Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Primer Semestre de 2025

IIC 2440 – Procesamiento de Datos Masivos
Actividad Evaluada 2

Instrucciones. Esta actividad debe ser resuelta en forma individual.
El formato de entrega consta de los siguientes archivos

- Un documento PDF con la respuesta a las preguntas.
- Un archivo donde esté el código responsable de sus simulaciones.

Fechas. La fecha de entrega de la tarea es el 21 de Abril, a las 20:00 hrs.

1. Simulación de Filtros de Bloom

Escribe un programa que pueda simular la probabilidad de falsos positivos que ocurren en un Filtro de Bloom cuando le entregamos un string que no es parte de su set de llaves. Específicamente, tu programa debe recibir como parámetros:

- El tamaño n del vector de bits B del Filtro de Bloom.
- La cantidad k de hashes a usar.
- La cardinalidad m del conjunto M de valores conocidos.
- La cantidad s de simulaciones a hacer.

Tu simulación debe incializar un filtro de bloom, y luego probar con s strings aleatorios que no estén en M, revisando si esos string son marcados como falsos positivos por el algoritmo, a fin de estimar la probabilidad. Puedes usar la librerìa `hashlib` de python para programar tus tablas de hash.

Ahora, ejecuta tu simulación una cantidad $s = 1000$ veces para combinaciones de parámetros $m = 1000$, $n = [m, 2m, 3m, 4m, 5m, 6m, 7m, 8m]$ y $k = [1, 2, 3, 4, 5]$.

Qué valores encontraste que disminuyan la probabilidad de falsos positivos?

2. Filtros de Bloom temporales

Supon ahora que tienes un nuevo conjunto M' con $m' = 4m$ valores.

Tu deber será diseñar un esquema de filtros de Bloom que, con espacio $2n$, sea capaz de lograr una probabilidad de falsos positivos similar a la de la parte 1 para ese n y m dado.

Vas a poder lograr esto debido a la siguiente suposición: **En cada instante de tiempo ves aleatoriamente un 20 % de los valores de M'. Si no ves un valor de M' en cuatro instantes de tiempo, puedes asumir que ese valor de M' ya no existe.**

Entonces:

1. Explica como podemos usar un arreglo de n posiciones B, con dos bits asociados a cada una, para poder simular una accion de borrado en un filtro de bloom que efectivamente elimine una posición de B si ninguno de los elementos de M' cuyos hashes fueron a parar a esa posición en B se ha visto en cuatro intervalos de tiempo.

Usa esta idea para diseñar un esquema de Filtro de Bloom que no tenga falsos negativos, y que use solo espacio $2n$, pero que incluya tu estrategia a fin de liberar algo de espacio cuando no se ven elementos en cuatro intervalos de tiempo.

Importante: Tu esquema no podrá usar más memoria para guardar información de forma persistente salvo la permitida por el filtro de bloom: $2n$. En particular, no puedes asumir que tienes M' posiciones adicionales de memoria sobre las que puedes ir contando que pasa con cada uno de los elementos.

2. Actualiza el código de tu simulación para que pueda simular este nuevo esquema. Ahora tu simulación recibe un parámetro adicional t, y va a funcionar de la siguiente forma:

   Al comienzo, debes correr tu simulación por t tiempos, simulando en cada instante que viste un 20 % de los valores de M' y actualizando el filtro de Bloom para que tome en cuenta que los valores que no se han visto en cuatro intervalos de tiempo seguidos ya no forman parte de M', de acuerdo a tu esquema (la respuesta a la parte 2.1). Luego de eso, recibes s strings aleatorios, a fin de simular la probabilidad de un falso positivo, como antes.

3. Simula de nuevo para $t = 100$ y $m' = 4m$, y las mismas conbinaciones de n y k que antes (recuerda que tu espacio permitido ahora es $2n$, o dos bits por cada entrada entre 0 y $n - 1$). Compara la mejor probabilidad de falso positivo en este caso con tu respuesta de la parte 1.