

Деревья. Бинарное дерево.



ПЛАН ЗАНЯТИЯ

1. Повторение
2. Деревья
3. Структура Бинарного дерева
4. Практическая работа
5. Домашнее задание



TEL-RAN
by Starta Institute

1

ПОВТОРЕНИЕ ИЗУЧЕННОГО

Разберем задачу

“Подарки”

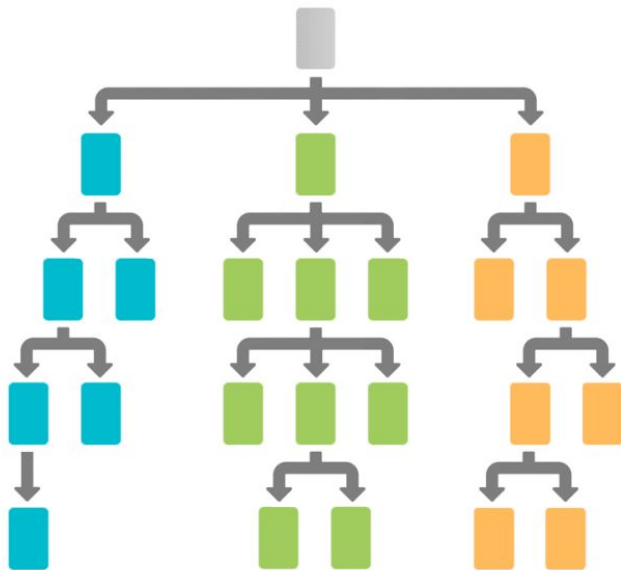


2

Деревья

Древовидные структуры

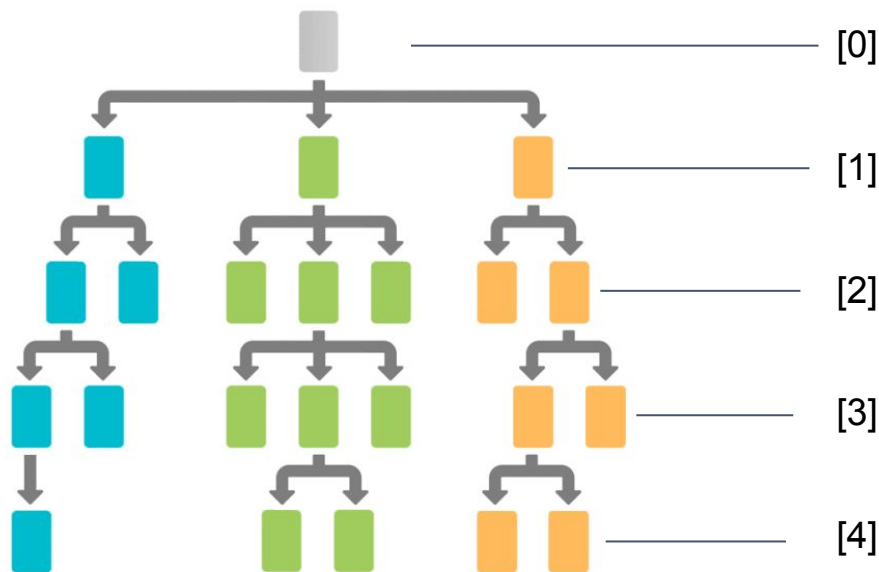
Дерево является нелинейной и представляет собой иерархическую структуру данных, состоящую из набора узлов, так что каждый узел дерева хранит значение и список ссылок на другие узлы («потомки»).



Нелинейная структура данных

Данные в дереве не хранятся последовательно, т. е. не линейно.

Вместо этого они расположены на нескольких уровнях, или можно сказать, что это иерархическая структура.



Основные термины в древовидной структуре данных

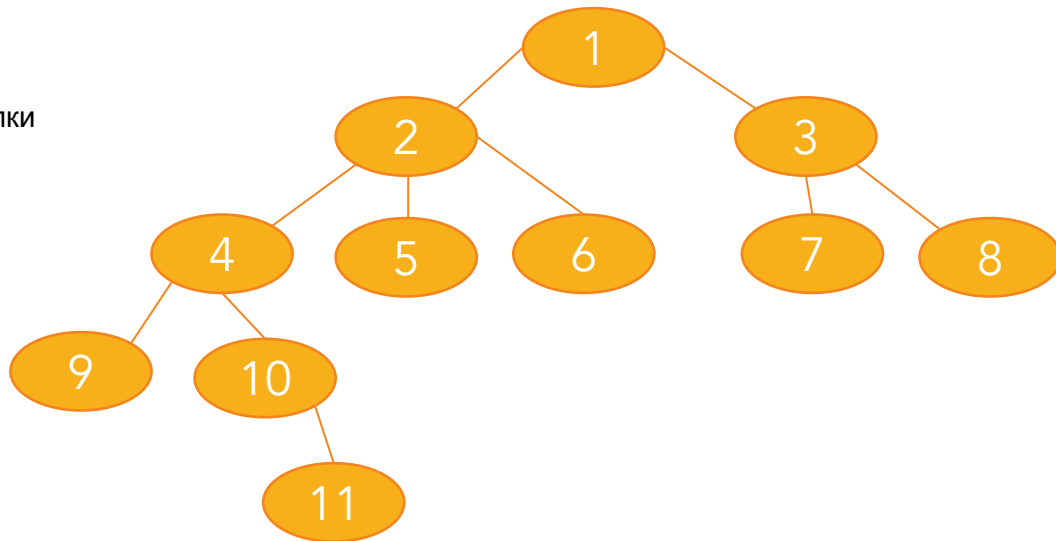
Корень: Верхний узел дерева.

Узел: Элемент дерева, содержащий данные и ссылки на дочерние узлы.

Лист: Узел, не имеющий дочерних узлов.

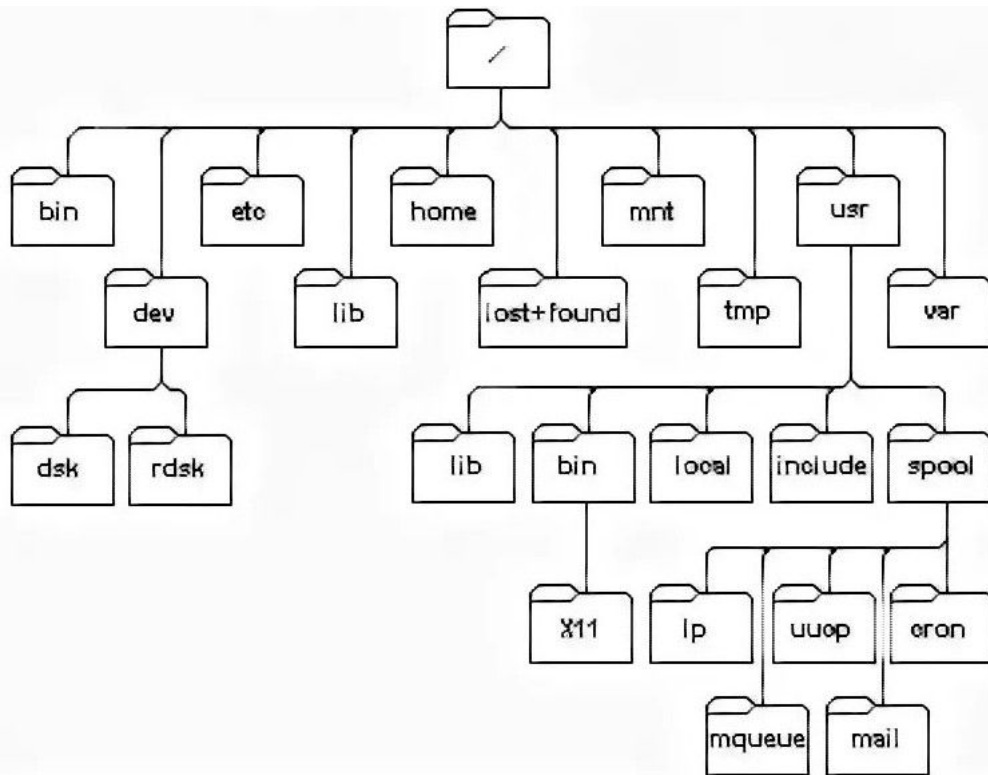
Глубина: Количество уровней от корня до узла.

Высота: Максимальная глубина дерева.



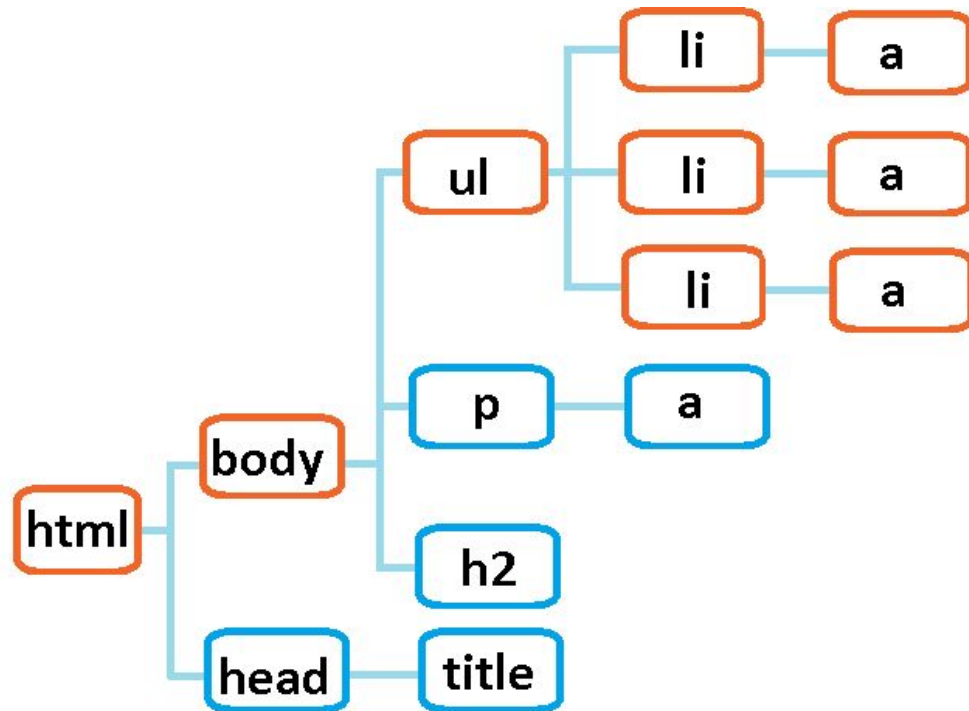
Использование на практике

Дерево каталогов



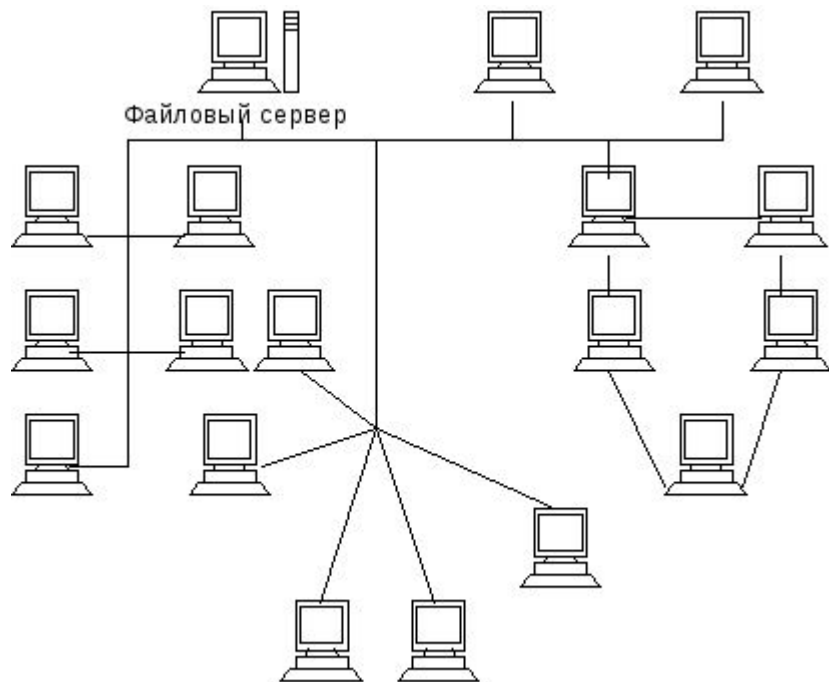
Использование на практике

HTML - дерево тегов



Использование на практике

Компьютеры в сети



Использование на практике

Ваши варианты?



Использование на практике

- **Поиск и сортировка данных:** Деревья бинарного поиска используются в базах данных и поисковых системах для быстрого поиска и сортировки информации.
- **Представление иерархических структур:** Файловые системы, организационные диаграммы и XML-документы могут быть представлены с помощью бинарных деревьев.
- **Компиляторы и интерпретаторы:** Синтаксические деревья, основанные на бинарных деревьях, используются для анализа и выполнения кода.
- **Сжатие данных:** Деревья Хаффмана, разновидность бинарных деревьев, применяются для эффективного сжатия данных.
- **Маршрутизация сетевых пакетов:** Бинарные деревья используются для оптимизации маршрутизации сетевых пакетов в компьютерных сетях.
- **Игры:** Деревья решений используются в алгоритмах искусственного интеллекта для принятия решений в играх.

Типы древовидных структур

General tree

Общая древовидная структура данных не имеет ограничений на количество узлов. Это означает, что родительский узел может иметь любое количество дочерних узлов.

Binary tree

У узла бинарного дерева может быть максимум два дочерних узла.

Balanced tree

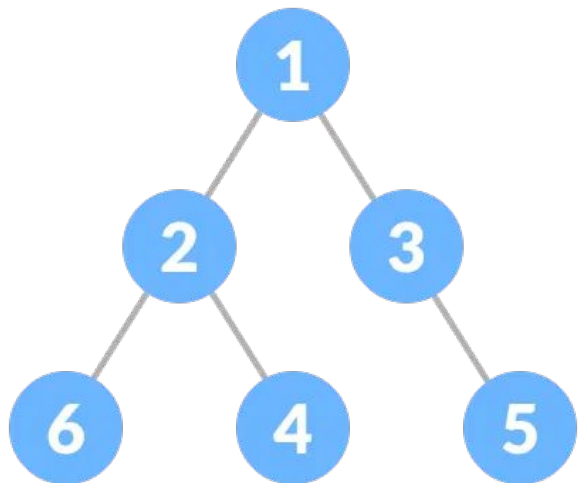
Если высота левого поддерева и правого поддерева равна или отличается не более чем на 1, дерево называется сбалансированным.

Binary search tree (BST)

Это нелинейная структура данных, которая показывает, что значение левого узла меньше, чем у его родителя, а значение правого узла больше, чем у его родителя.

Типы древовидных структур

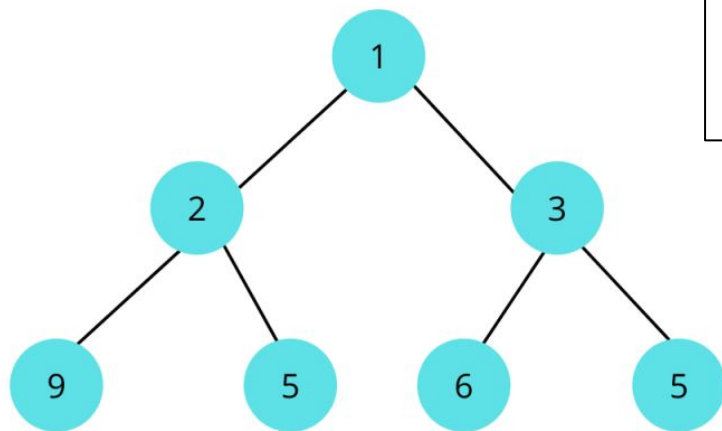
Бинарное дерево



У узла бинарного дерева может быть максимум два дочерних узла.

Типы древовидных структур

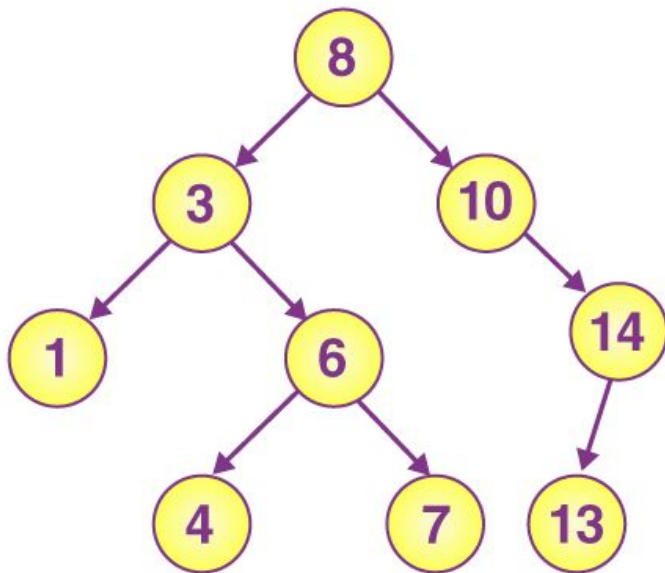
Сбалансированное Бинарное дерево



Если высота левого поддерева и правого поддерева равна или отличается не более чем на 1.

Типы древовидных структур

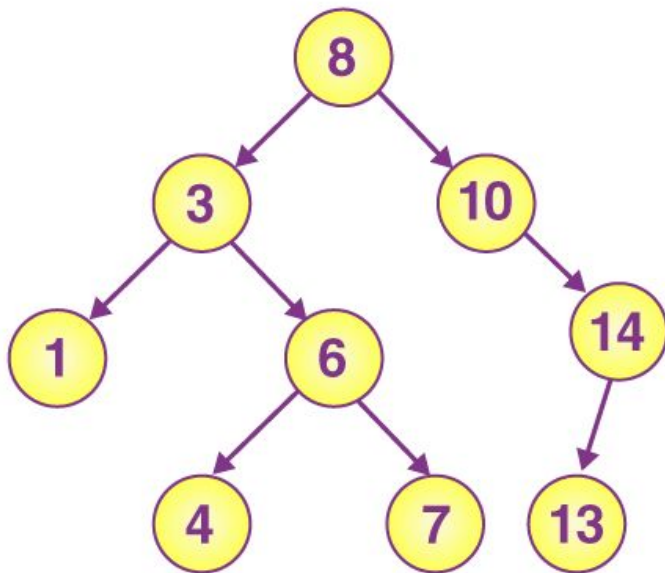
Бинарное дерево поиска (BST)



Что особенного в этом дереве?

Типы древовидных структур

Бинарное дерево поиска (BST)



- Значение левого потомка меньше значения родительского узла.
- Значение правого потомка больше значения родительского узла.

Экспресс-опрос

- **Вопрос 1.**

Почему дерево считается не линейной структурой?

- **Вопрос 2.**

Сколько может быть потомков у узла в обычном дереве?

- **Вопрос 3.**

Сколько может быть потомков у узла в Бинарном дереве?



Основные операции с деревом

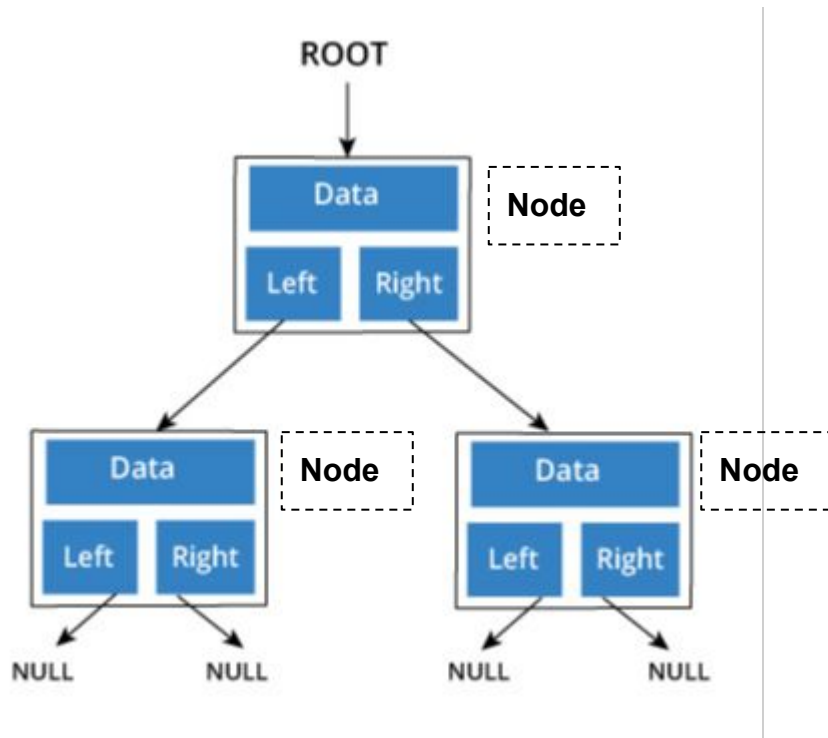
- Create – создать дерево в структуре данных.
- `insert` – вставляет данные в дерево.
- `search` – ищет определенные данные в дереве.
- `delete` – удаляет узел из дерева.
- `min` – находит узел с минимальным ключом
- `max` – находит узел с максимальным ключом



3

Структура Бинарного дерева

Узлы бинарного дерева

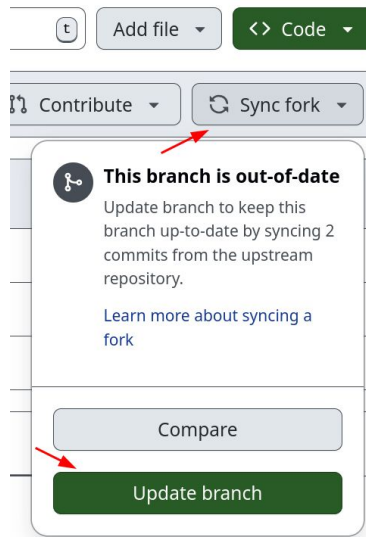


4

Практика

Затяните изменения с GitHub

1. Сохраните изменения в проекте(если есть) в отдельную ветку
git checkout -b <you_branch>
git add .
git commit -m "комментарий к коммиту"
2. Синхронизируйте Fork репозитория на своем аккаунте:
3. Затяните изменения из origin
git checkout
git pull



Практическое задание

Работаем с:

org.telran.[lecture_12_trees](#)

- Задача: practice.[BST](#)



5

Домашнее задание

Домашнее задание

Домашнее задание выполните в файле `lecture_12_trees.practice.BST`

1. Протестируйте методы, реализованные на занятии:

- `insertNode()` – доработка с дубликатами
- `searchNode()` – полная реализация метода
- `length()` – полная реализация метод

Если найдете ошибки – исправьте.

2. Завершить реализацию методов:

- `min()`
- `max()`



Полезные ссылки

- [How to Use Trees \(The Java™ Tutorials > Creating a GUI With Swing > Using Swing Components\)](#) (JAVA)
- [JavaScript Data Structures - Tree - 30 seconds of code](#) (JS)
- [Tree \(data structure\) - Wikipedia](#)

ЗАКЛЮЧЕНИЕ

