

Tarea 3: Comparación de *Hash0* y *Hash64*

Manuela Acosta Fajardo, Maria Paula Gaviria y Sebastián Martínez Vidal

Facultad de Ciencias Naturales y Matemáticas, Universidad del Rosario

19 de noviembre de 2019

1 Punto 1

La “eficiencia” de una función hash puede observarse, en parte, en la manera en la que la función distribuye elementos de un conjunto de llaves en una tabla hash. Una función hash eficiente distribuye de manera uniforme las llaves del conjunto, de esta forma las operaciones de diccionario: inserción, búsqueda y borrado, toman tiempo constante $\Theta(1)$ “en promedio”.

Para la comparación se tomó una lista con 80.000 palabras y se insertaron en las tablas respectivas de tamaño 101. Se graficaron histogramas de frecuencias que contabilizan cuántas palabras se ubicaron en cada una de las 101 posiciones de la tabla.

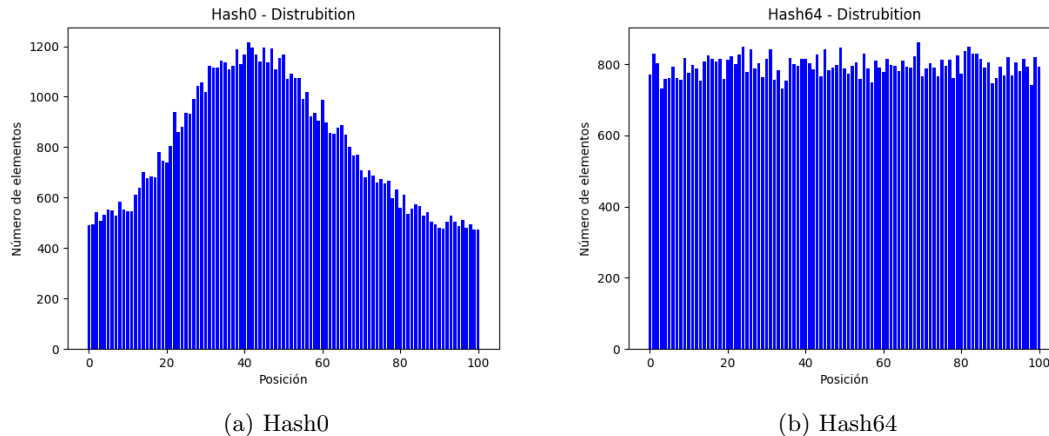


Figure 1: Distribución obtenida

Teniendo en cuenta las gráficas obtenidas, podemos concluir que la función *Hash64* es más eficiente que la función *Hash0*, como se ve en la Fig. 1. Esto se debe a que la primera distribuye de manera más equitativa las palabras en cada uno de los buckets de la tabla *Hash*. Esto permite que, al momento de búsqueda, remoción o añadidura de nuevos elementos, independientemente de la clave que se maneje, el tiempo sea más equitativo en cada caso. Mientras que con la función *Hash0*, en algunas claves, al haber muchos menos elementos, se tomaría menos tiempo que en las claves que tienen más elementos.

2 Punto 2

Además de lo anteriormente mencionado, se comparó el tiempo de acceso promedio en tablas *hash* que usan las funciones presentadas en el punto 1, con un tamaño de la tabla de 1013 y para diferentes valores del factor de carga $\lambda = \text{count} / \text{tablesize}$.

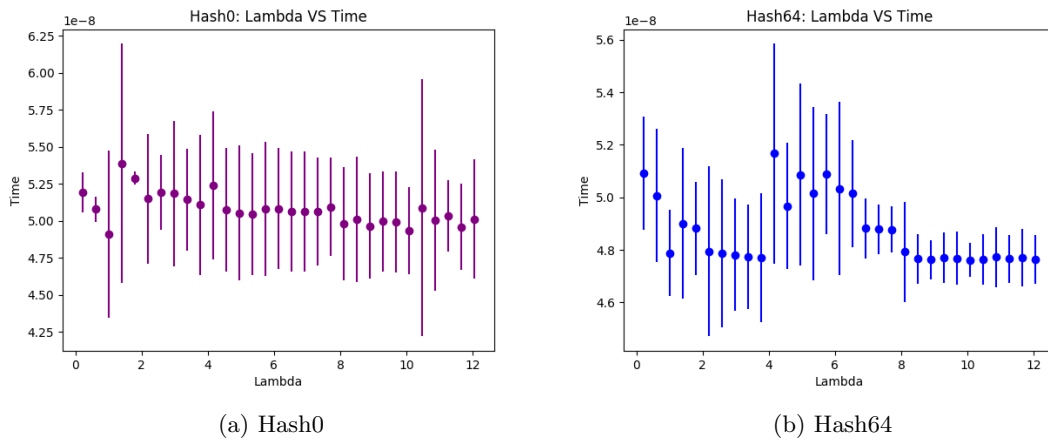


Figure 2: Promedio de tiempos de acceso

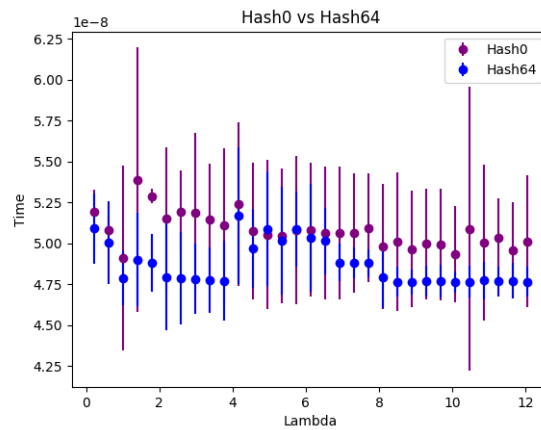


Figure 3: Comparación directa de tiempos de acceso

Teniendo en cuenta las Fig. 2 y 3, podemos observar ciertas diferencias puntuales entre ambas distribuciones. Note que, en general, *Hash64* se demora menos tiempo en acceder a las llaves, independientemente del λ escogido. Además, la desviación estándar de *Hash64* va disminuyendo y estabilizándose para valores de λ a partir de 7 aproximadamente, en contraste con la desviación de *Hash0*, que se mantiene constante en el intervalo 2 a 10, aproximadamente. Aunque en estas gráficas los tiempos son bastante pequeños, por lo que experimentalmente la diferencia no es notoria, para datos mucho más grandes importa más esta diferencia entre las distribuciones de las tablas.