

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«Алгоритми та структури даних»

Лабораторна робота № 2
«Бінарне дерево пошуку. Червоно-чорне дерево. AVL-дерево.»

Виконав:	Шевчук Максим Юрійович	Перевірів:	Бичков Олексій Сергійович
Група	ІПЗ-12/1	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		

2022

Умова задачі

Використовуючи структуру даних: **бінарне дерево пошуку, червоно-чорне дерево та AVL-дерево** реалізувати такі алгоритми:

1. Вставка вузла в дерево
2. Видалення вузла із дерева
3. Перефарбування двох однакових вузлів братів у протилежний колір
4. Лівий, правий повороти

Аналіз задачі

Для виконання роботи необхідно ознайомитись з теорією стосовно побудови бінарних дерев, оскільки саме вони є основою для подальшої побудови **Червоно-Чорного дерева**, та **AVL-дерева**.

Всі ці види дерев використовуються при необхідності зберігання значних масивів даних у відсортованому вигляді, без необхідності постійно проводити їх сортуванням. Ця задача покладається на структуру даних Бінарне Дерево Пошуку. Червоно-Чорне дерево и AVL-дерево мають таке саме завдання, але їх також, алгоритми, що в них закладені мають ціль пришвидшити пошук даних в цих структурах за допомогою їх балансування, таким чином зменшуючи час пошуку, незалежно від того, шукаємо ми елемент в правому чи в лівому піддереві.

Структура основних вхідних та вихідних даних

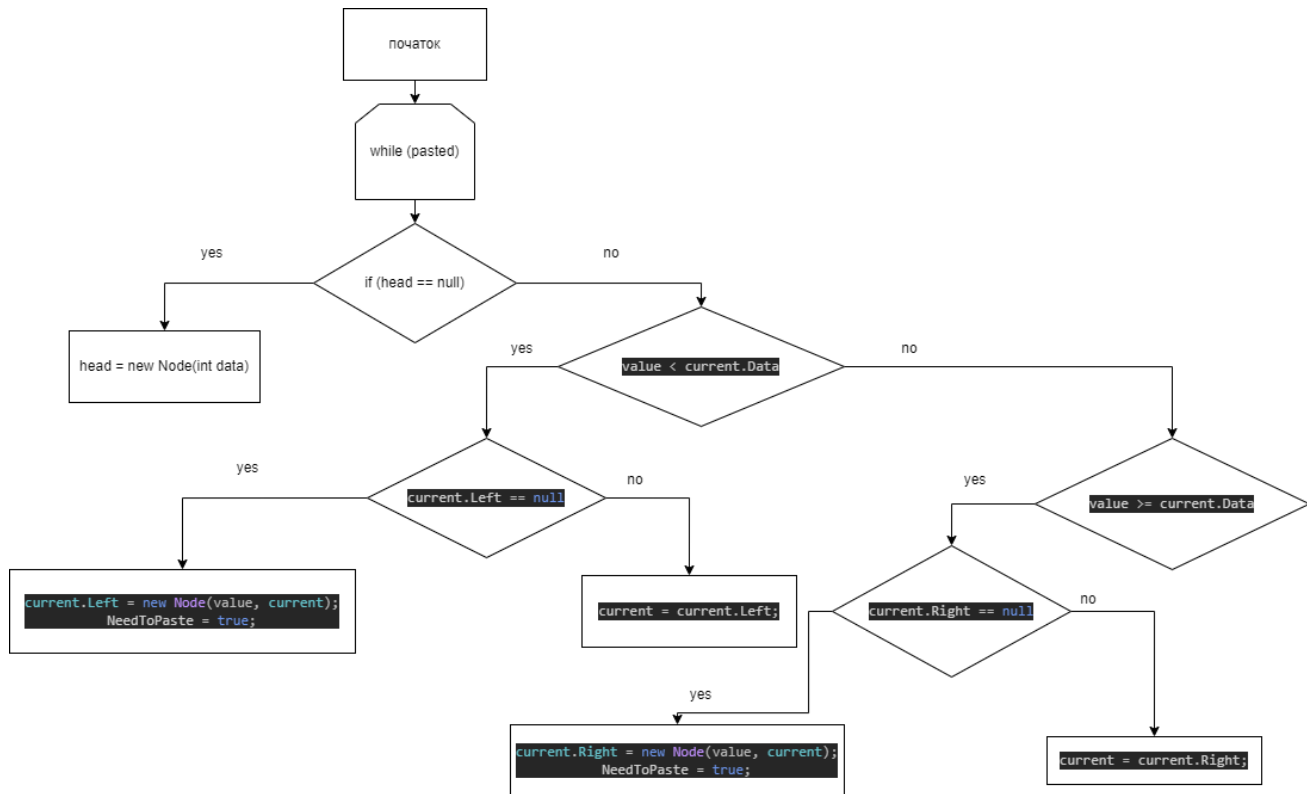
Вхідні дані, якими заповнюються дерева є цілі числа 1-1000. Це є найгіршим випадком для цих дерев, оскільки, відповідно до алгоритму цих дерев, ми отримаємо одне велике піддерево, яке по своїй структурі буде нагадувати звичайний масив, і час пошуку в ньому буде довгий.

Цю проблему можуть вирішити лише Червоно-Чорне дерево та AVL-дерево, завдяки своїм алгоритмам балансування.

На виході ми отримаємо структуру даних Бінарне дерево/Червоно-Чорне дерево/AVL-дерево, де всі наші вхідні дані будуть відсортовані.

Алгоритм розв'язання задачі

Для початку необхідно вставити елементи відповідно до правил вставки у бінарне дерево пошуку



1. При вставці у Червоно-Чорне дерево, за необхідності, виконуються повороти та перефарбування. Середня кількість поворотів при найгіршому варіанті вставки - 3.
2. При вставці у AVL-дерево, за необхідності, виконуються великі та малі повороти.

Текст програми

[Текст програми наявний на GitHub](#)

Набір тестів

Набір тестів - послідовність чисел від 1 до 1000.

В результаті, була отримана така порівняльна таблиця

Структура	Операція	Час
Бінарне дерево пошуку	Додавання 1000 елементів	0066326мс
Червоно-Чорне дерево	Додавання 1000 елементів	0006888мс
AVL-дерево	Додавання 1000 елементів	0156684мс
Бінарне дерево пошуку	Пошук максимального елемента	0001557мс
Червоно-Чорне дерево	Пошук максимального елемента	0001212мс
AVL-дерево	Пошук максимального елемента	0005016мс
Бінарне дерево пошуку	Видалення всіх елементів	0006334мс
AVL-дерево	Видалення всіх елементів	0119343мс

Висновок

В результаті виконання лабораторної роботи №2 було реалізовано такі структури даних: Бінарне дерево пошуку, Червоно-Чорне дерево, AVL-дерево та були написані алгоритми, що забезпечують їх роботу.

Було отримано наступний результат: бінарне дерево пошуку показує середню часову ефективність при виконанні різноманітних операцій. Найкраще себе показує червоно-чорне дерево, яке має виграв у часі при будь-якій операції. Найгірше себе показує AVL-дерево.