**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

**Кафедра інформаційних систем**

**Алгоритми та складність**

**Завдання № 1**

**Звіт**

Петренко Евгеній

Група К-29

**⦁ Умова завдання.**

Ідеальне хешування.

Ідеальна хеш-функція (Perfect Hash Function) — це така хеш-функція, яка перетворює заздалегідь відоме статичне безліч ключів в діапазон цілих чисел [0, m-1] без колізій, тобто один ключ відповідає тільки одному унікальному значенню.

**⦁ Опис алгоритму.**

**Основна ідея**

Ідеальне хешування використовується в задачах із статичним безліччю ключів (тобто після того, як всі ключі збережені в таблиці, їх безліч ніколи не змінюється) для забезпечення гарної асимптотики навіть в гіршому випадку. При цьому ми можемо додатково хотіти, щоб розмір таблиці залежав від кількості ключів лінійно.

В такому хешуванні для доступу до даних потрібно лише обчислення хеш-функцій (однієї або декількох), що робить даний підхід наібистрейшего для доступу до статичних даних. Дана технологія застосовується в різних словниках і базах даних, в алгоритмах із статичною (відомої заздалегідь) інформацією.

Будемо використовувати дворівневу схему хешування з універсальним хешированием на кожному рівні.

**Перший рівень**

Використовується той же принцип, що і в разі хеширования з ланцюжками: n ключів хешіруются в m осередків з використанням хеш-функції h (k) = ((a⋅k + b) modp) modm, випадково обраної з сімейства універсальних хеш-функцій Hp , m, де p - просте число, що перевищує m.

**Другий рівень**

На даному рівні замість створення списку ключів будемо використовувати вторинну хеш-таблицю Sj, що зберігає всі ключі, хешировать функцією h в осередок j, зі своєю функцією hj (k) = ((aj⋅k + bj) modp) modmj, обраної з безлічі Hp , mj. Шляхом точного вибору хеш-функції hj ми можемо гарантувати відсутність колізій на цьому рівні. Для цього потрібно, щоб розмір mj хеш-таблиці Sj дорівнював квадрату числа nj ключів, хешировать функцією h в осередок j.

Незважаючи на квадратичну залежність, нижче буде показано, що при коректному виборі хеш-функції першого рівня кількість необхідної для хеш-таблиці пам'яті буде O (n).

**⦁ Аналіз алгоритму.**

Для першого рівня хеш-таблиці T використовується m = n осередків. Нехай – випадкова величина, що показує кількість ключів, які потрапили в слот i першого рівня таблиці T. Використовуючи слотів для таблиці другого рівня , очікуваний загальний розмір для суми двох рівнів буде дорівнює:

E [] = Θ(n)