Aguirre Ariel *Cañón Cristian* *Algoritmos y Programación II*
100199 *100162* *Correctora: Constanza Gonzalez*

Trabajo Práctico 2

Twitter y una implementación de TT

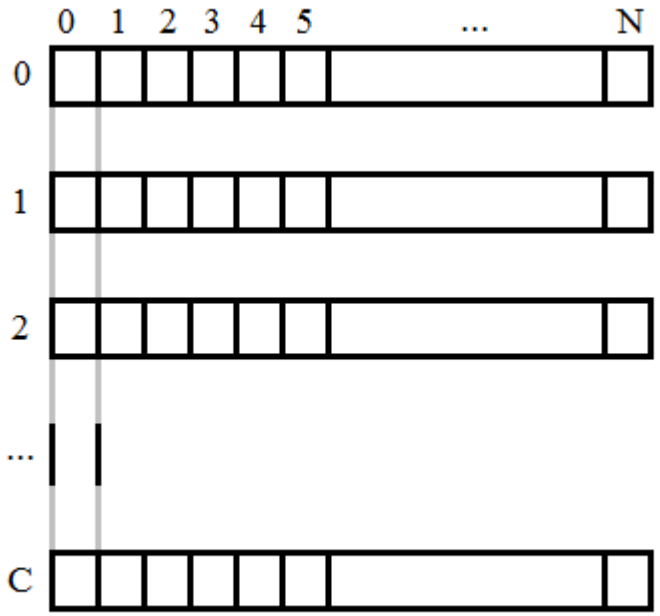
En el presente informe se explicará en detalle el funcionamiento y orden de complejidad de los programas procesar_tweets y procesar_usuarios
Ambos requieren de un archivo, llamado por salida estándar o por parámetro, que contenga un flujo de tweets formateados de manera que cada línea corresponda el autor del tweet seguido de la lista de los hashtags utilizados separados con coma.

Procesar tweets

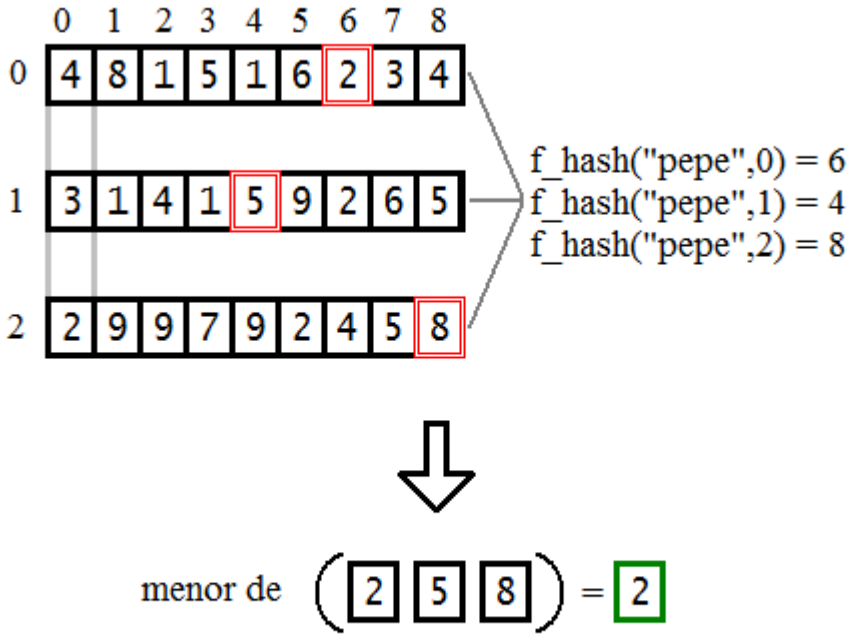
Este programa importará todo el archivo mediante un flujo de N tweets e imprimirá por pantalla una aproximación de los K "trending topics" del mismo.

Lectura

Para el leído de los tweets, usaremos una estructura llamada counting filters, la misma se basa en manejar una cantidad C de arreglos de enteros de tamaño N, cada una con una función de hashing individual, de esta manera se pueden almacenar muchas claves y manejar las colisiones seleccionando desde todos los elementos devueltos el menor de ellos, así se puede almacenar en un tamaño relativamente chico una aproximación de datos potencialmente grandes.



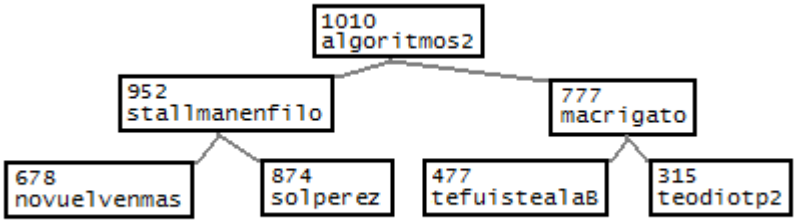
En esta implementación en particular se irán sumando las apariciones de cada hashtag en el flujo de tweets en cada uno de los C arreglos, y devolviendo el menor del que le corresponda a cada clave cada vez que quiera obtenerse su aproximación.



Para guardar las menciones de cada clave dijimos que usaremos un counting filter, pero la lista de hashtags ocurridos las guardaremos en un hash, de manera que tanto el agregado de hashtags como de menciones, para cada ocurrencia, es ***O(1)***, es decir que para todo el archivo será ***O(n)***.

Heapificado

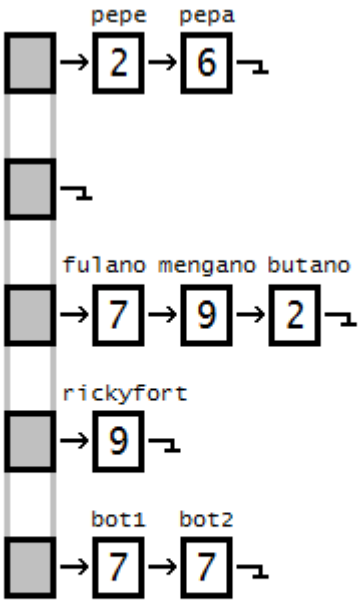
Una vez procesados los datos pasaremos a iterar la lista de hashtags, y asignarlas junto con su correspondiente valor en un struct, luego las encolaremos en un heap de máximos que guarde los K elementos con más menciones.



De esta forma tendremos todo el arreglo de tamaño K heapificado listo para ser devuelto. Ya que recorrimos los arreglos de tamaño N y encolamos y desencolamos posibles K elementos en un heap, el orden de toda esta operación será ***O(n log k)***. El enunciado de la consigna especificaba que se lea todo el flujo cada N iteraciones, de manera que el resultado que obtuvimos no es el final, sino que debemos repetir el proceso comparando y sumando las menciones del counting filters obtenidos hasta que termine el archivo, y de esa manera imprimir el resultado final.

Procesar usuarios

Este programa contará la cantidad de hashtags que utiliza cada usuario, para lo cual iterará todo el archivo guardando en un hash común para cada usuario un contador de hashtags que envió.



Ya que es un hash el costo de acceder y sumar un contador es ***O(1)***, y debido a que tendrá que recorrer todo el archivo, hacer esto en total será ***O(u + t)***, siendo u la cantidad de usuarios y t la cantidad de hashtags.

Luego de haber recorrido todo, se imprimirá de forma ordenada guardando el hash en un heap y desencolándolo elemento a elemento para que se devuelva ordenado, si asumimos que la cantidad ‘t’ de hashtags es potencialmente mayor a la de usuarios, el orden de esta operación será despreciable en la expresión final.

Aguirre Ariel *Cañón Cristian* *Algoritmos y Programación II*
100199 *100162* *Correctora: Constanza Gonzalez*

Pruebas

Se realizaron las siguientes pruebas de tiempo para distintos pesos de archivos y parámetros en cada uno de los programas

Procesar_usuarios

Prueba	Peso archivo	tiempo
1	2,5Kb	0seg
2	20,0Mb	2seg
3	37,3Mb	4seg
4	70,2Mb	8seg
5	140,9Mb	18seg
6	255,8Mb	34seg
7	459,5Mb	67seg

Procesar_tweets

Prueba	n	k	tiempo
1	500000	10	2.268s
2	500000	100	2.288s
3	500000	1000	2.308s
4	1000000	100	4.420s
5	1000000	1000	4.456s
6	1000000	10000	4.468s
7	1000000	100000	4.540s
8	2000000	10	9.036s
9	2000000	1000	9.256s
10	2000000	100000	9.436s

Se puede observar cómo los tiempos obtenidos, tanto en procesar_usuarios como en procesar_tweets confirman lo observado previamente sobre el orden de cada uno.