**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

**Кафедра інтелектуальних інформацій систем**

**Алгоритми та складніcть**

**Лабораторна робота №2**

**Завдання – “B+ - дерево”**

**Тип даних: дійсні числа**

**Виконала студентка 2-го курсу**

**Групи ІПЗ-23**

**Третяк Дарина**

2023

**1**

**Предметна область:** Інтернет магазин

**Варіант 11 тип даних дійсні числа**

**Об’єкти:** Категорії,Товари.

**Примітка:** Маємо множину категорій. Кожна категорія містить в собі множину товарів.

**Зміст**

Теоритичні відомості………………………………………………………………………………….3-4

Алгоритм………………………………………………………………………………………………………..4

Складність………………………………………………………………………………………………………5

Мова програмування…………………………………………………………………………………….5

Модулі програми………………………………………………………………………………………..5-6

Інтерфейс користувача…………………………………………………………………………………..6

Приклади ……………………………………………………………………………………………………7-8

Література ……………………………………………………………………………………………………..9

**2**

**Теоритичні відомості**

B+ - дерево є підвидом звичайних B – дерев тому дамо спочатку

визначення B- дереву:

• В-дерева – узагальнення бінарних дерев пошуку.

• Висока степінь розгалуження – вузли можуть мати до тисяч потомків.

Дерево Т з коренем root[T] та властивостями :

1. Кожен вузол x містить поля:

– n[x] – поточна кількість ключів вузла x;

– впорядковано збережені ключі, так що key1[x]≤key2 [x]≤... ≤keyn[x]

[x]; – логічне значення leaf[x], істинне, якщо x – лист.

2. Кожен внутрішній вузол x містить (n[x]+1) вказівник c1[x], ...,

cn[x]+1[x] на дочірні вузли.

3. Ключі key i[x] розділяють піддіапазони ключів піддерев: якщо ki –

ключ, що зберігається у піддереві з коренем ci [x], то k1 ≤ key[x] ≤ k1

≤key2[x] ≤ ... ≤ keyn[x][x] ≤ kn[x]+1.

4. Всі листи розташовані на одній глибині h, що дорівнює висоті дерева.

(Тобто В-дерево ідеально збалансоване за висотою.)

5. Мінімальна і максимальна кількість ключів у вузлі регламентовані

фіксованим цілим t≥2 (мінімальна степінь, minimum degree):

• кожен вузол крім кореня містить як мінімум (t–1) ключ, тобто

матиме принаймні t синів; непорожнє дерево має в корені хоча б

один ключ;

• кожен вузол містить не більше (2t–1) ключів, тобто матиме

максимум 2t синів; вузол вважається повним, якщо має рівно (2t–1)

ключ.

**3**

B+ - дерева

• Істинні значення ключів містяться тільки в листках, внутрішні вузли

містять лише ключі-роздільники діапазонів піддерев.

• Листки додатково зв’язані у список. Це дозволяє швидкий доступ до

ключів в порядку зростання.

• Легко реалізується незалежність програми від структури

інформаційної запису.

• Пошук обов'язково закінчується в листі. Видалення ключа завжди з

листа.

• Вимагають більше пам'яті для представлення, порівняно з B-

деревами.

**Алгоритм**

1.Створено об'єкт класу **BPlusTree**.

2.Створено об'єкти класу **Category** для кожної категорії товарів. 3.Встановлено назву, опис та унікальний ідентифікатор (id) для кожної категорії.

4.Створено об'єкти класу **Product** для кожного товару. Встановлено назву, ціну, наявність, опис та унікальний ідентифікатор (id) для кожного товару.

5.Додано товари до відповідних категорій за допомогою методу **addProduct** у класі **Category**.

6.Вставлено ідентифікатори категорій до B+-дерева за допомогою методу **insert** у класі **BPlusTree**.

7.Виведено B+-дерево за допомогою методу **printTree** у класі **BPlusTree**.

8.За потреби видалення категорії, за допомогою методу **remove** у класі **BPlusTree**, передаючи його унікальний ідентифікатор (id) категорії для видалення.

9.Виведення оновленого B+-дерева, після видалення категорії за допомогою методу **printTree** у класі **BPlusTree**.Для виведення списку товарів у певній категорії, використовуємо метод **search** у класі **BPlusTree**, передаючи його унікальний ідентифікатор (id) категорії. Отримуємо вказівник на об'єкт категорії і використоввуємо його метод **printProductList**, щоб вивести список товарів.

**4**

**Складність**

Складність операцій вставки та видалення в В+-дереві становить O(log n), де n - кількість ключів у дереві.

**Мова програмування**

**С++**

**Модулі програми**

Category.h

Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Автоматически созданное описание

Product.h

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Автоматически созданное описание**5**

BPlusTree.h

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

**Інтерфейс користувача**

Вхідні дані вводяться з консолі користувачем і виводяться в консоль.

**6**

**Тестовий приклад**

Ось приклад використання програми для додавання категорій і товарів у інтернет-магазин, видалення категорії та виведення списку товарів:

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

**7**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

В результаті виконання програми на консоль будуть виведені наступні результати:

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Автоматически созданное описание

**8**

**Література:**

* Лекція 4
* <https://habr.com/ru/articles/66926/>

**9**