



Universidad
Rafael Landívar

Tradición Jesuita en Guatemala

Tablas Hash

Aplicaciones

Ing. Miguel Matul Calderón

¿Qué son las colisiones?



¿Qué es el factor de carga?



¿Para qué sirven las tablas Hash?



Tablas Hash

key -> hash(key) -> index

0	
1	
2	
3	5
4	
5	
6	
7	23
8	
9	

N=10

Tablas Hash

5 -> hash(key) -> 3

0	95
1	47
2	36
3	5
4	38
5	1
6	4
7	23
8	19
9	50

N=10

Tablas Hash

¿Costo de buscar 95?

0	95
1	47
2	36
3	5
4	38
5	1
6	4
7	23
8	19
9	50

N=10

Tablas Hash

¿Costo de buscar 95?

$O(1)$

0	95
1	47
2	36
3	5
4	38
5	1
6	4
7	23
8	19
9	50

N=10

Tablas Hash

¿Costo de buscar 38?

0	95
1	47
2	36
3	5
4	38
5	1
6	4
7	23
8	19
9	50

N=10

Tablas Hash

¿Costo de buscar 38?

$O(1)$

0	95
1	47
2	36
3	5
4	38
5	1
6	4
7	23
8	19
9	50

N=10

Colisiones

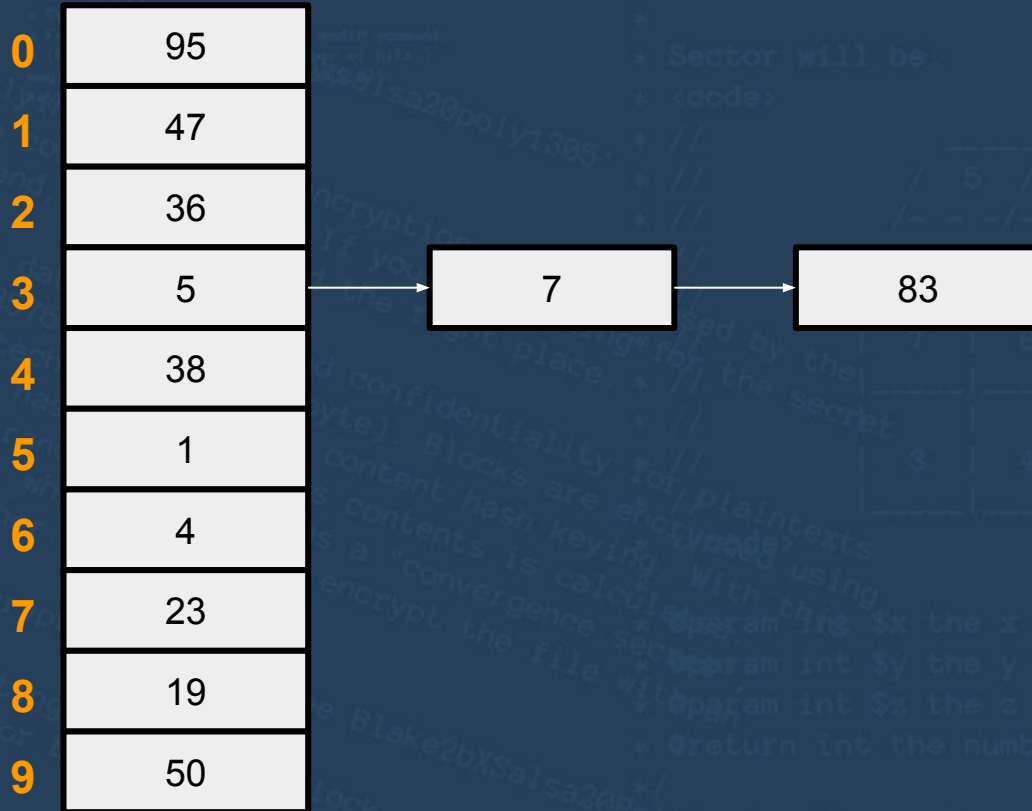
1 -> hash(key) -> 3

0	95
1	47
2	36
3	5
4	38
5	1
6	4
7	23
8	19
9	50

N=10

Listas enlazadas

N=10



Rehashing

Consiste en aumentar el tamaño de la tabla Hash para reducir el factor de carga y así disminuir la probabilidad de colisiones.

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

N=10



0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

N=11

Implementando las listas dentro de la aplicación en java

Socialización de los algoritmos

Aplicaciones de las tablas Hash en la informática

Corrector ortográfico



Corrección con un clic

Corrige cada error con un sólo clic.

ortografío

1

Hallado un posible error de ortografía

ortografía



añade "ortografío" a tu diccionario personal

Diccionarios

```
1  # Create a dictionary
2
3  my_dict = {'Alex': 5,
4             'Ben' : 10,
5             'Carly': 12,
6             'Danielle': 7,
7             'Evan' : 6}
8  my_dict
```

```
{'Alex': 5, 'Ben': 10, 'Carly': 12, 'Danielle': 7, 'Evan': 6}
```

Tablas de símbolos

TABLA DE SÍMBOLOS

Operador	Significado
!	Negación
+	Suma
-	Resto
*	Multiplicación
/	División
%	Módulo
<	Menor
<=	Menor igual
>	Mayor
>=	Mayor igual
!=	Diferente
&&	Conjunción Lógica(Y)
	Disyunción Lógica(O)
==	Igualdad

Shazam



Shazam

Es una aplicación para telefonía móvil que incorpora un servicio que permite la identificación de música. Shazam aprovecha el micrófono que llevan incorporados la mayoría de teléfonos móviles para poder grabar una muestra de música que se esté reproduciendo. Una **huella digital acústica** se crea a partir de la muestra y se compara con una base de datos para encontrar coincidencias. Una vez hecha la relación, el usuario puede recibir información tal como el título de la canción, artista, álbum, enlaces de interés a servicios



Algoritmo de Shazam

Una grabación de al menos cinco segundos dará los mejores resultados. Se puede comenzar a grabar en cualquier punto de la canción y Shazam enviará una coincidencia en cuestión de segundos. Para que este servicio funcione bien, Shazam tiene una base de datos en crecimiento de más de 8 millones de canciones / archivos de audio. Suponiendo que el archivo de audio promedio tiene una duración de tres minutos, ¿se necesitarían más de 45 años para reproducir cada uno de forma consecutiva! Con una base de datos de este tamaño, tienen una gran cobertura, pero ¿cómo encuentra Shazam una coincidencia tan rápidamente en una base de datos tan grande?



Algoritmo de Shazam

En su definición más básica, el sonido son partículas que vibran. Hay tres elementos que hacen que cada sonido sea único:

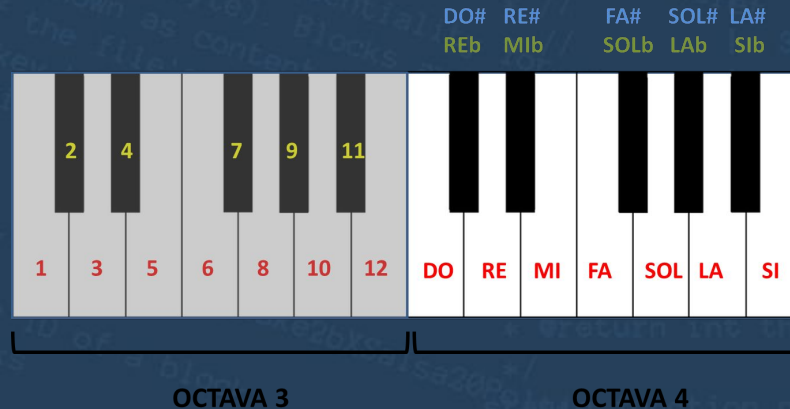
- **Amplitud**
- **Frecuencia**
- **Tiempo**

La amplitud es el tamaño de la vibración, que percibimos como el volumen del sonido. La frecuencia es la velocidad a la que se produce la vibración. La frecuencia de un sonido es lo que percibimos como tono.



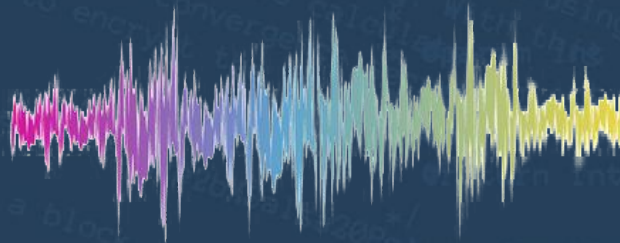
Algoritmo de Shazam

- La **frecuencia** se mide en hercios (Hz), que representa cuántas veces se repite una onda de sonido por segundo.
- El oído humano puede escuchar sonidos que van desde 20 Hz hasta 20 000 Hz.



Algoritmo de Shazam

- Muchos instrumentos pueden tocar las mismas notas, pero una nota en el violín y la misma nota en un piano sonarán diferentes.
- Esta diferencia en la calidad tonal se conoce como **timbre**.
- El **timbre** de un sonido se crea mediante **frecuencias** dentro del sonido que son más altas (se repiten a un ritmo más rápido) que el tono percibido del sonido.
- Estas **frecuencias** se conocen como **armónicos**.



Algoritmo de Shazam

El **tiempo** es importante porque indica en qué momento ocurre un sonido en relación con otros sonidos. Una canción determinada puede estar compuesta por muchos instrumentos que varían en **frecuencia** y **amplitud** a medida que se mueven en el **tiempo** entre sí. Debido a la complejidad de la **amplitud**, la **frecuencia** y el **tiempo**, y la capacidad de medirlos con precisión, dos versiones diferentes de la misma canción seguirán generando una **huella digital** de audio única.



Algoritmo de Shazam

Para hacer una **huella digital** de audio, un archivo de audio se convierte en un espectrograma donde el eje **y** representa la **frecuencia**, el eje **x** representa el **tiempo** y la densidad del sombreado representa la **amplitud** (Figura 1A).

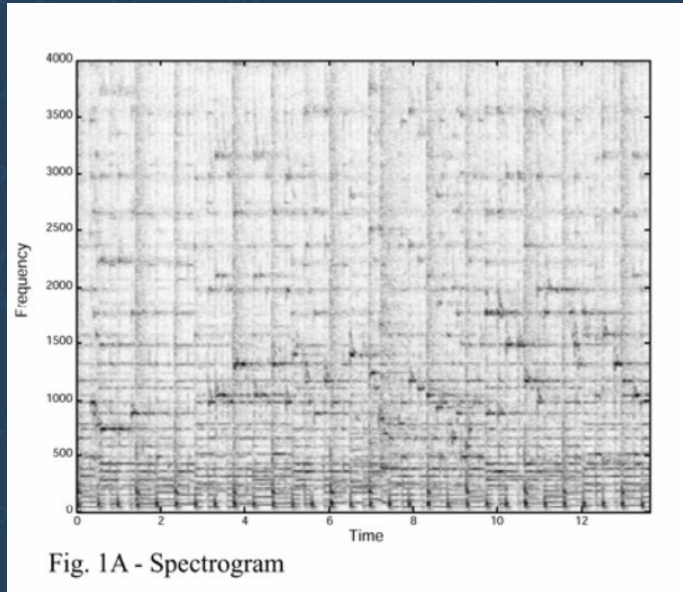
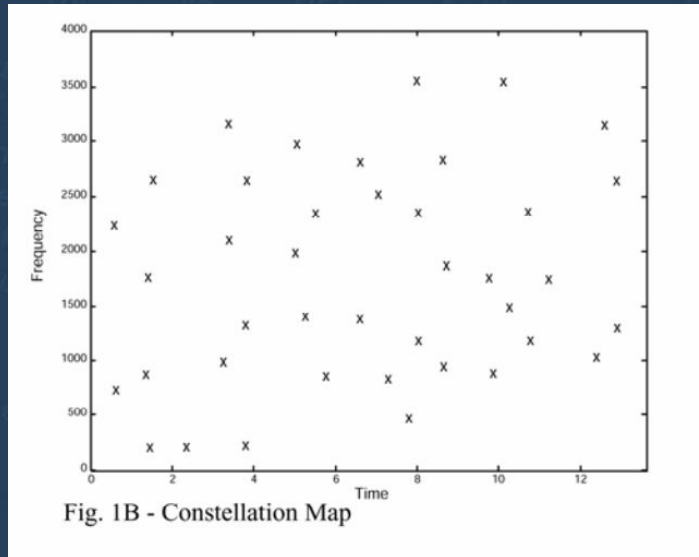


Fig. 1A - Spectrogram

Algoritmo de Shazam

Para cada sección de un archivo de audio, se eligen los picos más fuertes y el espectrograma se reduce a un diagrama de dispersión. En este punto, la amplitud ya no es necesaria (Fig. 1B).



Algoritmo de Shazam

Se tienen todos los datos básicos para hacer coincidir dos archivos que se han sometido al proceso de toma de **huellas digitales**. Sin embargo, solo es posible emparejarlos si un usuario de Shazam comenzó a grabar en el milisegundo exacto en el que comenzó una canción, ¿cuál es la probabilidad de ocurrencia?



Algoritmo de Shazam

Dado que este casi nunca es el caso, existen pasos adicionales para la toma de **huellas digitales** de audio. A través de un proceso llamado **hash combinatorio**, los puntos del gráfico de dispersión se eligen para que sean anclas que están vinculadas a otros puntos del gráfico que ocurren después del punto de anclaje durante una ventana de **tiempo y frecuencia** conocida como zona objetivo (Figura 1C).

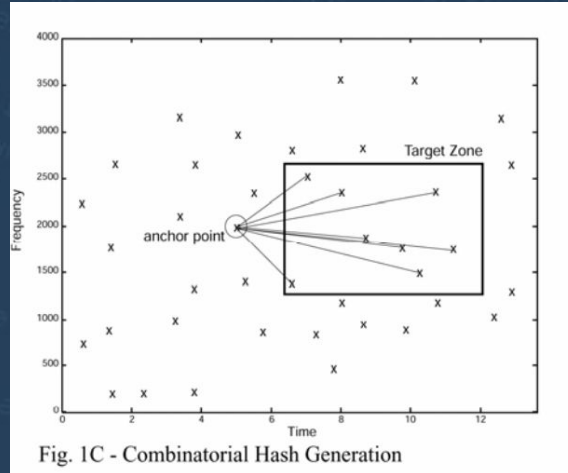


Fig. 1C - Combinatorial Hash Generation

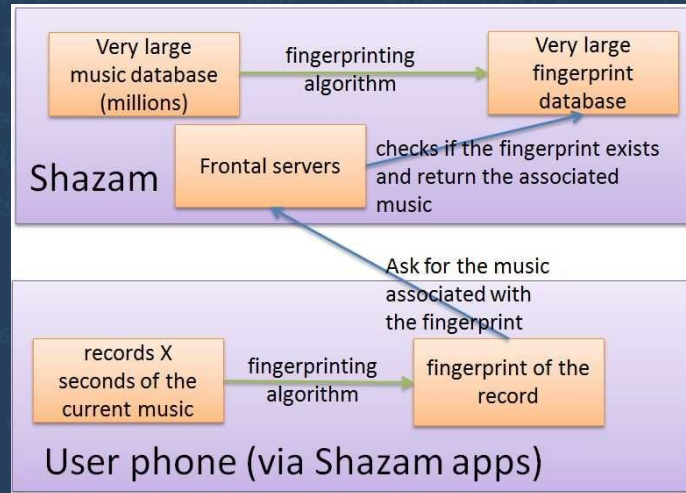
Algoritmo de Shazam

Cada par de puntos de anclaje se almacena en una tabla que contiene la **frecuencia** del ancla, la **frecuencia** del punto y el **tiempo** entre el ancla y el punto conocido como **hash**. Luego, estos datos se **vinculan** a una tabla que contiene el **tiempo** entre el ancla y el comienzo del archivo de audio. Los archivos de la base de datos también tienen ID únicos que se utilizan para recuperar más información sobre el archivo, como el título de la canción y el nombre del artista.



Algoritmo de Shazam

Ahora que se han creado **huellas digitales** para ambos archivos de audio, cada uno de los pares de puntos de anclaje de la grabación del usuario de Shazam se envía a la base de datos de Shazam para buscar pares de puntos de anclaje coincidentes. Esta búsqueda devolverá las **huellas digitales** de audio de todas las canciones que contienen coincidencias **hash**.



Algoritmo de Shazam

Una vez que se tienen todas las coincidencias posibles para la grabación del usuario de Shazam, se necesita encontrar la diferencia de **tiempo** entre el comienzo de la grabación del usuario de Shazam y el comienzo de una de estas posibles coincidencias de la base de datos. Este desfase en el **tiempo** se puede calcular restando el **tiempo** de ocurrencia del par de puntos de anclaje en la grabación del usuario de Shazam de la hora de ocurrencia del **hash** coincidente en el archivo de audio de la base de datos de Shazam. Si una cantidad significativa de **hashes** coincidentes tienen el mismo desplazamiento de tiempo, ¡se determina que esa canción es un match!

Algoritmo de Shazam

Cuando se asigna a un diagrama de dispersión donde el eje **y** representa el momento en el que ocurre el **hash** en la grabación del usuario de Shazam y el eje **x** representa el momento en el que ocurre el **hash** en el archivo de audio de la base de datos de Shazam, se formarán los **hash** coincidentes una línea diagonal (Fig 3A).

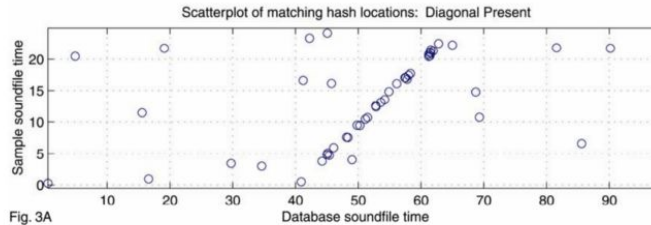


Fig. 3A

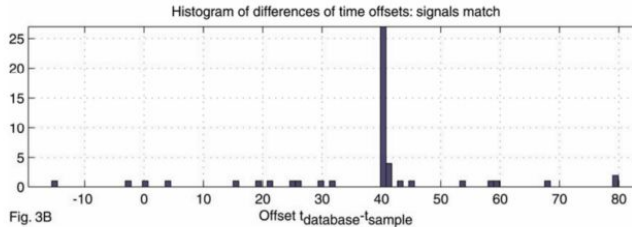


Fig. 3B

Algoritmo de Shazam

En un histograma de los mismos datos donde el eje **y** representa los tiempos de compensación y el eje **x** representa la cantidad de coincidencias, habrá un gran pico en el tiempo de compensación correcto (Fig. 3B).

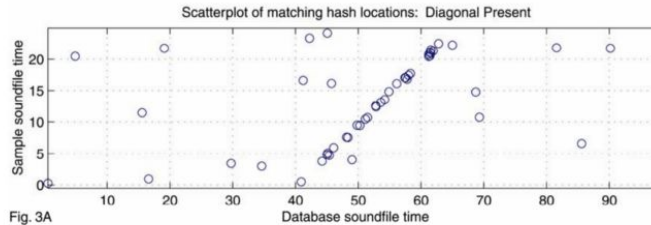


Fig. 3A

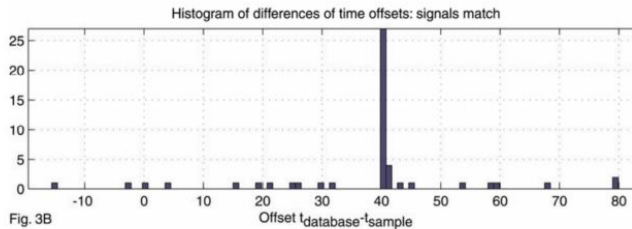


Fig. 3B

Algoritmo de Shazam

Este método de búsqueda de audio es lo suficientemente preciso como para encontrar coincidencias a pesar de que la grabación del usuario de Shazam contenga ruido como personas hablando, ruido de la carretera e incluso otras canciones.



Algoritmo de Shazam





Universidad
Rafael Landívar

Tradición Jesuita en Guatemala

Tablas Hash

Aplicaciones

Ing. Miguel Matul Calderón