

ANÁLISIS DE DATOS MASIVOS

BÚSQUEDA DE ELEMENTOS MÁS RECIENTES

Blanca Vázquez

8 de octubre de 2024

- Supongamos que tenemos una ventana de tamaño n en un flujo binario, la cual no podemos almacenar
- La tarea consiste en calcular el número de unos que hay en los últimos k bits para cualquier $k \leq n$ bits
 - La solución exacta requiere n bits
 - Existen algoritmos que pueden estimar este número con un menor número de bits

- Algoritmo para encontrar el número de 1s en una ventana binaria
- La pregunta a resolver es: ¿cuántos 1s hay en los últimos k bits? donde $k < N$
- Se basa en la construcción de cubetas*

...101011000101110110010110...

N = 24 (tamaño de la ventana)


- Para iniciar, cada bit en la ventana tiene una marca de tiempo (la posición en la que llega).
- Siempre empezamos del lado derecho y debe empezar con 1.
- Cada cubeta debe tener al menos un 1.
- Todas las cubetas deben estar en potencias de 2
- Las cubetas no pueden disminuir de tamaño, a medida que nos movemos en el tiempo

EJEMPLO: ALGORITMO DGIM

Flujo de datos:

...101011000101110110010110...

Paso 1: Crear cubetas

...101011000101110110010110...


Siempre empezamos del lado derecho y en 1.

...**101011** 000 **10111** 0 **11** 00 **101 1** 0...
 2^2 2^2 2^1 2^1 2^0

Todas las cubetas están en potencia de 2.

EJEMPLO: ALGORITMO DGIM

...101011000101110110010110...

Siempre empezamos del lado derecho y en 1.

Paso 1: Crear cubetas

...**101011** 000 **10111** 0 **11** 00 **101 1** 0... Todas las cubetas están en potencia de 2.
 2^2 2^2 2^1 2^1 2^0

87 92 95 98 100
...**101011** 000 **10111** 0 **11** 00 **101 1** 0...

Añadiendo un nuevo bit

87 92 95 98 100 101 Nuevo bit
...**101011** 000 **10111** 0 **11** 00 **101 1** 0... ← **0**

87 92 95 98 100 101
...**101011** 000 **10111** 0 **11** 00 **101 1** 0 **0**...

No hay cambios en la cubeta!

EJEMPLO: ALGORITMO DGIM

Añadiendo un nuevo bit

... ⁸⁷101011 000 ⁹²10111 0 ⁹⁵11 00 ⁹⁸101 ¹⁰⁰1 ¹⁰¹0 0... ← ¹⁰²1 Nuevo bit

... ⁸⁷101011 000 ⁹²10111 0 ⁹⁵11 00 ⁹⁸101 ¹⁰⁰1 ¹⁰¹0 0 1.. Ingresamos el nuevo bit

... ⁸⁷101011 000 ⁹²10111 0 ⁹⁵11 00 ⁹⁸101 ¹⁰⁰1 ¹⁰¹00 ¹⁰²1.. Creamos una nueva cubeta de tamaño 1

2^2 2^2 2^1 2^1 2^0 2^0

EJEMPLO: ALGORITMO DGIM

Añadiendo un nuevo bit

87 92 95 98 100 101 102 103 Nuevo bit
 \dots **101011** 000 **10111** 0 **11** 00 **101** **1** 00 1.. \leftarrow **1**
 2^2 2^2 2^1 2^1 2^0 2^0

87 92 95 98 100 101 102 103
 \dots **101011** 000 **10111** 0 **11** 00 **101** **1** 00 1 **1**. Ingresamos el nuevo bit
 2^2 2^2 2^1 2^1 2^0 2^0

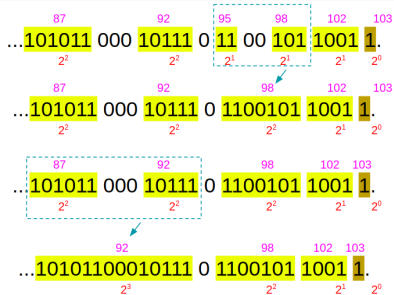
87 92 95 98 100 101 102 103
 \dots **101011** 000 **10111** 0 **11** 00 **101** **1** 00 1 **1**. Creamos una nueva cubeta de tamaño 1
 2^2 2^2 2^1 2^1 2^0 2^0 2^0

87 92 95 98 100 101 102 103
 \dots **101011** 000 **10111** 0 **11** 00 **101** **1** 00 1 **1**. Observamos que tenemos 3 cubetas de tamaño 1: 100,102,103
 2^2 2^2 2^1 2^1 2^0 2^0 2^0

87 92 95 98 102 103
 \dots **101011** 000 **10111** 0 **11** 00 **101** **1001** **1**. Combinamos las 2 cubetas más antiguas (más a la izquierda) y la marca de tiempo será la más reciente (la más a la derecha)
 2^2 2^2 2^1 2^1 2^1 2^0

EJEMPLO: ALGORITMO DGIM

Añadiendo un nuevo bit



Observamos que tenemos 3 cubetas de tamaño 2: 95, 98, 102

Combinamos las 2 cubetas más antiguas (más a la izquierda) y la marca de tiempo será la más reciente (la más a la derecha)

Observamos que tenemos 3 cubetas de tamaño 4: 98, 92, 87

Combinamos las 2 cubetas más antiguas (más a la izquierda) y la marca de tiempo será la más reciente (la más a la derecha)

EJEMPLO: ALGORITMO DGIM

Añadiendo un nuevo bit

... $\overset{87}{101011} \overset{92}{000} \overset{95}{10111} \overset{98}{011} \overset{102}{001} \overset{103}{1001} \overset{1}{1}.$
 $2^2 \quad 2^2 \quad 2^1 \quad 2^1 \quad 2^1 \quad 2^0$

Observamos que tenemos 3 cubetas de tamaño 2: 95, 98, 102

... $\overset{87}{101011} \overset{92}{000} \overset{98}{10111} \overset{102}{01100101} \overset{103}{1001} \overset{1}{1}.$
 $2^2 \quad 2^2 \quad 2^2 \quad 2^1 \quad 2^0$

Combinamos las 2 cubetas más antiguas (más a la izquierda) y la marca de tiempo será la más reciente (la más a la derecha)

... $\overset{87}{101011} \overset{92}{000} \overset{98}{1100101} \overset{102}{1001} \overset{103}{1}.$
 $2^2 \quad 2^2 \quad 2^2 \quad 2^1 \quad 2^0$

Observamos que tenemos 3 cubetas de tamaño 4: 98, 92, 87

... $\overset{92}{10101100010111} \overset{98}{0} \overset{102}{1100101} \overset{103}{1001} \overset{1}{1}.$
 $2^3 \quad 2^2 \quad 2^1 \quad 2^0$

Combinamos las 2 cubetas más antiguas (más a la izquierda) y la marca de tiempo será la más reciente (la más a la derecha)

Añadiendo un nuevo bit

Repetimos el proceso, ya sea 0 o 1

...

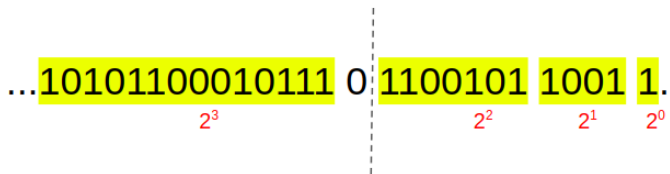
¿En qué momento se detiene el algoritmo?

- Continuamos cuando el tiempo actual menos el intervalo de tiempo más a la izquierda sea menor que N (tamaño de la venta)

Tiempo actual = 103, intervalo más a la izquierda = 92, $N = 24$
 $103 - 92 = 11 < 24$, **por lo tanto continuamos**

Cuando el resultado es mayor o igual, nos detenemos.

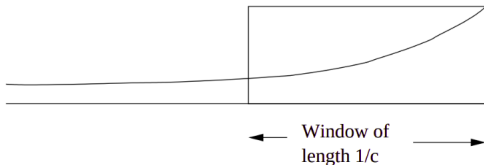
¿Cuántos 1s hay en los últimos 12 bits?



¿Cuántos 1s hay en los últimos 12 bits? $2^0 + 2^1 + 2^2 = 7$

- En el caso de películas, este algoritmo puede aplicarse (miles)
- Sin embargo, falla cuando estamos hablando de millones de registros como Amazon o Twitter.
- Otros algoritmos, como el desvanecimientos de ventanas puede ayudar definiendo un factor como 10^{-6} o 10^{-9}

Desvanecimiento de ventanas



$$\sum_{i=0}^{t-1} a_{t-i} (1-c)^i$$

Imagen tomada de Jure Leskovec, Anand Rajaraman, Jeff Ullman