Divide y Vencerás: Cuatro Problemas Clásicos

Moda de un vector, Multiplicación de enteros grandes, Multiplicación de matrices, y Subsecuencia de suma máxima

Preparado por ChatGPT

7 de septiembre de 2025

Planteamiento

Dado un vector A de n elementos (no necesariamente ordenado), hallar la **moda**: el valor con mayor frecuencia.

Restricción: Usar el paradigma *divide y vencerás*.

Idea Divide y Vencerás

- Dividir A en dos mitades A_L y A_R.
- Resolver recursivamente obteniendo *mapas de frecuencias* F_L y F_R.
- Combinar: fusionar los mapas sumando frecuencias clave a clave.
- La moda global se obtiene como $\arg\max_x F[x]$ del mapa fusionado.

Nota: La moda global *no* siempre es la moda de una mitad; por ello se fusionan frecuencias completas.

Pseudocódigo

Algorithm 1 Moda_DyV(A)

```
if |A| ≤ k then                                   ▷ caso base pequeño
    return mapa de frecuencias directo
dividir A en A_L, A_R
F_L ← Moda_DyV(A_L);    F_R ← Moda_DyV(A_R)
F ← mapa vacío
for all x en claves(F_L) do F[x] ← F[x] + F_L[x]
for all x en claves(F_R) do F[x] ← F[x] + F_R[x]
return F                                          ▷ el cliente toma arg máx_x F[x]
```

Complejidad (paso a paso)

Sea $T(n)$ el costo al procesar n elementos. La fusión es lineal.

Recurrencia:

$$T(n) = 2\,T(n/2) + c\,n, \quad T(k) = \Theta(k)$$

Por el Teorema Maestro (caso 2): $a{=}2, b{=}2, f(n){=}\Theta(n)$.
$n^{\log_b a} = n$. Entonces $T(n) = \Theta(n \log n)$.

Espacio: profundidad $\log n$, más mapas parciales $\Rightarrow \mathcal{O}(n)$ adicional.

Planteamiento

Multiplicar dos enteros no negativos X y Y de n dígitos (en base 10) usando un algoritmo de *divide y vencerás*.

Idea (Karatsuba)

Escribir $X = x_1 \cdot 10^m + x_0$, $Y = y_1 \cdot 10^m + y_0$ con $m \approx n/2$.
Producto clásico requiere 4 productos de tamaño m. Karatsuba reduce a 3:

$$\begin{aligned} z_0 &= x_0 y_0 \\ z_2 &= x_1 y_1 \\ z_1 &= (x_1+x_0)(y_1+y_0) - z_2 - z_0 \\ XY &= z_2 \cdot 10^{2m} + z_1 \cdot 10^m + z_0 \end{aligned}$$

Pseudocódigo

Algorithm 2 Karatsuba(X, Y)

if $X < 10$ **or** $Y < 10$ **then**
    return $X \cdot Y$
$m \leftarrow \lfloor \text{máx}(\text{dígitos}(X), \text{dígitos}(Y))/2 \rfloor$
dividir X en (x_1, x_0) y Y en (y_1, y_0) por 10^m
$z_0 \leftarrow \text{Karatsuba}(x_0, y_0)$
$z_2 \leftarrow \text{Karatsuba}(x_1, y_1)$
$z_1 \leftarrow \text{Karatsuba}(x_1 + x_0, y_1 + y_0) - z_2 - z_0$
return $z_2 \cdot 10^{2m} + z_1 \cdot 10^m + z_0$

Complejidad (paso a paso)

Recurrencia:

$$T(n) = 3\,T(n/2) + c\,n$$

- Árbol de recursión: niveles $0..\log_2 n$; en nivel i hay 3^i subproblemas de tamaño $n/2^i$ con costo lineal de combinación $\Theta(n/2^i)$ cada uno $\Rightarrow$ costo por nivel $\Theta(3^i \cdot n/2^i) = \Theta(n(3/2)^i)$.
- El último nivel domina: $i = \log_2 n \Rightarrow \Theta\left(n^{\log_2 3}\right)$.

Por Teorema Maestro (caso 1.5): $T(n) = \Theta\left(n^{\log_2 3}\right) \approx \Theta(n^{1{,}585})$.

Planteamiento

Multiplicar dos matrices cuadradas A y B de tamaño $n \times n$ con n potencia de 2, usando *divide y vencerás*.

Idea (Strassen)

Dividir cada matriz en 4 subbloques $n/2 \times n/2$ y combinar con 7 productos en lugar de 8:

$$
\begin{aligned}
M_1 &= (A_{11}+A_{22})(B_{11}+B_{22}) & M_2 &= (A_{21}+A_{22})B_{11} \\
M_3 &= A_{11}(B_{12}-B_{22}) & M_4 &= A_{22}(B_{21}-B_{11}) \\
M_5 &= (A_{11}+A_{12})B_{22} & M_6 &= (A_{21}-A_{11})(B_{11}+B_{12}) \\
M_7 &= (A_{12}-A_{22})(B_{21}+B_{22})
\end{aligned}
$$

y luego

$$
\begin{aligned}
C_{11} &= M_1 + M_4 - M_5 + M_7 \\
C_{12} &= M_3 + M_5 \\
C_{21} &= M_2 + M_4 \\
C_{22} &= M_1 - M_2 + M_3 + M_6
\end{aligned}
$$

Pseudocódigo

Algorithm 3 Strassen(A, B)

```
if n ≤ n_0 then
    return multiplicación clásica
Particionar A, B en subbloques n/2 × n/2
Calcular M_1, ..., M_7 recursivamente
Combinar en C_11, C_12, C_21, C_22
return C
```

Complejidad (paso a paso)

Recurrencia:

$$T(n) = 7\,T(n/2) + c\,n^2$$

Por Teorema Maestro: $a{=}7$, $b{=}2$, $n^{\log_b a} = n^{\log_2 7} \approx n^{2{,}807}$. Como $f(n) = \Theta(n^2) = o(n^{\log_2 7})$, domina la parte recursiva.

Resultado: $T(n) = \Theta(n^{\log_2 7}) \approx \Theta(n^{2{,}807})$.

En la práctica se usa umbral n_0 para cambiar a multiplicación clásica por constantes.

Planteamiento

Dado un arreglo A de enteros (positivos y negativos), hallar la **subsecuencia contigua** de suma máxima.

Idea Divide y Vencerás

- Dividir en mitades A_L y A_R.
- La mejor subarreglo es el máximo de:
  - mejor en A_L,
  - mejor en A_R,
  - mejor que *cruza* el centro (máx. sufijo en A_L + máx. prefijo en A_R).

Pseudocódigo

Algorithm 4 MaxSubarray(A)

if $|A| = 1$ **then**
    return $A[0]$
dividir A en A_L, A_R
$bestL \leftarrow \text{MaxSubarray}(A_L)$; $bestR \leftarrow \text{MaxSubarray}(A_R)$
$cross \leftarrow \text{mejor\_sufijo}(A_L) + \text{mejor\_prefijo}(A_R)$
return $\max(bestL, bestR, cross)$

Complejidad (paso a paso)

La búsqueda de prefijo/sufijo es lineal. Recurrencia:

$$T(n) = 2\,T(n/2) + c\,n \Rightarrow T(n) = \Theta(n \log n).$$

Nota: Existe una solución lineal (*Kadane*) $\Theta(n)$, pero aquí se usa DyV.

Cierre

El paradigma *divide y vencerás* permite diseñar soluciones modulares y analizables:

Problema	**Complejidad**
Moda (fusión de frecuencias)	$\Theta(n \log n)$
Karatsuba (enteros grandes)	$\Theta(n^{\log_2 3})$
Strassen (matrices)	$\Theta(n^{\log_2 7})$
Subsecuencia de suma máxima	$\Theta(n \log n)$