Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Лабораторна робота №3

“Реалізація В+ дерева”

Виконав студент 2-го курсу

Групи ІПС-21

Дубина Андрій

2024

**Завдання**:

Реалізувати B+ - дерево для комплексних чисел

**Теорія**

B+ дерево — тип [дерева](https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D1%80%D0%B5%D0%B2%D0%BE_(%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)), яке подає відсортовані дані в вигляді, що дозволяє швидке додавання, отримання і видалення записів, кожен з яких ототожнений *ключем*. Це динамічний, багаторівневий індекс, з верхньою та нижньою межами на кількість ключів в кожному сегменті індекса (блоці або вершині). В B+ дереві, на відміну від [B-дерева](https://uk.wikipedia.org/wiki/B-%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE), всі записи зберігаються на рівні листових вузлів дерева; у внутрішніх вузлах зберігаються лише ключі.

Основні характеристики B+ дерева:

Сховище даних в листках: У B+ дереві дані зберігаються тільки в листових вузлах. Внутрішні вузли містять лише ключі, які вказують на листя, де знаходяться відповідні дані.

Порядок дерева: B+ дерево має фіксований порядок, що визначає максимальну кількість ключів, які можуть бути збережені в одному вузлі.

Упорядкованість даних: Дані у кожному листковому вузлі впорядковані за ключами. Це дозволяє швидко виконувати діапазонні запити та виконувати послідовний обхід даних.

Локалітет даних: Сусідні дані часто зберігаються поруч один з одним у листових вузлах, що покращує локальність доступу до даних.

**Алгоритми**

Вставка

Перш ніж вставляти елемент у дерево B+, слід мати на увазі ці властивості.  
1. Корінь має принаймні двох дітей.  
2. Кожен вузол, окрім кореневого, може мати максимум m дочірніх елементів і принаймні m/2 дочірніх вузлів.  
3.Кожен вузол може містити максимум m - 1 ключів і мінімум ⌈m/2⌉ - 1 ключів.

Щоб вставити елемент, необхідно виконати наступні дії:  
Оскільки кожен елемент вставляється у листовий вузол, перейдіть до відповідного листового вузла.  
Вставте ключ у листовий вузол.

Випадок І

Якщо аркуш не заповнений, вставте ключ у листовий вузол у порядку зростання.

Випадок II

Якщо аркуш заповнений, вставте ключ у листовий вузол у порядку зростання та збалансуйте дерево наступним чином.Розірвіть вузол на м/2 позиції.Також додайте m/2-й ключ до батьківського вузла.Якщо батьківський вузол уже заповнений, виконайте кроки 2–3.

Приклад:

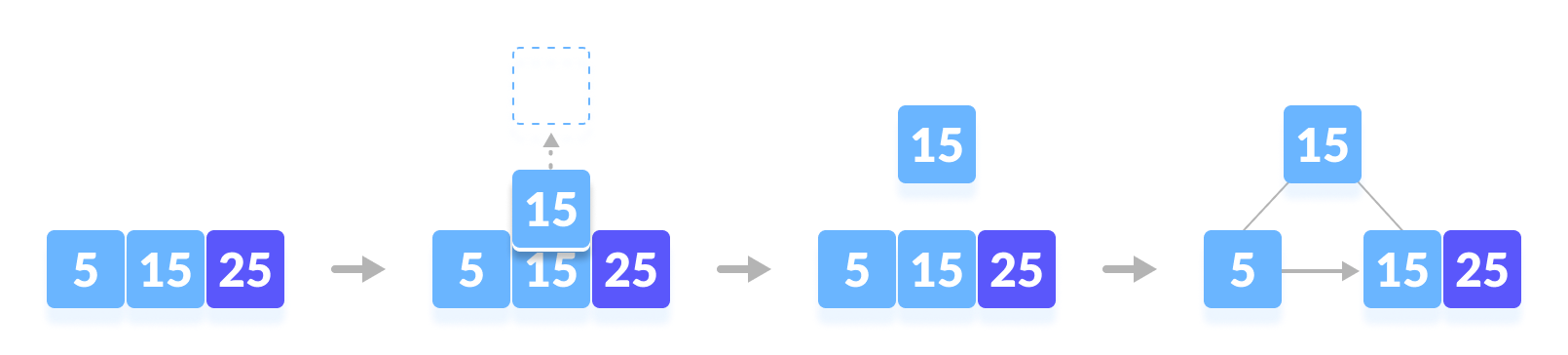
Вставляємо 5.



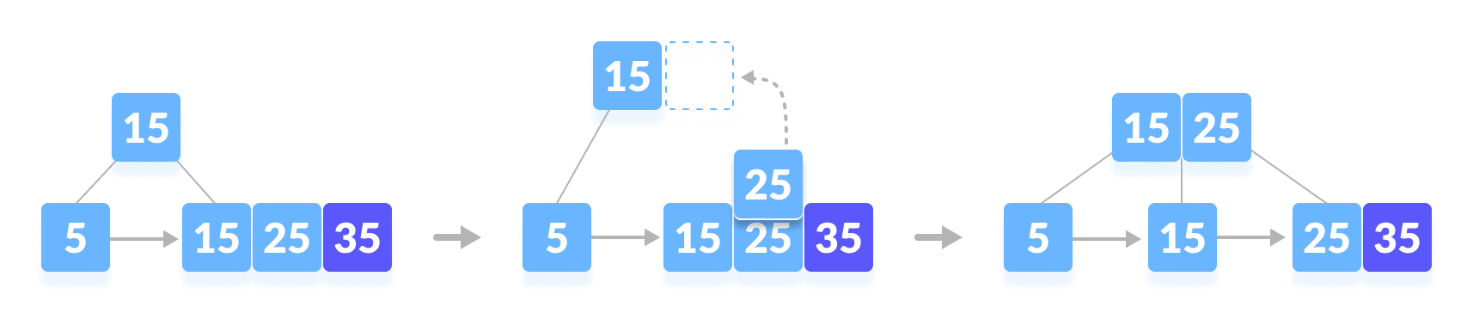
Вставляємо 15



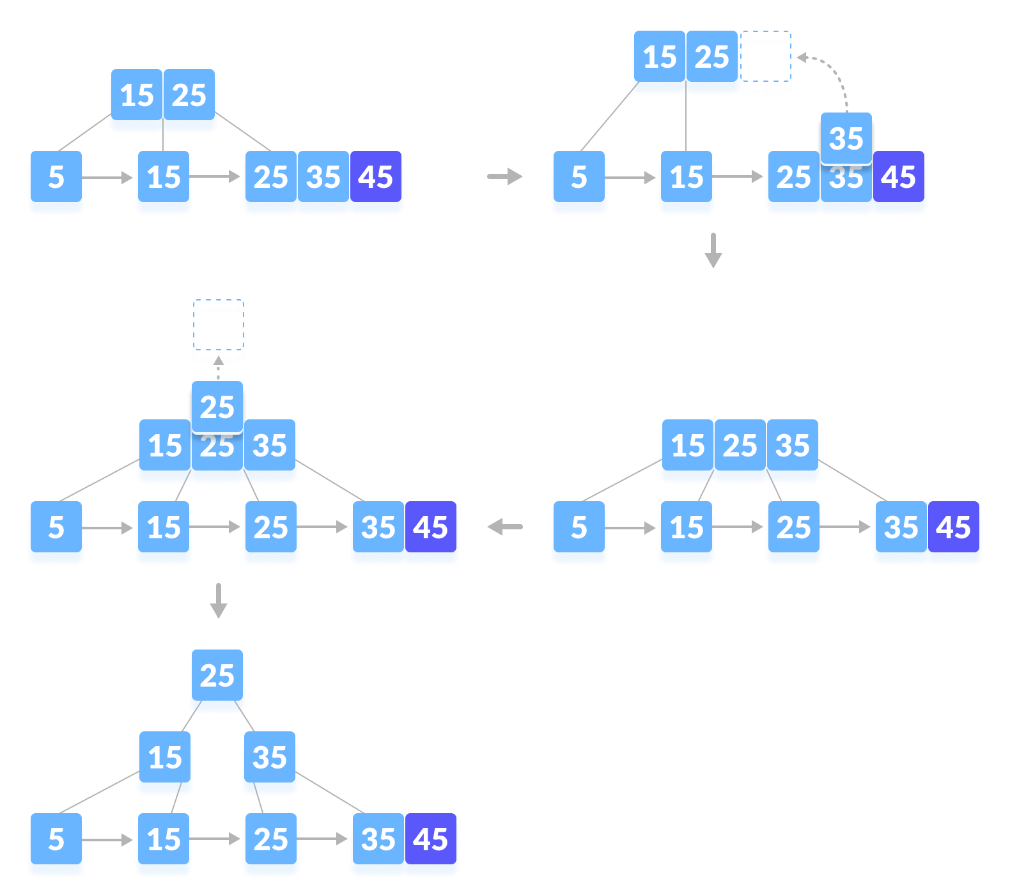
Вставляємо 25



Вставляємо 35



Вставляємо 45



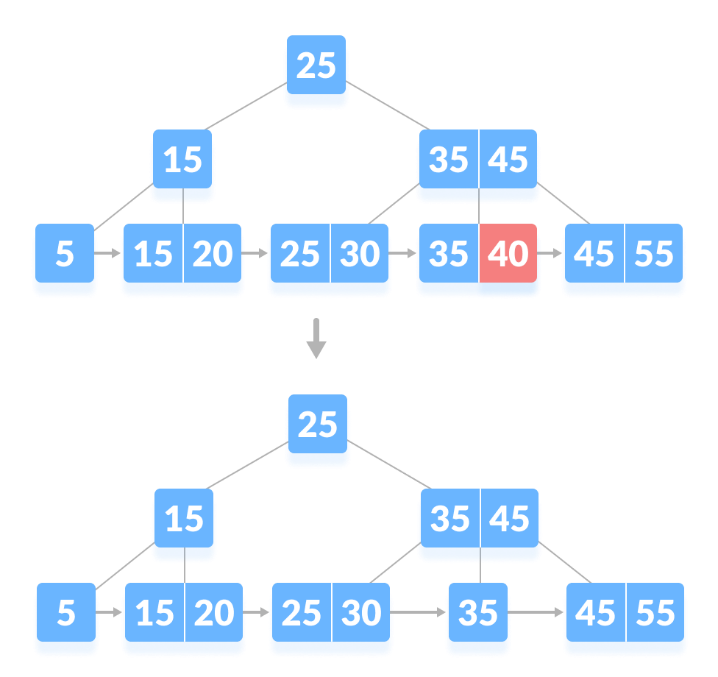
Видалення

При виконанні видалення треба пам’ятати всі ті ж властивості, що були описані і при вставці.

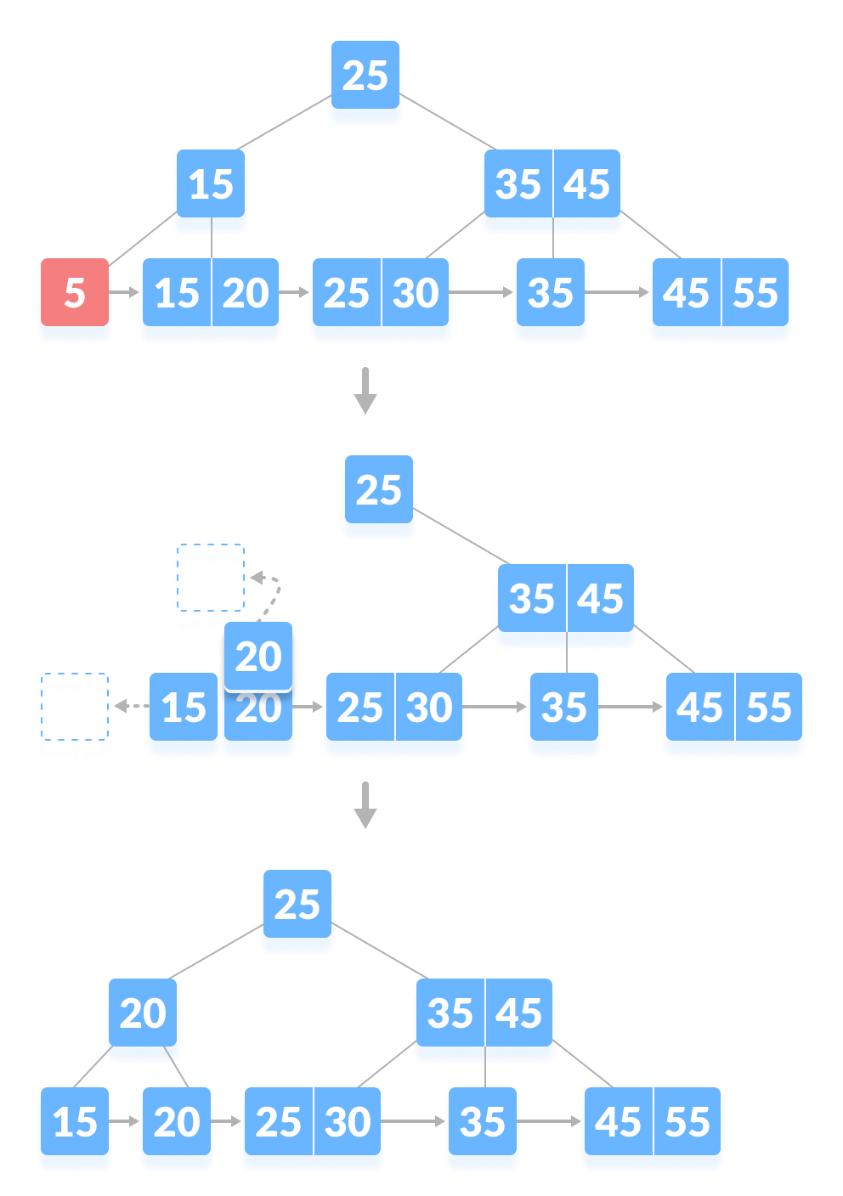
Випадок І

Ключ, який потрібно видалити, присутній лише на кінцевому вузлі, а не в індексах (або внутрішніх вузлах). Для цього є два випадки:

1. Кількість ключів у вузлі перевищує мінімальну. Просто видаліть ключ.



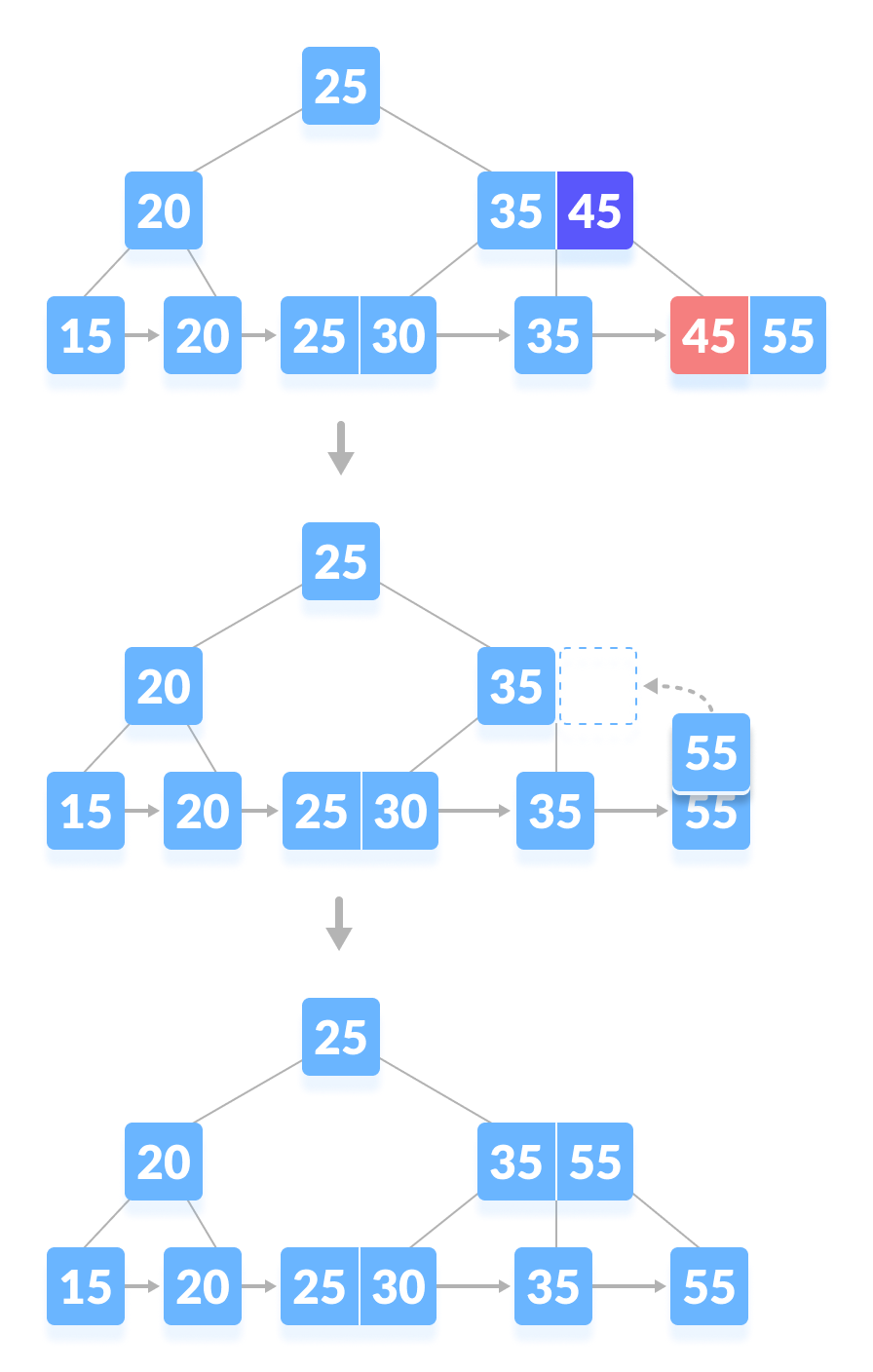
1. Існує точна мінімальна кількість ключів у вузлі. Видалити ключ і позичити ключ у найближчого брата. Додайте середній ключ братнього вузла до батьківського.



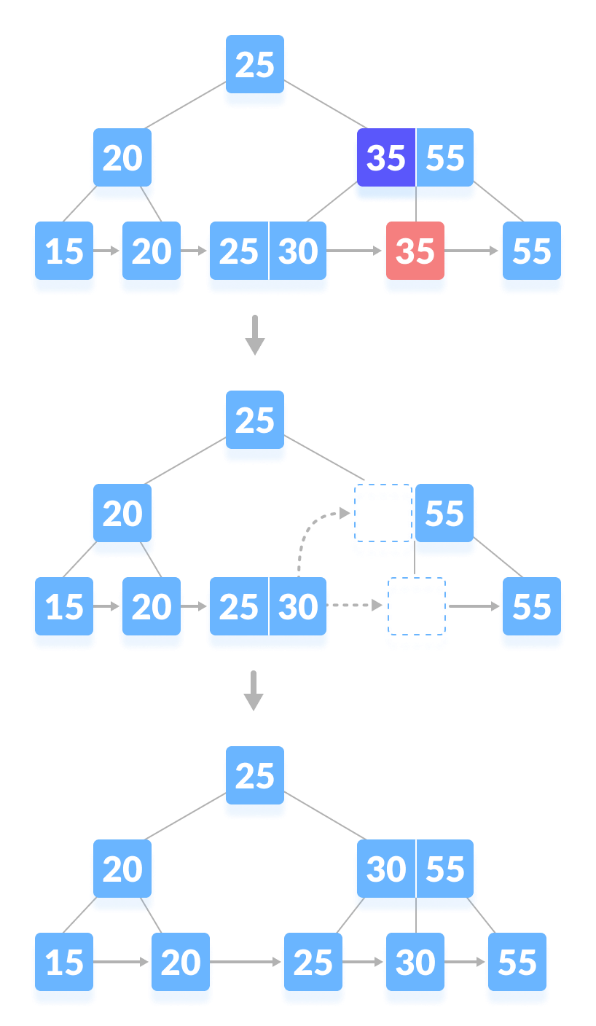
Випадок II

Ключ, який потрібно видалити, також присутній у внутрішніх вузлах. Потім ми також повинні видалити їх із внутрішніх вузлів. Для цієї ситуації існують наступні випадки.

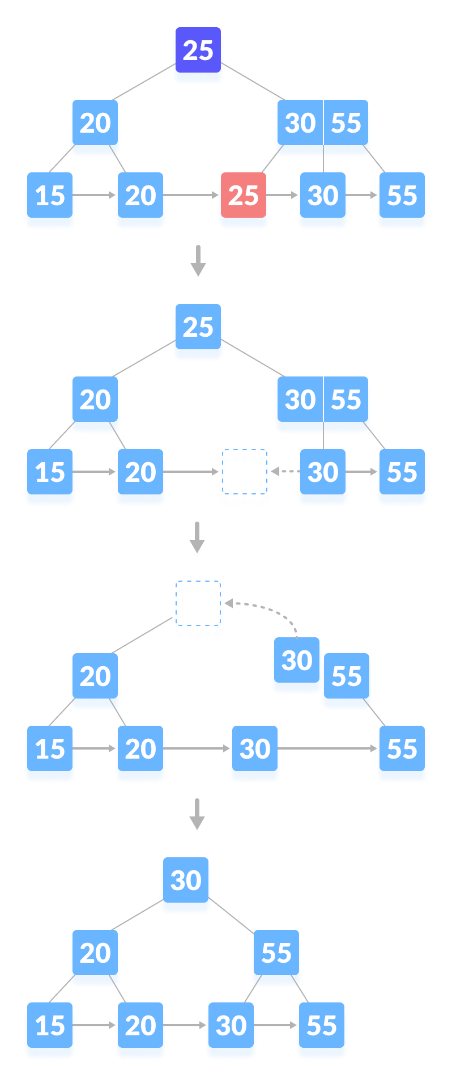
1. Якщо кількість ключів у вузлі перевищує мінімальну, просто видаліть ключ із кінцевого вузла та також видаліть ключ із внутрішнього вузла.Заповніть порожній простір у внутрішньому вузлі наступником у порядку.



1. Якщо у вузлі є точна мінімальна кількість ключів, видаліть ключ і запозичте ключ у його найближчого брата (через батьківський вузол).Заповніть порожній простір, створений в індексі (внутрішньому вузлі), запозиченим ключем.

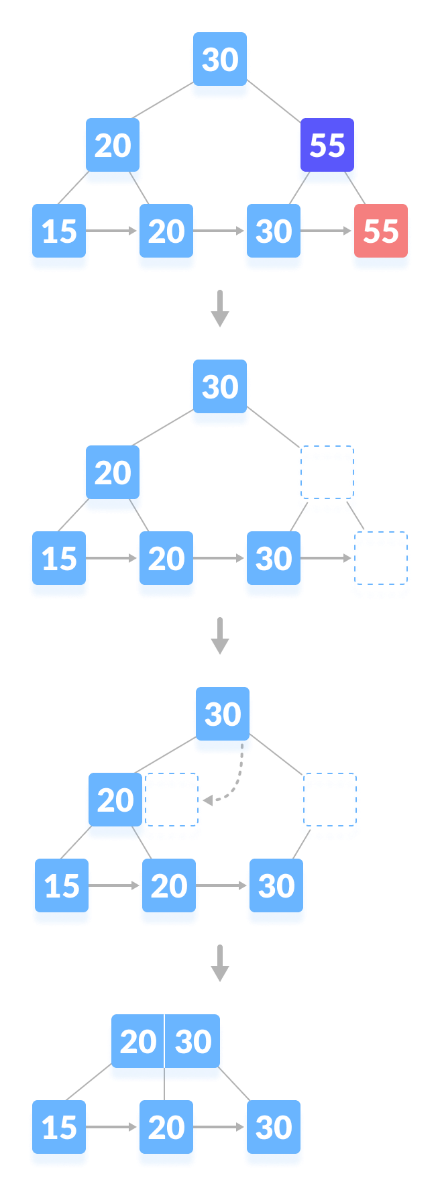


1. Цей випадок схожий на випадок II(1), але тут порожній простір створюється над безпосереднім батьківським вузлом.Після видалення ключа об’єднайте порожнє місце з його братом.Заповніть порожній простір у дідовому вузлі наступником у порядку.



Випадок III

У цьому випадку висота дерева зменшується. Це трохи складніше. Видалення 55 з дерева нижче призводить до цієї умови. Це можна зрозуміти на ілюстраціях нижче.



**Складність**

Операції вставки та видалення в B+ дереві мають часову складність O(log n), де n - кількість записів у дереві.

**Мова програмування**

C++20

**Модулі програми**

int main() { //основна функція

struct Complex { //опис комплексного числа та операцій над ним

class BPlusTree { //клас для В+ дерева та операцій на ньому

struct Node { //структура вузла

void insert(const Complex& value) { //вставка вузла

void remove(const Complex& value) { //видалення вузла

void removeRecursive(Node\* node, const Complex& value) { //рекурсивна фнкція видалення

Node\* getParent(Node\* currentNode, Node\* childNode) { //пошук батьківського вузла

Node\* findLeaf(Node\* node, const Complex& value) { //пошук листка

void insertIntoLeaf(Node\* leaf, const Complex& value) { //вставка в листок

void splitLeaf(Node\* leaf) { //поділ листка

void insertIntoParent(Node\* left, const Complex& key, Node\* right) { //вставка в батьківський вузол

void splitNode(Node\* node) { //поділ вузла

void printTree() { //виведення дерева

**Інтерфейс користувача**

Вхідні дані вводяться програмно (в функції int main()). Результат виводиться в консоль.

**Тестові приклади**

Додаємо вершини з числами: (1, 2), (-3, 4), (5, -6)



Потім видалимо, наприклад, вершину (-3, 4)



Повернемо її і видалимо вершину (1, 2)  


**Висновок**

B+ дерево є вдосконаленою версією B-дерева, яке широко використовується для оптимізації роботи з базами даних і файловими системами. Основні особливості B+ дерева полягають у тому, що всі дані розташовані в листових вузлах, які з'єднані між собою у вигляді зв'язаного списку, а також в тому, що всі ключі знаходяться в листових вузлах і внутрішні вузли містять лише ключі для навігації. Така структура дозволяє ефективно виконувати операції пошуку, вставки і видалення в середньому за час O(log n), де n - кількість ключів у дереві.

**Джерела**

<https://uk.wikipedia.org/wiki/B%2B_дерево>

<https://stackoverflow.com>

https://www.programiz.com/dsa/b-plus-tree