МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Львівський національний університет імені Івана Франка

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАТИКИ

Кафедра дискретного аналізу та інтелектуальних систем

**Звіт з лабораторної роботи №11**

**“дерево avl”**

Роботу виконав:

**Тимчишин Ярема Андрійович**

Студент групи Пмі-13

Перевірила:

**Корольчук Світлана Ярославівна,** завідувач Лабораторії програмування, асистент кафедри програмуванняЛьвівського національного

університету імені Івана Франка

Львів – 2022

**ЗМІСТ**

ВСТУП 3

РОЗДІЛ 1 10

***1.1. Створення клас “вузол”*** 10

***1.2. Деякі допоміжні функції*** 10

***1.3. Створення функцій для повороту вправо та вліво*** 11

***1.4. Функція для знаходження балансу*** 11

***1.5. Функція для додавання нових вузлів*** 12

***1.6. Функція яка повертає вузол з найменшим значенням*** 13

***1.7. Функція для вилучення вузлів*** 13

***1.8. Функція для друку дерева*** 14

РОЗДІЛ 2 15

***2.1. Створення дерева*** 15

***2.2. Результат*** 15

ВИСНОВКИ 16

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 16

# ВСТУП

Дерево називається збалансованим по висоті (ЛВЛ-деревом) тоді і лише тоді, коли для кожного вузла висота його двох піддерев відрізняється не більше ніж на 1.

Критерій їх збалансованості виявляється найкращим для реалізації операцій сортування та пошуку інформації. Варто відзначити, що всі ідеально збалансовані дерева є також ABJI-деревами.

Це визначення не лише просте, але також приводить до легко виконуваного балансування, а середня довжина пошуку залишається практично такою ж, як в ідеально збалансованого дерева.

Зі збалансованими деревами можна виконувати наступні операції за 0(log2n) одиниць часу, навіть у найгіршому випадку:

- знайти вузол із заданим значенням;

- включити вузол із заданим значенням;

- видалити вузол із заданим значенням.

Це є прямим наслідком теореми, доведеної Адельсоном — Вельским і Ландісом, яка стверджує, що збалансоване дерево ніколи не буде більше ніж на 45% вище від відповідного ідеального збалансованого дерева, незалежно від кількості вузлів.

Якщо позначити висоту збалансованого дерева з п вузлами через hB(n), то справедливе співвідношення:

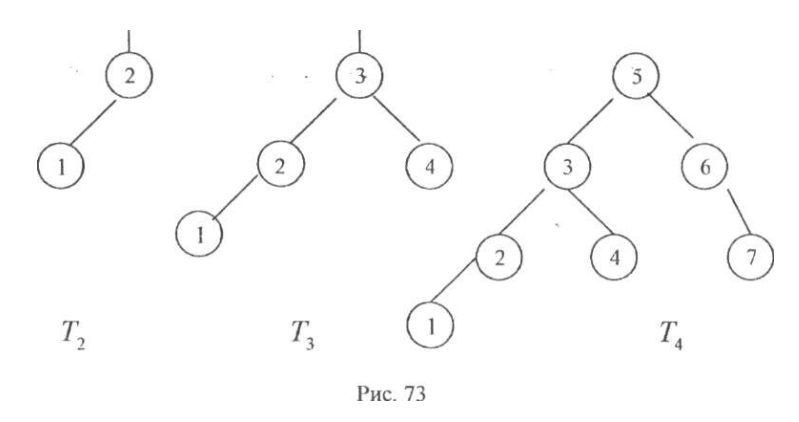


Оптимальність досягається, якщо дерево ідеально збалансоване, при n=2k - l. У цьому випадку оцінка найкраща.

Виникає запитання про те, яка структура найгіршого АВЛ-збалансованого дерева?

Щоб знайти максимальну висоту h всіх збалансованих дерев з п вузлами, візьмемо фіксоване h і спробуємо побудувати збалансоване дерево з мінімальною кількістю вузлів.

Позначимо таке дерево з висотою h через Th і всі такі дерева назвемо Т-деревами. Очевидно, що Т0 - порожнє дерево, T1 — дерево з одним вузлом. Щоб побудувати дерево Тh для h>l, ми задамо корінь з двома піддеревами, які також мають мінімальну кількість вузлів. Отже, піддерева також є Т-деревами. Очевидно, що одне піддерево обов'язково має висоту h-1, а другому дозволено мати висоту h-2. На рис. 73 показано дерева Т2 , Т 3, Т 4.



Оскільки принцип їх організації нагадує принцип побудови чисел Фібоначчі, подібні дерева називаються деревами Фібоначчі. їх визначають так:

1. Порожнє дерево є дерево Фібоначчі з висотою 0.
2. Один вузол є дерево Фібоначчі з висотою 1.
3. Якщо Th-1 та Th-2 - дерева Фібоначчі, то Th = < Th-1, х, Th.2> є дерево Фібоначчі з висотою h.
4. Жодні інші дерева не є деревами Фібоначчі.

Кількість вузлів в Th визначається простим рекурентним співвідношенням:

N0 = 0, N1 =1

Nh = Nh\*1 + Nh\*2

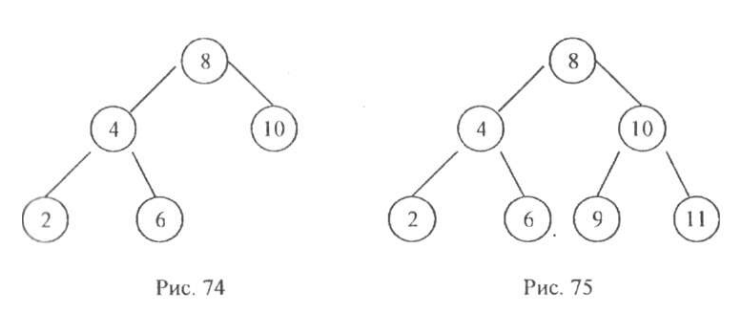
Отже, найгіршими з ABJ1 дерев є дерева Фібоначчі.

**Додавання у збалансоване дерево**

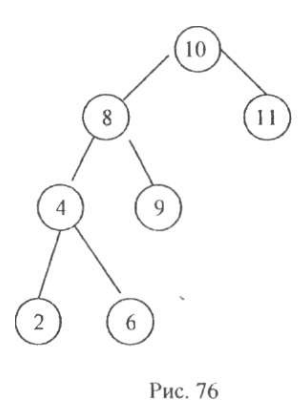
Нехай дано корінь г з лівим і правим піддеревами L та R. Припустимо, що в L включається новий вузол, викликаючи збільшення його висоти на 1. Можливі три випадки:

1. hL = hR : L та R стають нерівної висоти, але критерій збалансованості не порушується.
2. hL < hR : L та R отримують рівну висоту, тобто баланс покращується.
3. hL > hR : критерій збалансованості порушується, і дерево треба перебудувати.

Для ілюстрації операції додавання у збалансоване дерево розглянемо дерево на рис. 74. У корені знаходиться вузол зі значенням 8. Вузли з ключами 9 та 11 у таке дерево можна додати без балансування і при цьому баланс покращується (див. рис. 75).



З такими ж значеннями ключів може бути і витягнуте дерево (див. рис. 76), але воно не відповідає означенню збалансованих.



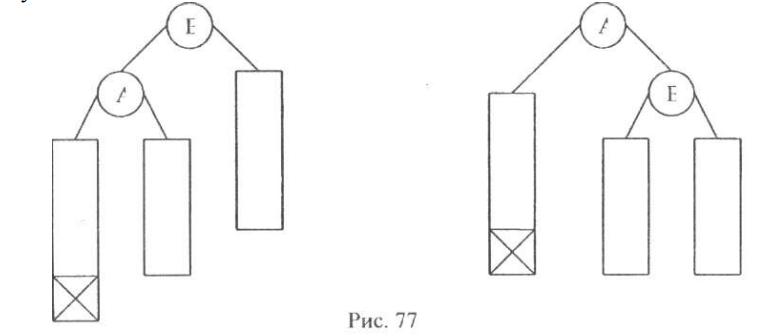
Розглянемо включення у дерево, зображене на рис. 74, вузлів зі значеннями 1, 3, 5 або 7. Кожне таке включення вимагає балансування.

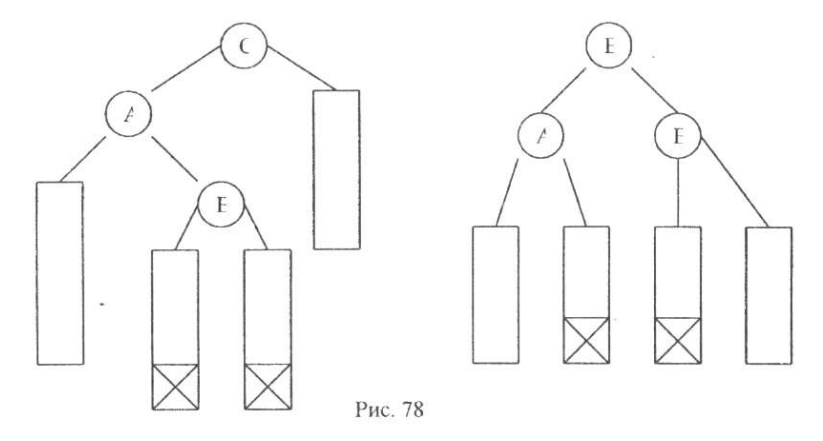
Існує лише дві суттєво різні можливості, які вимагають індивідуального підходу. Інші можуть бути отримані симетричними перетвореннями цих двох.

Випадок 1 виникає при включенні вузлів зі значенням 1 або 3. Вони будуть поєднуватись до вузла зі значенням 2 і при цьому буде порушено баланс. Такий випадок вимагає перебалансування обертанням, у якому задіяно два вузли (однократне обертання). На рис. 77 показано як виглядає дерево до балансування і після перебалансування.

Випадок 2 виникає при включенні вузлів зі значенням 5 або 7. Вони будуть під'єднуватись до вузла зі значенням 6 і при цьому також буде порушено баланс. При перебалансуванні буде задіяно три вузли (двократне обертання). На рис. 78 показано як виглядає дерево до балансування і після такого перебалансування.

В обох випадках перекреслені квадрати позначають під'єднаний вузол.





Прості перетворення цих двох структур відновлюють потрібний баланс. Відзначимо, що допускаються переміщення лише у вертикальному напрямку, в той час як відносне горизонтальне розташування показаних вузлів і піддерев повинне залишатись без змін.

Алгоритм включення і балансування повністю визначається способом зберігання інформації про збалансованість дерева. Одне з вирішень цієї проблеми полягає у зберіганні цієї інформації повністю у самій структурі дерева. Але в цьому випадку включення торкається показника збалансованості вузла, що призводить до надзвичайно високих затрат. Інший варіант - зберігати показник збалансованості разом з інформацією, пов'язаною з кожним вузлом. Показник збалансованості можна інтерпретувати як різницю між висотами правого та лівого піддерева hR – hL. Тоді у кожному вузлі можна тримати, наприклад, інформацію такого виду: bal: -1..1;

**Алгоритм включення:**

1. Переміщатись шляхом пошуку, доки не з'ясують, що ключа, який передбачають включити, немає в дереві.
2. Включити новий вузол і визначити новий показник збалансованості.
3. Пройти назад шляхом пошуку і перевірити показник збалансованості для кожного вузла.

Очікувана висота AVL-збалансованого дерева рівна h = log(n) + с, де с - мала константа (с = 0,25). Це означає, що на практиці AVL-збалансовані дерева ведуть себе як ідеально збалансовані дерева, хоча з ними простіше працювати.

Балансування в середньому відбувається приблизно один раз на два включення. Причому однократне і двократне обертання рівно можливі.

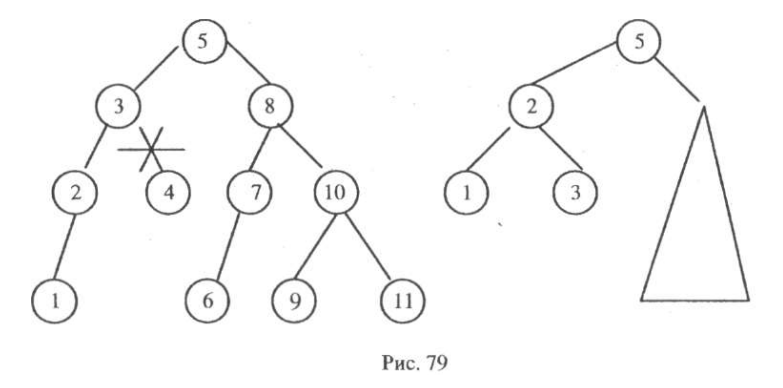
Через складність операцій балансування вважається, що збалансовані дерева слід використовувати лише у тому випадку, коли пошук інформації відбувається частіше, ніж включення.

**Видалення із збалансованого дерева**

Ця операція ще складніша, ніж включення. Операція балансування така ж як і під час додавання елемента. Зокрема, балансування полягає в однократному чи двократному повороті вузла.

Видалення зі збалансованого дерева ґрунтується на алгоритмі видалення у пошуковому дереві.

Простими випадками є видалення термінальних вузлів і вузлів з одним нащадком. Якщо ж вузол, який треба видалити, має два піддерева, знову будемо замінювати його найправішим вузлом лівого піддерева. Але при цьому виникає проблема перебалансування. Видалення з перебалансуванням показано на рис. 79.

****

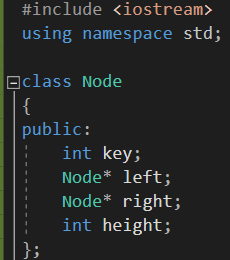
Очевидно, що видалення елемента в AVL-дереві може бути виконано за O(logn) кроків у найгіршому випадку. Проте існує істотна різниця у виконанні процедур включення та видалення. В той час, коли включення одного ключа може викликати щонайбільше одне обертання двох чи трьох вузлів, видалення може вимагати обертання в кожному вузлі вздовж шляху пошуку. Розглянемо, наприклад, видалення найправішого вузла у дереві Фібоначчі. Видалення довільного вузла у дереві Фібоначчі призводить до зменшення його висоти, а видалення найправішого вузла вимагає максимальної кількості поворотів. Отже, так отримують найневдаліше поєднання; найгірший вибір вузла у найгіршому AVL-збалансованому дереві. У результаті імпіричних перевірок отримано, що в той час як одне обертання викликається приблизно двома включеннями, то під час видалення вузла відбувається одне обертання на цілих п`ять видалень. Тому видалення з AVL-збалансованого дерева таке ж просте, чи таке ж складне, як і включення.

Крім таких дерев, існують різноманітні їх модифікації, з якими можна ознайомитись в [9].

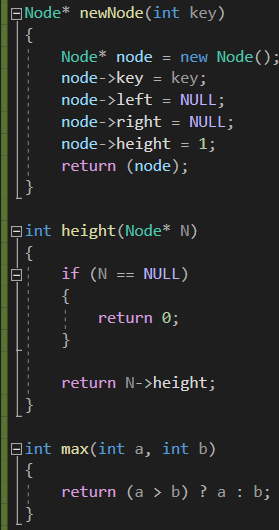
# РОЗДІЛ 1

**Написання головної програми**

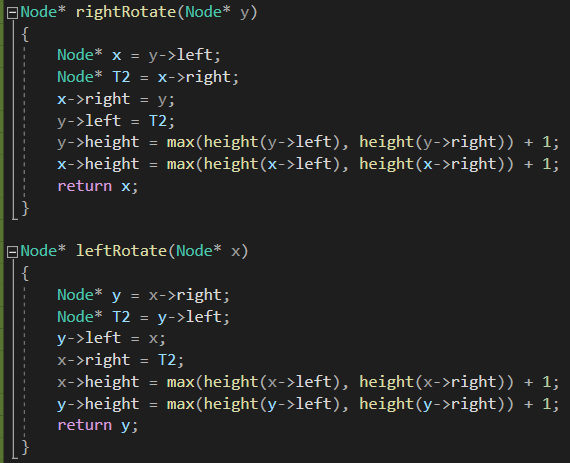
## *1.1. Створення клас “вузол”*



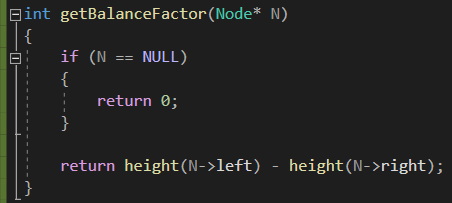
## *1.2. Деякі допоміжні функції*



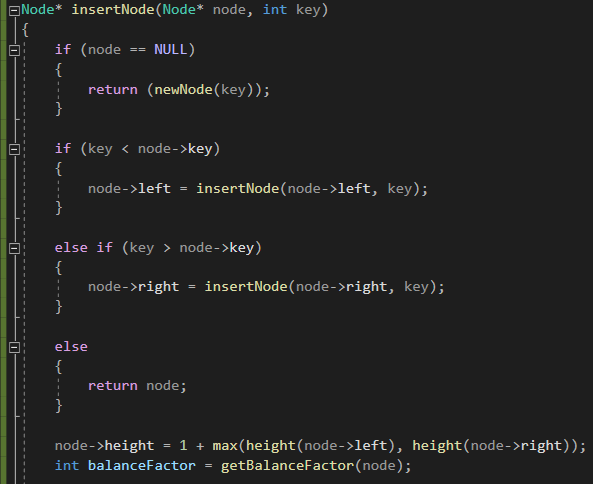
## *1.3. Створення функцій для повороту вправо та вліво*

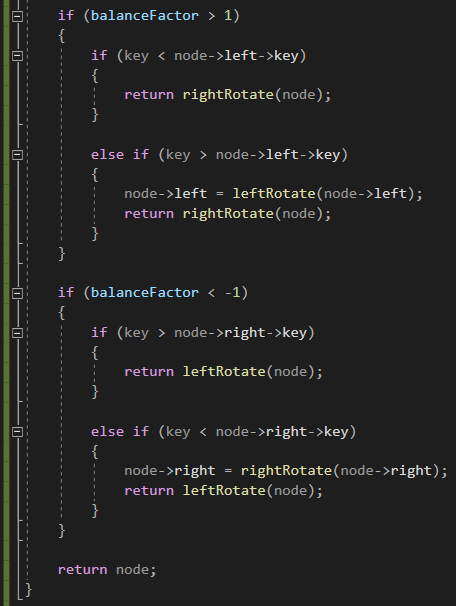


## *1.4. Функція для знаходження балансу*

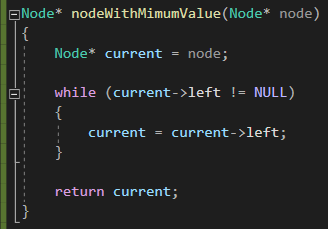


## *1.5. Функція для додавання нових вузлів*

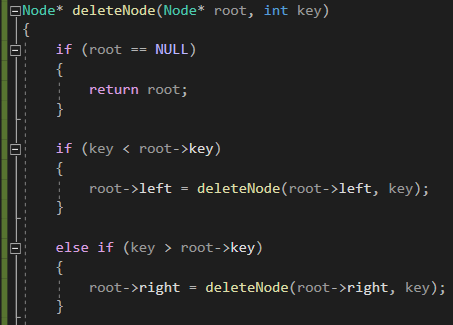




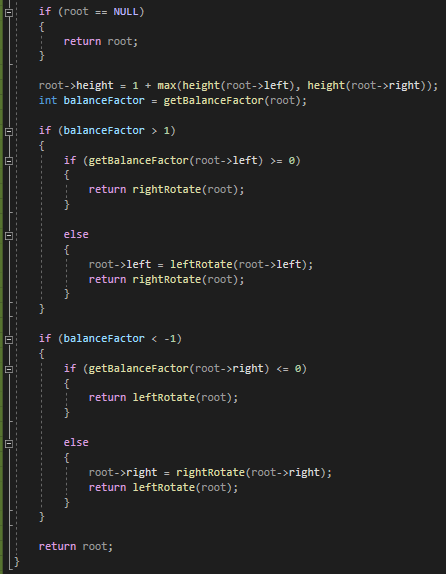
## *1.6. Функція яка повертає вузол з найменшим значенням*



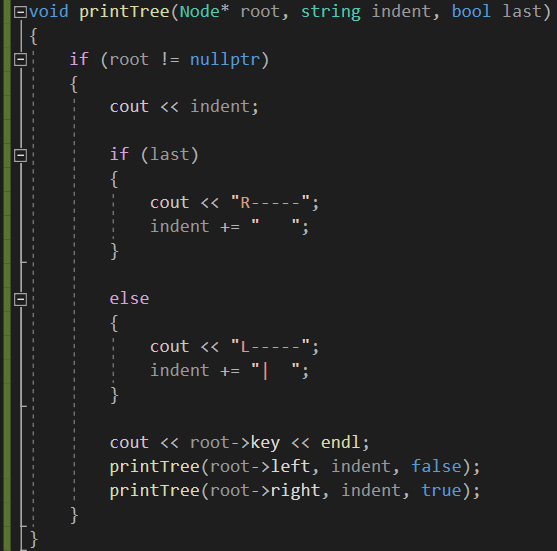
## *1.7. Функція для вилучення вузлів*







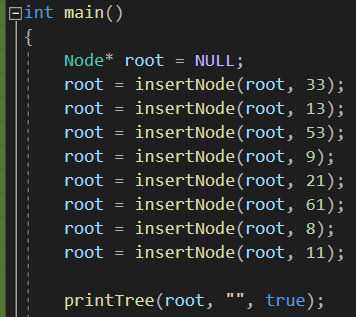
## *1.8. Функція для друку дерева*



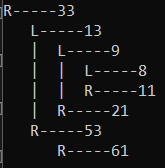
# РОЗДІЛ 2

**Перевірка справності написаної програми**

## *2.1. Створення дерева*



## *2.2. Результат*



# ВИСНОВКИ

У результаті виконання роботи:

1. Розроблено програму для роботи з AVL деревом.
2. Перевірено правильність виконання написаної програми.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. О. Костів. Структури даних: Навч. посібн. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2005. – 146 с