

$$T(n) \begin{cases} T(1) = 1 \\ T(n) = 2T\left(\frac{n}{3}\right) + n - 1 \end{cases}$$

$$n = 3^m$$

k iterador

$$(T(3^m) - 2T(3^{m-1}) = 3^m - 1) 2^0$$

$$(T(3^{m-1}) - 2T(3^{m-2}) = 3^{m-1} - 1) 2^1$$

$$(T(3^{m-2}) - 2T(3^{m-3}) = 3^{m-2} - 1) 2^2$$

$$\left(\underbrace{T(3^k)}_{a_k} - 2 \underbrace{T(3^{k-1})}_{a_{k-1}} = 3^k - 1 \right) 2^{m-k}$$

$$a_k = 2^{m-k} T(3^k)$$

$$a_n - a_0$$

| | | | | | | |
|---|-----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| A | -23 | 18 | 20 | -7 | 12 | -8 |

Maximum subarray

Divide & Conquer



Algo(A, p, r) // devuelve el max. subarray de A[p...r]
 if (p == r)
 return A[p]

$$q = \lfloor (p+r)/2 \rfloor$$

$$T_1 = \text{Algo}(A, p, q)$$

$$T_2 = \text{Algo}(A, q+1, r)$$

$$T_3 = \text{Cruzado}(A, p, q, r)$$

return max {T₁, T₂, T₃}

Complejidad:

$$T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \underbrace{\Theta(n)}_{\text{Extremo}} = \Theta(n \lg n)$$

Multiplicación Básica

Algo(A, B)

$$\begin{array}{r} 23 \\ 45 \\ \hline 15 \\ 10 \\ 12 \\ 8 \\ \hline 1035 \end{array}$$

$$\begin{array}{r} 12 \quad 34 \\ 56 \quad 78 \\ \hline \end{array}$$

Counting inversions

1 2 3 4 5

Ordenado

0

1 2 3 5 4

casi ordenado

1

5 1 4 2 3

desordenado

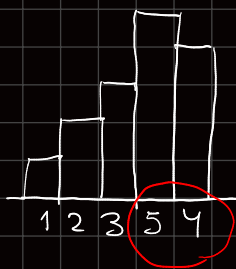
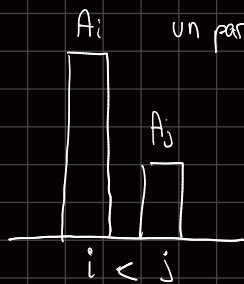
6

5 4 3 2 1

El array más desordenado

10

• Queremos un parámetro que nos ayude a definir qué tan desordenado está un array



El valor de la izquierda es mayor del valor de la derecha.

1
= Inversión