Київський національний університет

імені Тараса Шевченка

**Звіт**

**Лабораторна робота №1:**

**З дисципліни «Алгоритми та структури даних»**

**Тема: «Хеш-таблиці»**

Виконав

студент 3 курсу групи МІ-3

факультету комп’ютерних наук та кібернетики

Єрмілов Олександр Михайлович

Київ, 2019 р.

Постановка задачі

Реалізувати алгоритм, що з використанням роздільного зв’язування вставляє N випадкових чисел у таблицю розміром N/100, а потім визначає довжину самого довгого та самого короткого списків.

Теоретична частина

Хеш-таблиця – структура даних, що реалізує інтерфейс асоціативного масиву, а саме, вона дозволяє зберігати пари (ключ, значення) і здійснювати три операції: операцію додавання нової пари, операцію пошуку і операцію видалення за ключем.

Важлива властивість хеш-таблиць полягає в тому, що при деяких розумних припущеннях всі три операції (пошук, вставка, видалення) в середньому виконуються за час О(1). Але при цьому не гарантується, що час виконання окремої операції достатньо малий. Це пов’язано з тим, що при досягненні деякого значення коефіцієнта заповнення необхідно здійснювати перебудову індексу хеш-таблиці.

Існує два основних варіанта хеш-таблиць: з ланцюжками і з відкритою адресацією. Хеш-таблиця містить в собі деякий масив Н елементами якого є пари (хеш-таблиця з відкритою адресацією) або списки пар (хеш-таблиця з ланцюжками).

Виконання операцій в хеш-таблиці починається з обчислення хеш-функцій від ключа. Отримане значення i = hash(key) відіграє роль індексу в масиві Н. Після цього операція перенаправляється об’єктові який знаходиться у комірці масиву Н[і].

Ситуація, коли для різних ключів отримуються одне й те саме хеш-значення називається колізією. Такі події непоодинокі – наприклад, при додаванні в хеш-таблицю розміром 365 комірок усього лише 23 елементів ймовірність колізії перевищує 50% (за умови що ймовірність потрапити в кожну комірку однакова).

В деяких особливих випадках вдається взагалі уникнути колізій. Наприклад, якщо всі ключі елементів відомі заздалегідь (або дуже рідко змінюються), тоді для них можна знайти деяку досконалу хеш-функцію, яка розподілить їх за комірками хеш-таблиці без колізій. Такі хеш-таблиці називають хеш-таблицями з відкритою адресацією.

Алгоритм

Використовуючи роздільне зв’язування для вирішення колізій, при заданій хеш-функції, виконаємо почергово вставку N випадкових елементів та визначимо довжину найдовшої та найкоротшої комірки, що дозволить зробити висновки про якість обраної хеш-функції.

Нехай задано N випадкових чисел та маємо N/100 комірок у хеш-таблиці. Будемо підтримувати зв’язний список для кожної з комірок. Для визначення комірки для кожного числа потрібно підібрати оптимальну хеш-функцію. Оберемо лінійну універсальну хеш-функцію у формі h(k) = (a \* k + b) % p, де р – найменше просте число, більше за N. Параметри a, b оберемо довільно.

Результати роботи програми

Маємо такі результати роботи програми:

|  |  |  |
| --- | --- | --- |
| N | Min | Max |
| 1000 | 94 | 110 |
| 10000 | 73 | 122 |
| 100000 | 70 | 139 |
| 1000000 | 66 | 136 |

Висновок

Як видно з результатів, при збільшенні кількості чисел збільшується максимальна кількість колізій, але зменшується мінімальна кількість колізій. При цьому відхилення досить не значні, що свідчить про вдалий вибір хеш-функції.