

«ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ»

ЛАБОРАТОРНАЯ РАБОТА №4

Бинарные деревья поиска

Автор: Гуков Сергей Юрьевич

Версия от 11.01.2025

Для допуска к экзамену нужно сдать три первых лабораторных работы (в любые сроки) и посетить минимум 50% лекций. Для получения автоматом оценки за экзамен необходимо посетить также минимум 50% лекций и сдать соответствующее количество лабораторных работ в установленные для них сроки. Если хотя бы одна лабораторная защищена и загружена позже установленного срока, то студент отправляется на экзамен без возможности получить оценку автоматом. Количество баллов за лабораторные зависит от полноты выполнения задания и качества защиты, они косвенно влияют на итоговую оценку за экзамен. Для получения автоматом оценки «Удовлетворительно» необходимо защитить лабораторные работы №1, №2, №3. Для получения автоматом оценки «Хорошо» помимо предыдущих требований необходимо выполнить лабораторную работу №4. Для получения автоматом оценки «Отлично» помимо предыдущих требований необходимо выполнить лабораторную работу №5. Также на экзамене есть возможность получить оценку на один балл выше относительно количества защищенных лабораторных.

Цель

Изучить структуру данных бинарного дерева поиска, разработать расширенный класс для работы с ним и получить практический навыки использования и визуализации этой структуры данных.

Задание

Необходимо разработать программу, которая генерирует на вход случайную числовую последовательность случайного размера и строит из этой последовательности бинарное дерево.

Программа также должна иметь следующий функционал:

- удаление / добавление узла с выводом статуса операции (например, успешное добавление, ошибка валидации, узел не найден);
- поиск узла в дереве;
- в консольном варианте интерфейса хоть какое-то наглядное и понятное отображение родителя и левого и правого потомков (например, через ASCII-графику в консоли);
- графическое представление дерева (опционально);
- поиск узла в заданном поддереве (опционально);
- балансировка дерева (опционально, божественный уровень).

Проект может быть выполнен либо в качестве консольного приложения (тогда обязателен командно-текстовый интерфейс), либо иметь графический пользовательский интерфейс (User Interface, UI), а также может быть написан на любом языке программирования.

Требования к структуре проекта

- ✓ Применение принципов ООП (наследования, инкапсуляции, полиморфизма, абстракции) и SOLID
- ✓ Дружелюбный командно-текстовый либо графический пользовательский интерфейс

Отчет должен содержать

1. Цель работы
2. Задание
3. Краткое описание хода разработки и назначение используемых технологий
4. Исходный код программы (с комментариями в необходимых местах)
5. Результаты работы программы с примерами разных сценариев (скриншоты)

6. Выводы

При оценке выполнения работы будут учитываться грамотность оформления исходного кода, работоспособность программы и соответствие отчета правилам оформления текстовых документов по ГОСТ 7.32-2017. Титульные листы лабораторных работ представлены на сайте ГУАП (<https://guap.ru/standart/doc>).

Полезные ссылки

- Пирамиды и бинарные деревья поиска (<https://github.com/netology-code/algo-homeworks/blob/master/tasks/Trees.md>)
- Бинарные деревья поиска и рекурсия – это просто (<https://habr.com/ru/post/267855/>)
- Бинарные деревья (https://ru.hexlet.io/courses/algorithms-trees/lessons/binary/theory_unit)
- Дерево поиска, наивная реализация — Викиконспекты (https://neerc.ifmo.ru/wiki/index.php?title=Дерево_поиска,_наивная_реализация)
- Бинарные деревья | Алгоритмы на деревьях (https://ru.hexlet.io/courses/algorithms-trees/lessons/binary/theory_unit)
- Балансировка | Алгоритмы на деревьях (https://ru.hexlet.io/courses/algorithms-trees/lessons/balancing/theory_unit)
- Балансировка бинарного дерева поиска (<https://du-blog.ru/40>)
- Балансировка АВЛ-дерева – Программирование на C, C# и Java (<https://vscode.ru/articles/balansirovka-avl-dereva.html>)
- AVL-деревья (<https://www.rsdn.org/article/alg/bintree/avl.xml>)