Основи програмування – 2. Алгоритми та структури даних

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 № 7 з дисципліни

«Алгоритми та структури даних-2.

Структури даних»

«Проектування та аналіз алгоритмів пошуку»

Варіант 5

Виконав студент ІП-15, Буяло Дмитро Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірив Соколовський Владислав Володимирович

(прізвище, ім'я, по батькові)

Київ 2022

Основи програмування – 2. Алгоритми та структури даних

**Лабораторна робота 7**

**Проектування та аналіз алгоритмів пошуку**

**Мета** – вивчити основні підходи аналізу обчислювальної складності алгоритмів пошуку, оцінити їх ефективність на різних структурах даних.

**Індивідуальне завдання**

**Варіант 5**

**Завдання**

|  |  |
| --- | --- |
| **№** | **Алгоритм пошуку** |
| 5 | Метод Хеш-функції (Хешування FNV 32), вирішення колізій методом ланцюжків |

Провести аналіз часової складності пошуку в гіршому, кращому і середньому випадках та записати часову складність в асимптотичних оцінках.

Використати безіндексну структуру даних розмірності n, що містить пару ключ-значення рядкового типу. Ключ – унікальне рядкове поле до 20 символів, значення – рядкове поле до 200 символів. Виконати пошук значення по заданому ключу. Розмірність хеш-таблиці регулювати відповідно потребам, а початкову її розмірність обрати самостійно.

Провести ряд випробувань алгоритму на структурах різної розмірності (100, 1000, 5000, 10000, 20000 елементів) і побудувати графіки залежності часових характеристик оцінювання від розмірності структури.

Зробити висновок з лабораторної роботи.

Основи програмування – 2. Алгоритми та структури даних

**Виконання**

1. Псевдокод алгоритму

|  |
| --- |
| **Початок** |
| init32 = 0x811c9dc5 |
| prime32 = 0x01000193 |
| mod32=2^32 |
| hash = init32 |
| **Повторити** для і від 0 до arr.length |
| hash ^= arr[i] |
| hash \*= prime32 |
| hash %= mod32 |
| **Все повторити** |
| **Кінець** |

1. Аналіз часової складності

Найкращий випадок: О(1)

Середній випадок: О(1)

Найгірший випадок: О(n)

Рисунок 2.1 – Найгірший випадок

Основи програмування – 2. Алгоритми та структури даних

Рисунок 2.2 – Середній випадок

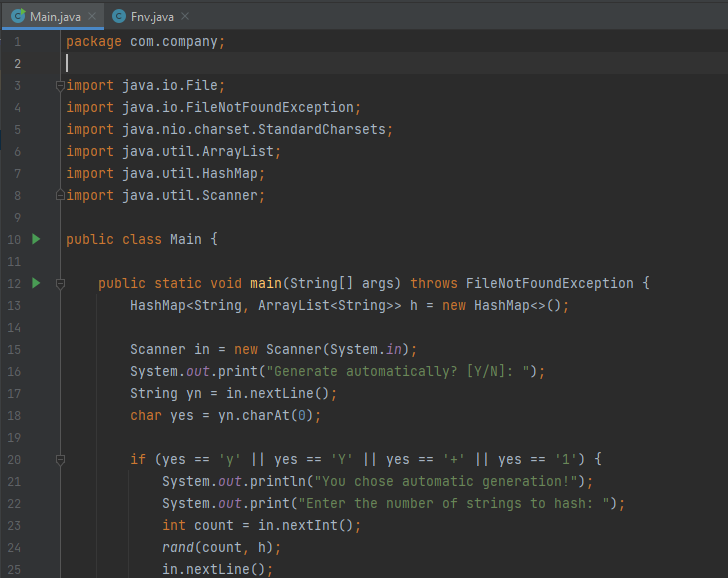
Рисунок 2.3 – Найкращий випадок

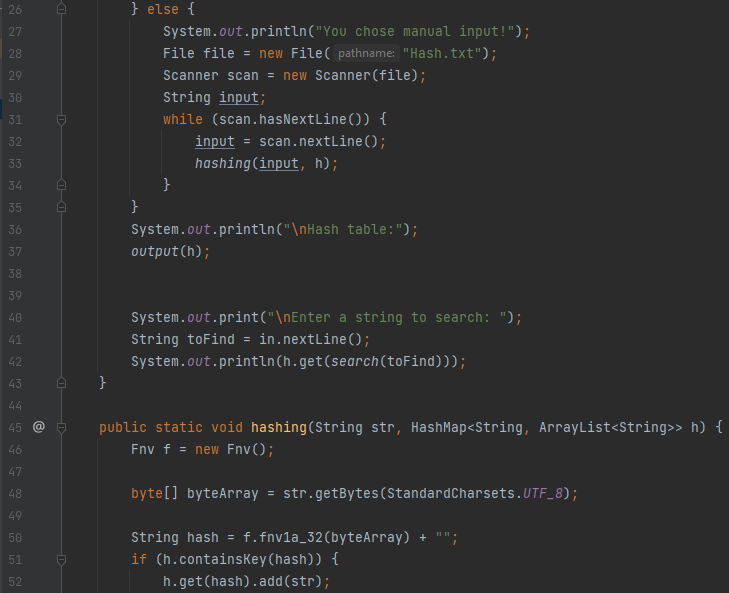
Щоб перевірити асимптотичну оцінку, наведемо рисунки 2.1, 2.2 та 2.3.

Графіки були побудовані за середніми результатами тестувань. Для кожної розмірності хеш-таблиці було зроблено 7 випробувань та обрано середнє.

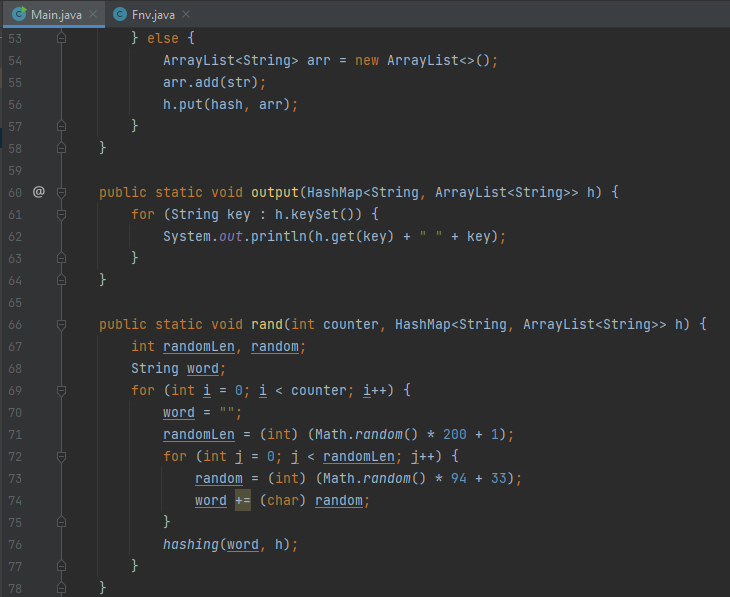
Основи програмування – 2. Алгоритми та структури даних

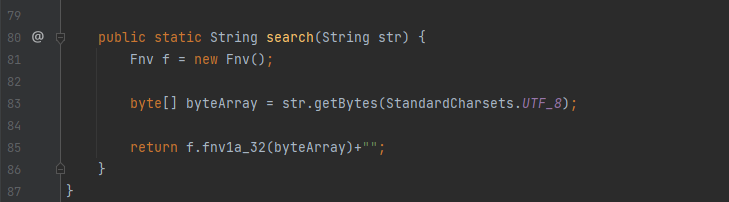
1. Програмна реалізація алгоритму
   1. Вихідний код



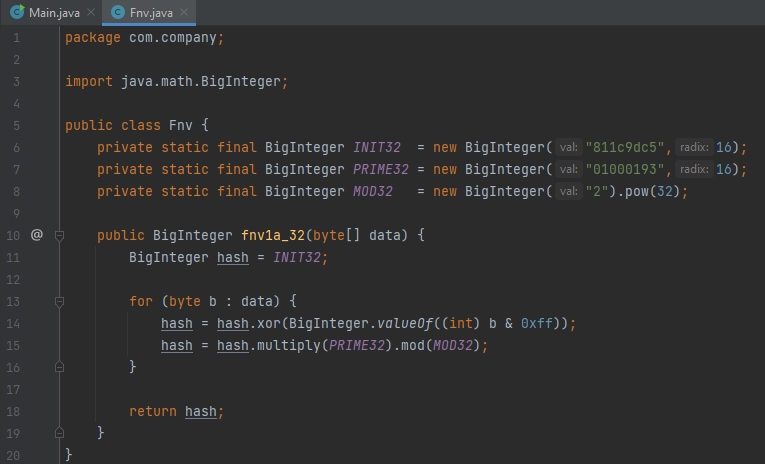


Основи програмування – 2. Алгоритми та структури даних

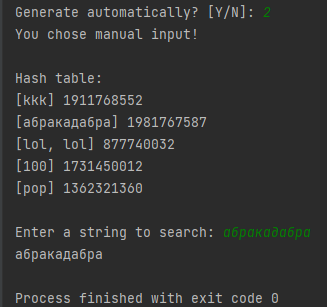
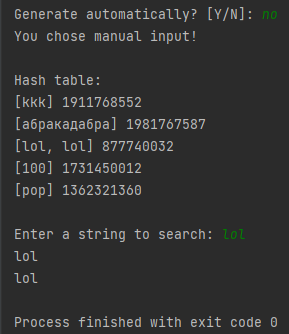




Основи програмування – 2. Алгоритми та структури даних



* 1. Приклад роботи



Основи програмування – 2. Алгоритми та структури даних

1. Тестування алгоритму
   1. Часові характеристики оцінювання

У таблиці 4.1 наведені характеристики оцінювання часу при пошуку елемента за допомогою хеш-функції fnv32 для хеш таблиць різної розмірності.

Таблиця 4.1 – Характеристика оцінювання алгоритму пошуку хеш-функції

|  |  |
| --- | --- |
| Розмірність структури | Середній час виконання, мс |
| 100 | 944 |
| 500 | 1356 |
| 1000 | 1671 |
| 5000 | 2811 |
| 10000 | 4561 |

* 1. Графіки залежності часових характеристик оцінювання від розмірності структури

Рисунок 4.1 – Графік залежності часових характеристик оцінювання

Інші графіки були наведені у пункті 2 (Аналіз часової складності).

Основи програмування – 2. Алгоритми та структури даних

**ВИСНОВОК**

При виконанні сьомої лабораторної роботи, вивчили основні підходи аналізу обчислювальної складності алгоритмів пошуку, оцінили їх ефективність на різних структурах даних.

Розглянули алгоритми пошуку за допомогою модифікованої хеш-функції FNV32. Дослідили їх властивості, провели аналіз часової складності, провели ряд випробувань на різних наборах вхідних даних. За допомогою графіків порівняли часові характеристики оцінювання.

Отже, можемо прийти до висновку, що хеш-функція FNV 32 є ефективною. Проблему колізій вирішували методом ланцюжків, проте в наведеній функції хешування колізії дуже малоімовірні.