**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи № 3 з дисципліни

«Проектування алгоритмів»

„ **Проектування структур даних**”

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

ІП-14 Прокопенко Олексій

Київ 2022

Зміст

[**1** **Мета лабораторної роботи 3**](#_Toc114359761)

[2 Завдання 4](#_Toc114359762)

[3 Виконання 7](#_Toc114359763)

[3.1 Псевдокод алгоритмів 7](#_Toc114359764)

[3.2 Часова складність пошуку 7](#_Toc114359765)

[3.3 Програмна реалізація 7](#_Toc114359766)

[3.3.1 Вихідний код 7](#_Toc114359767)

[3.3.2 Приклади роботи 7](#_Toc114359768)

[3.4 Тестування алгоритму 8](#_Toc114359769)

[3.4.1 Часові характеристики оцінювання 8](#_Toc114359770)

[Висновок 9](#_Toc114359771)

[Критерії оцінювання 10](#_Toc114359772)

# Мета лабораторної роботи

Мета роботи – вивчити основні підходи проектування та обробки складних структур даних.

# Завдання

Відповідно до варіанту (таблиця 2.1), записати алгоритми пошуку, додавання, видалення і редагування запису в структурі даних за допомогою псевдокоду (чи іншого способу по вибору).

Записати часову складність пошуку в структурі в асимптотичних оцінках.

Виконати програмну реалізацію невеликої СУБД з графічним (не консольним) інтерфейсом користувача (дані БД мають зберігатися на ПЗП), з функціями пошуку (алгоритм пошуку у вузлі структури згідно варіанту таблиця 2.1, за необхідності), додавання, видалення та редагування записів (запис складається із ключа і даних, ключі унікальні і цілочисельні, даних може бути декілька полів для одного ключа, але достатньо одного рядка фіксованої довжини). Для зберігання даних використовувати структуру даних згідно варіанту (таблиця 2.1).

Заповнити базу випадковими значеннями до 10000 і зафіксувати середнє (із 10-15 пошуків) число порівнянь для знаходження запису по ключу.

Зробити висновок з лабораторної роботи.

Таблиця 2.1 – Варіанти алгоритмів

|  |  |
| --- | --- |
| **№** | **Структура даних** |
| 13 | Файли з не щільним індексом з перебудовою індексної області, однорідний бінарний пошук |

# Виконання

## Псевдокод алгоритмів

Пошук:

**Початок**

**blocks = Зчитати всі блоки**

**ЯКЩО** blocks.Count = 1

**ПОВЕРНУТИ** GetRecordInBlockByKey (key, blocks[0])

block = GetBlockByKey(key)

**ПОВЕРНУТИ** block

**GetRecordInBlockByKey (key, block)**

(binary search)

min = 0

max = block.records.Count — 1

**ПОКИ min <= max**

mid = (min + max) / 2

**ЯКЩО** key **=** block.Records[mid].Key

**ПОВЕРНУТИ**  block.Records[min]

**ІНАКШЕ ЯКЩО** key < block.Records[mid].Key

max = mid — 1

**ІНАКШЕ**

min = mid + 1

**GetBlockByKey (key)**

**ЯКЩО** blocks.Count = 1

**ПОВЕРНУТИ**  Blocks[0])

key -= key % 50

**ЯКЩО** key = 1

**ПОВЕРНУТИ**  Blocks[0])

(binary search)

min = 0

max = Blocks.Count — 1

**ПОКИ min <= max**

mid = (min + max) / 2

**ЯКЩО** key **=** Blocks[mid].FirstIndex

**ПОВЕРНУТИ**  Blocks[mid]

**ІНАКШЕ ЯКЩО** key < Blocks[mid].FirstIndex

max = mid — 1

**ІНАКШЕ**

min = mid + 1

Додавання:

**blocks = Зчитати всі блоки**

lastBlock = blocks.Last

**ЯКЩО** lastBlock.Records.Count == CountRecordsInBlock

Створити новий блок та новий запис в ньому

Записати всі блоки в файл

**ПОВЕРНУТИ** створений запис

lastRecord = lastBlock.Records.Last

Створити новий запис з ключем lastRecord.Key + 1

Додати до останнього блоку цей запис

Записати всі блоки в файл

**ПОВЕРНУТИ** створений запис

Видалення:

**DeleteRecordByKey(key)**

block = GetBlockByKey(key)

record = GetRecordInBlockByKey(key, block)

block.Records.Remove(record)

**ЯКЩО** block.Records.Count = 0

Blocks.Remove(block)

Записати блоки

## Часова складність пошуку

…

## Програмна реалізація

### Вихідний код

https://github.com/olexiypr/algorithm-labs/tree/features/3-laba-3/Laba3/Laba3

### Приклади роботи

На рисунках 3.1 і 3.2 показані приклади роботи програми для додавання і пошуку запису.

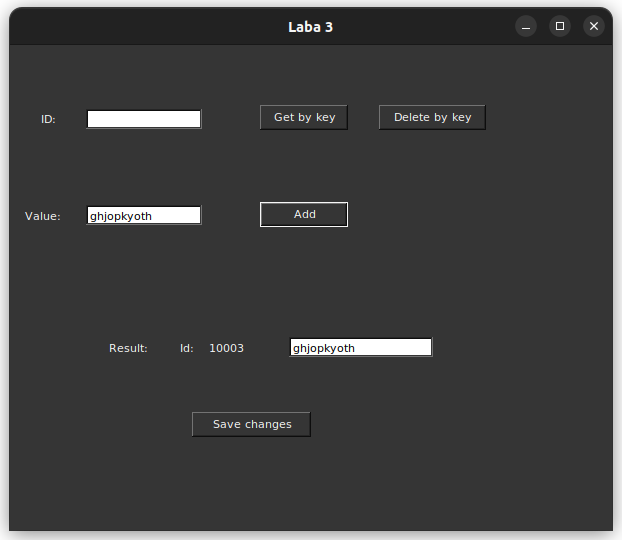


Рисунок 3.1 –Додавання запису

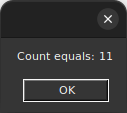
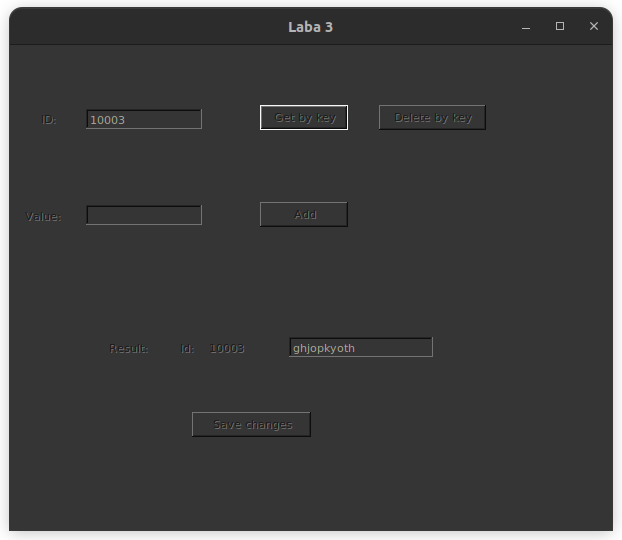


Рисунок 3.2 – Пошук запису

## Тестування алгоритму

### Часові характеристики оцінювання

В таблиці 3.1 наведено кількість порівнянь для 15 спроб пошуку запису по ключу.

Таблиця 3.1 – Число порівнянь при спробі пошуку запису по ключу

|  |  |
| --- | --- |
| Номер спроби пошуку | Число порівнянь |
| 1 | 11 |
| 2 | 13 |
| 3 | 12 |
| 4 | 5 |
| 5 | 6 |
| 6 | 4 |
| 7 | 14 |
| 8 | 9 |
| 9 | 18 |
| 10 | 15 |
| 11 | 13 |
| 12 | 13 |
| 13 | 9 |
| 14 | 11 |
| 15 | 13 |

Висновок

В рамках лабораторної роботи я реалізував файл з нещільним індексом, тобто розбив його на блоки, у кожного з яких було по 50 записів, і початковий індекс, який відповідав ключу першого запису у блоці. Для додавання нового запису перевірявся останній блок на наявність вільної комірки та записувався туди, або створювався новий блок з початковим індексом, як і в цього запису. При пошуку ключ спочатку від значення ключа відніми остачу від 50, а потім, за допомогою бінарного пошуку, шукався блок з таким початковим індексом, при знаходженні блоку, за допомогою бінарного пошуку, шукався необхідний запис. При видаленні знаходився блок, и запис, та видалявсю з нього.

Критерії оцінювання

За умови здачі лабораторної роботи до 13.11.2022 включно максимальний бал дорівнює – 5. Після 13.11.2022 максимальний бал дорівнює – 1.

Критерії оцінювання у відсотках від максимального балу:

* псевдокод алгоритму – 15%;
* аналіз часової складності – 5%;
* програмна реалізація алгоритму – 65%;
* тестування алгоритму – 10%;
* висновок – 5%.

+1 додатковий бал можна отримати за реалізацію графічного зображення структури ключів.