



Repaso de Sumatorias Σ

Maira Diaz Marraffini

maira.marraffini@gmail.com  *maira1001001*  *Maira Diaz Marraffini*

1 de octubre, 2012

Índice

| | |
|--|-----------|
| 1. Porqué escribí este apunte | 2 |
| 2. Introducción a Sumatorias | 2 |
| 3. Notación | 2 |
| 3.1. Ejemplos | 3 |
| 4. Sumatorias genéricas | 4 |
| 4.0.1. Términos genéricos | 4 |
| 4.0.2. Límites genéricos o arbitrarios | 5 |
| 4.0.3. Ejemplos de sumatorias genéricas | 5 |
| 5. Manipulando sumatorias | 5 |
| 5.1. Sacar una constante fuera de una sumatoria ($\sum_{i=1}^n ca_i = c \times \sum_{i=1}^n a_i$) | 6 |
| 5.2. Sumatorias de constantes ($\sum_{i=1}^n c = c + c + \dots + c$ (n veces) = $c \times n$ (c constante)) | 8 |
| 5.3. Separar sumatorias ($\sum_{i=1}^n a_i + b_i = \sum_{i=1}^n a_i + \sum_{i=1}^n b_i$) | 9 |
| 5.4. Cortar sumatorias ($\sum_{i=1}^k a_i = \sum_{i=1}^n a_i - \sum_{i=k+1}^n a_i$, $\sum_{i=k}^n a_i = \sum_{i=1}^n a_i - \sum_{i=1}^{k-1} a_i$) | 10 |
| 6. Resumen de propiedades | 11 |
| 7. Resumen de fórmulas para simplificar algunas sumatorias comunes | 12 |
| 8. De código Java a expresión matemática | 12 |

9. Un poco de historia **12****10. Series y sumatorias** **13****1. Porqué escribí este apunte**

Este documento fue escrito para la cátedras "Algoritmos y Estructuras de datos" de la carrera [Lic. en Informática](#), [Lic. en Sistema](#) y [Analista Programador Universitario](#) cuando fui ayudante adscripta durante el 2012, cargo designado por la [Universidad Nacional de La Plata](#)

2. Introducción a Sumatorias

La notación de sumatoria es simplemente una forma de *escribir sumas complejas de manera más sencilla, compacta y fácil de manipular*. Generalmente, cuando tenemos una sumatoria que representa el tiempo de ejecución de un algoritmo, buscamos *resolver* la sumatoria, es decir, expresar su resultado de una forma más simple. Lo esencial al resolverlas es nunca olvidar que lo único que hace es sumar números, la operación aritmética más fácil de aplicar.

3. Notación

$$\sum_{i=valorInicial}^{valorFinal} a_i \quad (1)$$

Esto nos dice que tenemos que sumar el valor del término, a_i , para los valores de i , desde *valorInicial* hasta *valorFinal*. El valor del término, a_i no es un número fijo, sino que es un valor que depende de la variable de índice i . Por ejemplo, la sumatoria de los primeros cuatro números naturales se escribe

$$\sum_{i=1}^4 i = 1 + 2 + 3 + 4 = 10 \quad (2)$$

y tiene como valor inicial a 1, como valor final a 4, como variable de índice a i (abajo del símbolo \sum , donde dice $i = 1$), $a_i = i$ (a la derecha del símbolo \sum), y el resultado es 10. Si cambiamos los valores iniciales y finales, generalmente llamados los *límites* de la sumatoria, tenemos:

$$\sum_{i=3}^5 i = 3 + 4 + 5 = 12 \quad (3)$$

Con valor inicial 3, valor final 5, variable de índice i , $a_i = i$, y 12 de resultado. Cambiando el valor del término por i^2 , y volviendo a los valores iniciales y finales del primer ejemplo, tenemos:

$$\sum_{i=1}^4 i^2 = 1^2 + 2^2 + 3^2 + 4^2 = 1 + 4 + 9 + 16 = 30 \quad (4)$$

Con $a_i = i^2$, y 30 de resultado. Si cambiamos nuevamente el valor del término:

$$\sum_{i=1}^4 (i * 2 + 1) = (1*2+1) + (2*2+1) + (3*2+1) + (4*2+1) = 3 + 5 + 7 + 9 = 24 \quad (5)$$

La sumatoria tiene como valor inicial a 1, como valor final a 4, variable de índice a i , $a_i = i * 2 + 1$, y su resultado es 24. Finalmente,

$$\sum_{i=1}^4 2 = 2 + 2 + 2 + 2 = 8 \quad (6)$$

Tenemos la misma variable de índice y los mismos valores iniciales y finales que antes, pero el valor del término es constante ($a_i = 2$), por ende simplemente se suma 4 veces.

3.1. Ejemplos

$$\begin{array}{ll} \sum_{i=1}^4 (i + 1) = 2 + 3 + 4 + 5 & \sum_{i=2}^5 (i - 1) = 1 + 2 + 3 + 4 \\ \sum_{i=1}^{1350} i = 1 + \dots + 1350 & \sum_{i=1}^4 2^i = 2^1 + 2^2 + 2^3 + 2^4 = 2 + 4 + 8 + 16 \\ \sum_{i=2}^5 2 = 2 + 2 + 2 + 2 = 8 & \sum_{i=1}^4 c = c + c + c + c \text{ (c es una constante)} \end{array}$$

4. Sumatorias genéricas

4.0.1. Términos genéricos

Muchas veces queremos hablar de sumatorias de una forma genérica o general, especialmente para probar propiedades de las mismas. En este caso, lo más común es que no nos importe cuál es el término en particular de la sumatoria: podría ser i , i^2 , $i * 2 + 1$, 4 , 2^i o lo que sea; no nos interesa. Entonces, es válido escribir la sumatoria como:

$$\sum_{i=1}^4 a_i = a_1 + a_2 + a_3 + a_4 = ?$$

Como antes, el valor inicial es 1, el final 4, pero el valor del término no está especificado, por eso lo llamamos a_i ; la a nos dice que el término podría ser cualquier cosa, el subíndice i nos dice que el término depende de i . Obviamente, hablar del resultado definitivo de esta sumatoria no tiene mucho sentido ya que no sabemos cuáles son sus términos.

Por supuesto, la a en a_i no tiene nada de especial; podríamos haber escrito la sumatoria anterior como:

$$\sum_{i=1}^4 b_i = b_1 + b_2 + b_3 + b_4$$

O:

$$\sum_{i=1}^4 c_i = c_1 + c_2 + c_3 + c_4$$

También podemos escribir:

$$\sum_{i=1}^4 (a_i + b_i) = (a_1 + b_1) + (a_2 + b_2) + (a_3 + b_3) + (a_4 + b_4)$$

En este caso, cada término de la sumatoria es $a_i + b_i$, o sea, la suma de 2 términos que dependen de i . Un ejemplo de una sumatoria que tiene esta forma podría ser:

$$\sum_{i=1}^4 (i + i^2) = (1 + 1^2) + (2 + 2^2) + (3 + 3^2) + (4 + 4^2)$$

donde $a_i = i$ y $b_i = i^2$.

4.0.2. Límites genéricos o arbitrarios

Casi todas la sumatorias que encontramos para representar tipos de ejecución no tienen un valor final definido, sino que dependen del tamaño del problema que resuelve el algoritmo, como el tamaño de un arreglo, la altura de un árbol, etc. Si utilizamos la letra n para representar dicho tamaño la sumatoria queda como:

$$\sum_{i=1}^n a_i = a_1 + a_2 + \cdots + a_{n-1} + a_n = ?$$

La sumatoria de los primeros n números naturales se puede escribir como:

$$\sum_{i=1}^n i = 1 + 2 + \dots + (n-1) + n$$

4.0.3. Ejemplos de sumatorias genéricas

$$\begin{array}{ll} \sum_{i=1}^n (i+1) = 2 + 3 + \cdots + n + (n+1) & \sum_{i=1}^n 2^i = 2^1 + \cdots + 2^{n-1} + 2^n \\ \sum_{i=1}^n a_i = a_1 + a_2 + \cdots + a_{n-1} + a_n & \sum_{i=1}^{2n} i = 1 + 2 + \cdots + (2(n-1)) + 2n \\ \sum_{i=1}^{2n} a_i = a_1 + a_2 + \cdots + a_{2(n-1)} + 2a_n & \sum_{i=1}^{\log_2 n} i = 1 + 2 + \cdots + \log_2 n \end{array}$$

5. Manipulando sumatorias

De nuevo: la manera más fácil de manipular sumatorias es recordar que solamente estamos sumando números. Por ejemplo, es posible escribir $1 + 2 + 3 + 4$ de muchas formas diferentes:

$$\begin{array}{l} \sum_{i=1}^4 i = \sum_{i=1}^3 i + 4 = \sum_{i=1}^2 i + 3 + 4 = \sum_{i=1}^2 i + \sum_{i=3}^4 i = \\ \sum_{i=1}^1 i + 2 + \sum_{i=3}^4 i = 1 + 2 + \sum_{i=3}^4 i = 1 + 2 + \sum_{i=3}^3 i + 4 = 1 + 2 + 3 + 4 = 10 \end{array}$$

A continuación veremos algunas de las técnicas más usuales para manipular sumatorias; el objetivo es, siempre, simplificar la sumatoria, llevarla a una expresión que sea fácil de resolver, que ya conozcamos.

5.1. Sacar una constante fuera de una sumatoria ($\sum_{i=1}^n ca_i = c \times \sum_{i=1}^n a_i$)

Lejos, la operación más común al manipular sumatorias es sacar una constante afuera de la misma. Consideremos:

$$\sum_{i=1}^4 2i$$

Es fácil resolver esta sumatoria directamente: $2 + 4 + 6 + 8 = 20$. Pero si el límite superior no fuese 4, si fuera 4000 o n , sería más complicado. Así que, intentemos de otra manera.

$$\sum_{i=1}^4 2i = 2 \times 1 + 2 \times 2 + 2 \times 3 + 2 \times 4$$

Podemos sacar factor común con el 2 que está multiplicando a todos los términos, entonces tenemos:

$$\sum_{i=1}^4 2i = 2 \times 1 + 2 \times 2 + 2 \times 3 + 2 \times 4 = 2 \times (1 + 2 + 3 + 4)$$

Pero $1 + 2 + 3 + 4$ es una sumatoria que ya conocemos (y a esta altura ya nos aburre bastante): $\sum_{i=1}^4 i$. Utilizando esto:

$$\sum_{i=1}^4 2i = 2 \times 1 + 2 \times 2 + 2 \times 3 + 2 \times 4 = 2 \times (1 + 2 + 3 + 4) = 2 \times \sum_{i=1}^4 i$$

O sea:

$$\sum_{i=1}^4 2i = 2 \times \sum_{i=1}^4 i$$

En este caso, la constante que sacamos afuera fue el 2, pero esto funciona para cualquier constante, o sea:

$$\sum_{i=1}^4 ci = c \times 1 + c \times 2 + c \times 3 + c \times 4 = c \times (1 + 2 + 3 + 4) = c \times \sum_{i=1}^4 i$$

Y, de hecho, en este caso el término es ci , pero podemos hablar de un término más genérico, como ca_i , y tendríamos:

$$\sum_{i=1}^4 ca_i = c \times a_1 + c \times a_2 + c \times a_3 + c \times a_4 = c \times (a_1 + a_2 + a_3 + a_4) = c \times \sum_{i=1}^4 a_i$$

Por último, esta idea claramente funciona con límites arbitrarios, dado que si bien tenemos más términos, seguimos simplemente sumando:

$$\sum_{i=1}^n ca_i = c \times a_1 + c \times a_2 + \cdots + c \times a_n = c \times (a_1 + a_2 + \cdots + a_n) = c \times \sum_{i=1}^n a_i$$

En estos dos últimos pasos, generalizamos nuestro ejemplo ($\sum_{i=1}^4 2i = 2 \times \sum_{i=1}^4 i$) utilizando términos y límites arbitrarios; haremos lo mismo cuando veamos el resto de las técnicas.

Muchas veces vamos a necesitar aplicar esta técnica aunque el término de la sumatoria no este escrito exactamente como $c \times a_i$. Por ejemplo,

$$\sum_{i=1}^4 2^{i+3} = 2^4 + 2^5 + 2^6 + 2^7$$

tiene como término 2^{i+3} ; sería más fácil si dicho término fuera 2^i . Podemos lograr esto viendo que:

$$2^{i+3} = 2^i \times 2^3$$

Por ende:

$$\sum_{i=1}^4 2^{i+3} = \sum_{i=1}^4 (2^3 \times 2^i) = 2^3 \times \sum_{i=1}^4 2^i$$

5.2. Sumatorias de constantes ($\sum_{i=1}^n c = c+c+\dots+c$ (n veces) = $c \times n$ (c constante))

Un caso especial muy importante de la técnica anterior se da cuando tenemos la sumatoria de un término constante, como:

$$\sum_{i=1}^4 2 = 2 + 2 + 2 + 2$$

Podemos encarar la simplificación desde 2 perspectivas similares. Por un lado, $2 + 2 + 2 + 2 = 2 \times 4 = 8$. Generalizando el término a una constante c :

$$\sum_{i=1}^4 c = c + c + c + c \text{ (4 veces) } = c \times 4$$

Generalizando el límite:

$$\sum_{i=1}^n c = c + c + \dots + c \text{ (n veces) }$$

La otra forma de verlo, quizás un poco más difícil de entender pero también un poco más elegante, es que aplicando la propiedad anterior:

$$\sum_{i=1}^n c = \sum_{i=1}^n (c \times 1) = c \sum_{i=1}^n 1$$

Está claro que $\sum_{i=1}^n 1 = n$, ya que simplemente sumamos n veces 1. Entonces, de esta manera también llegamos a que:

$$\sum_{i=1}^n c = c \sum_{i=1}^n 1 = c \times n$$

Algo muy importante a notar es que los límites de la sumatoria son los que definen por cuanto multiplicamos a c . Por ejemplo,

$$\begin{aligned}
\sum_{i=1}^3 c &= c + c + c = c \times 3 \\
\sum_{i=2}^4 c &= c + c + c = c \times 3 \\
\sum_{i=2}^5 c &= c + c + c + c = c \times 4 \\
\sum_{i=0}^4 c &= c + c + c + c + c = c \times 5 \\
\sum_{i=1}^{n-1} c &= c + c + \cdots + c = c \times (n-1) \\
\sum_{i=2}^n c &= c + c + \cdots + c = c \times (n-1) \\
\sum_{i=0}^n c &= c + c + \cdots + c = c \times (n+1) \\
\sum_{i=0}^{n-1} c &= c + c + \cdots + c = c \times n
\end{aligned}$$

En general, si la sumatoria tiene límites m y n constantes, tenemos:

$$\sum_{i=m}^n c = ((n-m) + 1) \times c \quad (7)$$

5.3. Separar sumatorias ($\sum_{i=1}^n a_i + b_i = \sum_{i=1}^n a_i + \sum_{i=1}^n b_i$)

Supongamos que tenemos una sumatoria como:

$$\sum_{i=1}^3 i + 2^i = (1 + 2^1) + (2 + 2^2) + (3 + 2^3)$$

Cada término es $a_i = i + 2^i$; sería más fácil si pudiésemos tener dos sumatorias, para tratar el i y el 2^i por separado. Dado que del lado derecho de la ecuación sólo estamos sumando, podemos reordenar los términos:

$$(1 + 2^1) + (2 + 2^2) + (3 + 2^3) = (1 + 2 + 3) + (2^1 + 2^2 + 2^3) = \sum_{i=1}^3 i + \sum_{i=1}^3 2^i$$

Y entonces:

$$\sum_{i=1}^3 i + 2^i = \sum_{i=1}^3 i + \sum_{i=1}^3 2^i$$

Es posible aplicar esta técnica siempre que nuestro término a_i sea la suma de dos (o varias) cosas. La recordamos como:

$$\sum_{i=1}^n a_i + b_i = \sum_{i=1}^n a_i + \sum_{i=1}^n b_i$$

5.4. Cortar sumatorias ($\sum_{i=1}^k a_i = \sum_{i=1}^n a_i - \sum_{i=k+1}^n a_i$, $\sum_{i=k}^n a_i = \sum_{i=1}^n a_i - \sum_{i=1}^{k-1} a_i$)

Es fácil ver que:

$$1 + 2 + 3 + 4 = (1 + 2) + (3 + 4)$$

Siguiendo esa idea, y recordando que con una sumatoria sólo estamos sumando, vemos que:

$$\sum_{i=1}^4 i = 1 + 2 + 3 + 4 = (1 + 2) + (3 + 4) = \left(\sum_{i=1}^2 i\right) + \left(\sum_{i=3}^4 i\right)$$

Generalizando, tenemos:

$$\sum_{i=1}^n a_i = \sum_{i=1}^k a_i + \sum_{i=k+1}^n a_i$$

Por otro lado, como $0 = (3 + 4) - (3 + 4)$

$$1 + 2 = (1 + 2) + (3 + 4) - (3 + 4) = (1 + 2 + 3 + 4) - (3 + 4)$$

Entonces

$$\sum_{i=1}^2 i = 1 + 2 = (1 + 2 + 3 + 4) - (3 + 4) = \left(\sum_{i=1}^4 i\right) - \left(\sum_{i=3}^4 i\right)$$

Generalizando:

$$\sum_{i=1}^k a_i = \sum_{i=1}^n a_i - \sum_{i=k+1}^n a_i$$

Por último,

$$3 + 4 = (3 + 4) + (1 + 2) - (1 + 2) = (1 + 2 + 3 + 4) - (1 + 2)$$

Entonces

$$\sum_{i=3}^4 i = 3 + 4 = (1 + 2 + 3 + 4) - (1 + 2) = \left(\sum_{i=1}^4 i\right) - \left(\sum_{i=1}^2 i\right)$$

Generalizando:

$$\sum_{i=k}^n a_i = \sum_{i=1}^n a_i - \sum_{i=1}^{k-1} a_i$$

6. Resumen de propiedades

$$1. \sum_{i=1}^n c \cdot a_i = c \times \sum_{i=1}^n a_i$$

$$\text{Ejemplo: } \sum_{i=1}^{73} 5i = 5 \times \sum_{i=1}^{73} i$$

$$2. \sum_{i=1}^n c = (n - 1 + 1) \times c = n \times c$$

$$\text{Ejemplo: } \sum_{i=7}^{19} cte = (19 - 7 + 1) \times cte = 13 \times cte$$

$$3. \sum_{i=1}^n a_i + b_i = \sum_{i=1}^n a_i + \sum_{i=1}^n b_i$$

$$\text{Ejemplo: } \sum_{i=1}^n i + 3i^2 = \sum_{i=1}^n i + \sum_{i=1}^n 3i^2$$

$$4. \sum_{i=1}^n a_i = \sum_{i=1}^k a_i + \sum_{i=k+1}^n a_i$$

$$\text{Ejemplo: } \sum_{i=1}^n 3i^2 = \sum_{i=1}^5 3i^2 + \sum_{i=6}^n 3i^2$$

$$5. \sum_{i=k}^n a_i = \sum_{i=1}^n a_i - \sum_{i=1}^{k-1} a_i$$

$$\text{Ejemplo: } \sum_{i=7}^{51} (cte \cdot i + 3) = \sum_{i=1}^{51} (cte \cdot i + 3) - \sum_{i=1}^{50} (cte \cdot i + 3)$$

$$6. \sum_{i=1}^n a_i \times b_i \neq \sum_{i=1}^n a_i \times \sum_{i=1}^n b_i$$

$$7. \sum_{i=1}^n a_i^2 \neq \left(\sum_{i=1}^n a_i\right)^2$$

7. Resumen de fórmulas para simplificar algunas sumatorias comunes

1. $\sum_{i=1}^n i = \frac{n(n+1)}{2}$
2. $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$
3. $\sum_{i=1}^n i^3 = \left(\frac{n(n+1)}{2}\right)^2$
4. $\sum_{i=1}^n i^4 = \frac{n(n+1)(2n+1)(3n^2+3n-1)}{30}$
5. $\sum_{i=1}^n a^i = \frac{a^{n+1}-a}{a-1}$

8. De código Java a expresión matemática

Tenemos una parte del código en Java y queremos expresar con fórmula matemática la cantidad de veces que se ejecuta dicho algoritmo.

```
for(int i=2; i<n; i++){
    for(int j=i; j<=n; j++){
        algo_de_orden(1)
    }
}
```

Tenemos que realizar la suma de todas las iteraciones del ciclo. El tiempo de ejecución de un algoritmo debe medirse en función del tamaño de los datos de entrada que debe procesar. En nuestro caso, la cantidad de veces que se realiza depende de n . Si lo expresamos con una sumatoria, quedaría de la siguiente forma:

$$\sum_{i=2}^{n-1} \sum_{j=i}^n cte$$

9. Un poco de historia

Érase una vez un niño alemán llamado **Carl Friedrich Gauss**. Cuando tenía diez años, en 1787, su profesor de la escuela, enojado porque sus alumnos se portaban mal, les puso un problema matemático al pequeño Carl y a sus compañeros. Los niños debían sumar todos los números del 1 al 100;

es decir, $1 + 2 + 3 + 4 + 5 + 6 + \dots + 99 + 100$. El profesor se sentó en su silla a leer el periódico, confiaba en que tendría horas hasta que los niños sumaran todos los números. Sin embargo, el pequeño Gauss no tardó ni cinco minutos en ir hacia el profesor y darle el resultado: 5050. *¿Cómo lo había hecho?*

Gauss tenía que sumar lo siguiente: $1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + \dots + 95 + 96 + 97 + 98 + 99 + 100$

Se dio cuenta de que reordenar los elementos de esta suma, sumando siempre los simétricos, facilitaba enormemente las cosas:

$$\left\{ \begin{array}{rclcl} 1 & + & 100 & = & 101 \\ 2 & + & 99 & = & 101 \\ 3 & + & 98 & = & 101 \\ 4 & + & 97 & = & 101 \\ 5 & + & 96 & = & 101 \\ \dots & & & & \\ 46 & + & 55 & = & 101 \\ 47 & + & 54 & = & 101 \\ 48 & + & 53 & = & 101 \\ 49 & + & 52 & = & 101 \\ 50 & + & 51 & = & 101 \end{array} \right\} \text{ 50 veces 101, es decir 50x101}$$

De donde se deduce la fórmula de la sumatoria de los n primeros números:

$$\sum_{i=1}^n i = \frac{n}{2} \cdot (n + 1) = \frac{n \cdot (n + 1)}{2}$$

10. Series y sumatorias

Generalmente no distinguimos entre series y sumatorias, pero técnicamente la sumatoria es solamente la *notación* para escribir **series**. Intuitivamente, una serie es simplemente la suma de ciertos términos. Si bien esto suena muy parecido a una sumatoria, la diferencia, entonces, es similar a la diferencia entre la sintaxis del for, que varía dependiendo del lenguaje, y el concepto del for o la iteración. La sintaxis del for sirve para darle una representación al concepto del for, de la misma forma que una sumatoria sirve para representar una serie.

Podemos considerar dos tipos de series, las finitas y las infinitas. Las que utilizamos para resolver los tiempos de ejecución pertenecen al grupo de las

series finitas, donde siempre sumamos una cantidad finita de términos. En este caso, podemos ver a las series simplemente como una suma de números.

Cuando la cantidad de términos a sumar es **infinita**, por ejemplo, $4 + 4 + 4 + 4 + \dots = \sum_{i=1}^{\infty} 4$, estamos lidiando con una **serie infinita**, entonces las propiedades usuales que vimos en esta sección no funcionan siempre, y no podemos ver a la sumatoria simplemente como una forma de escribir sumas de manera más sencilla. Por ejemplo, la serie $\sum_{i=1}^{\infty} (-1)^n = (-1) + 1 + (-1) + 1 + \dots$ no da 0, sino que está indefinida, o sea, no es **sumable**.