	Прошитое двоичное дерево
•Декартово дерево или дерамида	Дерево ван Емде Боаса
•Сбалансированное дерево	Ориентированный граф
•UB-дерево	Направленный ациклический граф
•АВЛ-дерево	Бинарная диаграмма решений
•Дерево Фибоначчи	Дерево отрезков
•B-дерево	Интервальное дерево
•2-3-дерево	Куча
•2-3-4-дерево	Двоичная куча
•B+-дерево	Биномиальная куча
•B*-дерево	Фибоначчиева куча
•Красно-чёрное дерево	Сливаемая куча
	2-3-куча
	Мягкая куча
	Дерево разбора
	Квадродерево и Октодерево
	Связка
	Сильноветвящееся дерево, многопутевое дерево (M-Way Tree)

Бинарное (двоичное) дерево

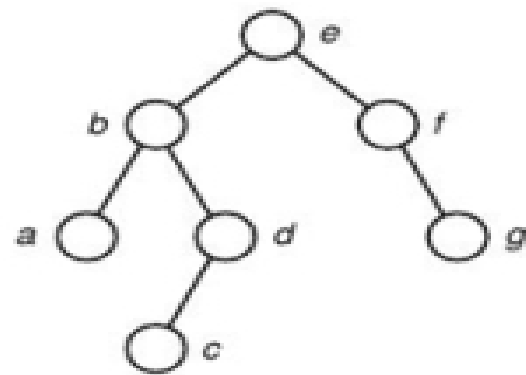
Свойства:

- вставка/удаление элемента — $O(\log n)$;
- поиск элемента — $O(\log n)$;
- каждый узел имеет не более двух потомков;
- строгое БД — узел имеют степень два или ноль;
- полное БД — каждый узел имеет по 2 дочерних узла;
- сбалансированное БД — для каждой вершины, кол-во вершин в левом и в правом поддереве различаются не более чем на единицу;
- вырожденное БД — каждый узел имеет всего один дочерний узел.

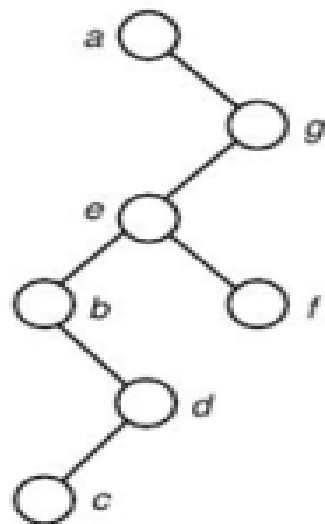




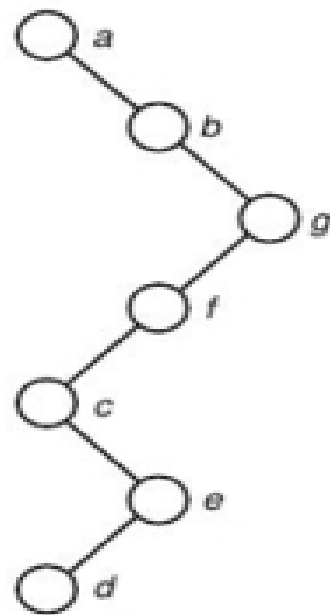
(a)



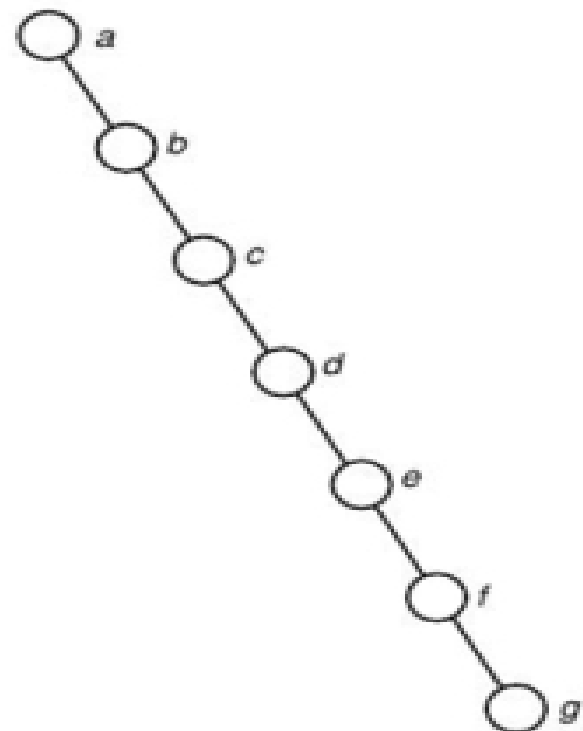
(b)



(c)



(d)



```
class Tree {
    T data      {0};          // данные хранимые узлом
    Tree* left  {nullptr};    // указатель на левый дочерний эл-т
    Tree* right {nullptr};    // указатель на правый дочерний эл-т

    Tree* CreateNode(const T& val) {
        auto* node = new Tree(val);
        return node;
    }
public:
    void AddLeftNode(const T& val) {
        this->left = CreateNode(val);
    }

    void AddRightNode(const T& val) {
        this->right = CreateNode(val);
    }

    ~Tree() {
        // освобождение памяти дочерних узлов
    }
};
```


Построение бинарного дерева поиска (BST)

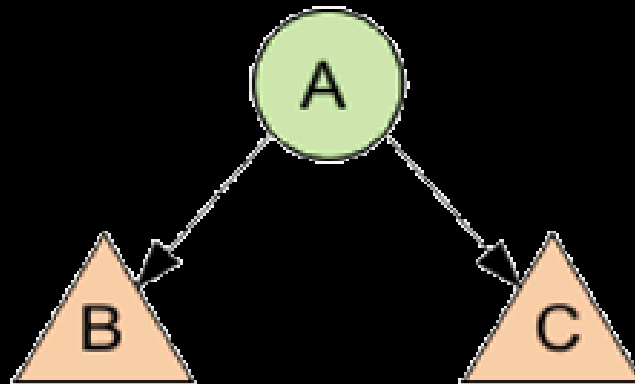
- любое значение меньшее значения текущего узла становится левым дочерним узлом или переходит в левое поддерево;
- любое значение большее или равное значению текущего узла становится правым дочерним узлом или переходит в правое поддерево;

Дана последовательность:

8 2 4 3 5 1 10 9 12 13 8

Алгоритмы обхода дерева

- Обход дерева сверху вниз (в прямом порядке): А, В, С — префиксная форма.
- Обход дерева в симметричном порядке (слева направо): В, А, С — инфиксная форма.
- Обход дерева в обратном порядке (снизу вверх): В, С, А — постфиксная форма.



Удаление узлов

- если у узла нет потомков — удаляем его;
- если узел имеет 1 потомка — заменяем родителя потомком;
- если узел имеет 2 потомка — выбираем узел (из дочерних) с минимальным значением и заменяем им родителя;
- удаляя узел из поддерева необходимо заменить его элементом, который будет строго меньше правой части удалённого узла, и больше (или равен) левой части удалённого узла.

<https://youtu.be/B34LP4TS824>