

# Comparación de Funciones de Hashing

Max Maldonado  
Universidad de Artes Digitales

Guadalajara, Jalisco

Email: idv16a.msolano@uartesdidgitales.edu.mx

Profesor: Efraín Padilla

Mayo 09, 2019

## I. DEFINICIÓN DEL PROBLEMA

En este ejercicio hemos hecho una comparación de diversas funciones de Hashing, con respecto a la distribución de elementos entre sus slots. Buscamos detectar la función que mejor distribuya los elementos a lo largo del número de slots.

## II. INPUTS Y OUTPUTS

Input	Serie de UUIDs generadas
Output	HashTable que almacena las UUIDs

## III. SOLUCIÓN

### A. *Multiplication KNUTH*

Esta función de hashing consiste en multiplicar la llave por una constante A donde  $0 < A < 1$  y extraer la parte fraccional de la multiplicación entre la Key \* A. Posteriormente multiplicar dicho resultado por el número de buckets. como se presenta en la siguiente imagen:

$$h(K) = \left\lfloor M \left( \left( \frac{A}{w} K \right) \bmod 1 \right) \right\rfloor$$

### B. *Division MODULE*

Esta función de hashing consiste en obtener el índice del slot, al obtener el residuo de la división entre la Key y el número de Slots disponible. Para sacar el mejor provecho a dicha función es necesario que el número de buckets sea primo.

### C. *Universal*

Este metodo consiste en una tabla hash que consta de varios funciones hashing para obtener el índice del slot. La función aplicada en este ejemplo fué la siguiente:

$$h_{a,b}(x) = ((ax + b) \bmod p) \bmod m$$

## IV. CODE

*A. Multiplication*

```

uint32
hash(uint32 _id)
{
    float kA = _id * this->m_a;

    float bucket_id = kA - Math::Floor(kA);

    return Math::Floor(this->m_buckets.size() * bucket_id);
}

```

*B. Division*

```

uint32
hash(uint32 _id)
{
    return _id % this->m_buckets.size();
}

```

*C. Universal*

```

uint32
hash(uint32 _id)
{
    uint32 bucket_id = (this->m_a * _id) + this->m_b;

    bucket_id = bucket_id % this->m_p;
    bucket_id = bucket_id % this->m_buckets.size();

    return bucket_id;
}

```

## V. COMPARACIÓN

