

Sumatorios y Productorios

Introducción

Saber introducir sumatorios y productorios en los algoritmos es esencial para el buen desarrollo de la asignatura. Sin embargo, a veces se dan un poco por encima o es la primera vez que te enfrentas a ellos y te puedes sentir un poco perdido. No pasa nada, es muy sencillo.

Sumatorios

Esta operación nos indica una serie de sumas cuyos miembros tienen algo en común.

Se escriben de la siguiente forma:

$$\sum_{i=1}^n a_i = a_1 + a_2 + a_3 + \dots + a_n$$

$i=1$ nos dice que i empieza en 1 e irá aumentando de uno en uno hasta llegar a n , el número de elementos que tenemos que sumar. Esos elementos están marcados por a_i (suele estar indicado por una fórmula donde solo hay que sustituir i).

Recordatorio: i , n y a son variables que pueden recibir cualquier otro nombre

Con un ejemplo se ve mejor. Si queremos escribir la suma de los 5 primeros números enteros

empezando en el 1, pondríamos $\sum_{i=1}^5 i$. El primer término sería 1, luego $+2 = 3$, $3 + 3 = 6$, $6 + 4 = 10$, $10 +$

$5 = 15$, y ya estaría. Se podrían usar otras fórmulas, como la suma de los n primeros números pares ($2i$, $2 + 4 + 6 + 8 \dots$), refiriéndonos también a las componentes de un vector o una matriz (a los que llamaremos v_i -componente del vector v en posición i - o $A_{i,j}$ -componente de la fila i , columna j)

Productorios

Esta operación nos indica una serie de multiplicaciones cuyos miembros tienen algo en común. Se escriben de la siguiente forma:

$$\prod_{i=1}^n a_i = a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot a_n$$

Funcionan igual que los sumatorios. La i avanza de uno en uno empezando en 1 ($i=1$) y acaba en n . La i se sustituirá en una fórmula o se referirá a la componente de un vector...

Podemos poner de ejemplo $\prod_{i=1}^n i$. El primer término sería 1, el segundo 2, el tercero 3 y así hasta n. Es decir, estaríamos haciendo $1 \cdot 2 \cdot 3 \dots \cdot n$, o lo que es lo mismo, el factorial de n.

Uso en algoritmia

Si pensamos un poco en estas operaciones, veremos rápidamente que tienen algo en común: son muy repetitivas. En ambas repites una misma operación una y otra vez. ¿Y qué estructura de algoritmia hace lo mismo una y otra vez? Los bucles, así que eso es lo que usaremos para construir nuestro algoritmo. Vamos a usar un ejemplo, el producto escalar de dos vectores.

Lo primero es buscar la fórmula que vamos a usar. Supongamos que vamos a usar dos vectores del mismo tamaño, V y U. Sabemos que el producto escalar es la primera componente de V por la primera componente de U **más** la segunda componente de V por la primera componente de U **más**... O lo que es lo mismo $V_1U_1 + V_2U_2 + V_3U_3 \dots$ es decir $\sum V_iU_i$. Como tenemos que empezar en la primera componente, $i=1$ y tenemos que llegar hasta la última. En este caso, supongamos que U y V tienen un tamaño n. La

fórmula nos quedaría así $\sum_{i=1}^n V_iU_i$. Ahora vamos a transformarlo en un bucle.

Empecemos definiendo el bucle. Los bucles están definidos como "Para [variable] desde [inicio] hasta [final]". La variable no es problema, se suele elegir i, j, k, etc. pero puede ser cualquier cosa. Para el inicio, cogemos $i=1$, es decir 1, y para el final, n. De modo que nos queda "Para i desde 1 hasta n".

Ahora vamos con la operación que realiza el bucle, lo que hay dentro. Eso sería lo que hay después del \sum , en este caso, V_iU_i . Si guardamos el valor del sumatorio en, por ejemplo, la variable SUM, nos quedaría el bucle tal que así

Para i desde 1 hasta n

SUM = SUM + V_iU_i #se suma la variable una y otra vez porque es un sumatorio

Fin del bucle

Sin embargo, si lo dejamos así tenemos un problema, ya que la primera vez que se lea el bucle, la variable SUM no está definida, el ordenador no la podría encontrar. Tenemos que predefinir la variable SUM. Como es un sumatorio, basta con colocar antes del bucle $SUM=0$, al ser una suma, sumar cero no afecta al resultado (si fuera un productorio, habría que predefinir $SUM=1$, ya que multiplicar por 1 no altera el resultado). Finalmente, nos quedaría el bucle como

SUM=0

Para i desde 1 hasta n

SUM = SUM + V_iU_i

Fin del bucle

Metainformación

Tema	Tiempo	Tipo	Destinatario
Algoritmia Productorios y sumatorios	Diciembre, 2021	Resumen	Alumno que está muy perdido con sumatorios y productorios y no sabe cómo introducirlo a algoritmia