Example 3.5. *Let* $T : \mathbb{N} \to \mathbb{R}^+$ *defined as*

$T(1) = 1 \leq C1 - 1$

$T(1) = C \geq 2$

$P(\lfloor n/2 \rfloor)$

$$T(n) = \begin{cases} 1 & n = 1 \\ 2T(\lfloor \frac{n}{2} \rfloor) + 1 & \textit{otherwise} \end{cases}$$

Prove by induction that $T(n) = O(n)$.

$\cancel{P(n-1)} \to P(n)$

Queremos probar que $T(n) \leq \boxed{Cn + D}\ \forall n \geq n_0$, Con $C = 2$, $\boxed{D = -1}$

$\cdot\ P(n) = T \quad \forall n \geq n_0$.

depende del problema

CB: Probar manualmente, $P(n_0), \ldots P(n_1)$

HI Para un $k > n_1$ Sg $\boxed{\begin{matrix} P(n_0) \\ P(n_1) \\ \vdots \\ P(k-1) \end{matrix}} = T$

TT. ¿ $P(k)$?

$T \wedge P(n_0) \wedge P(n_1) \wedge \ldots \wedge P(k-1) \longrightarrow P(k)$

$\boxed{T(k) \leq Ck}$

C.B ¿$P(1)$? $T(1) \leq 2 - 1$ ✓

HI Sg $\forall k > n_1$ $P(i) = \text{True}$ $\forall i \in [n_0, k-1]$ ¿ $P(k)$?

$\cdot\ T(k) = 2T\lfloor \frac{k}{2} \rfloor + 1 \leq 2C\lfloor \frac{k}{2} \rfloor + 1 \leq 2 \cdot C \cdot \frac{k}{2} + 1 = Ck + 1$

$\boxed{T(k) \leq Ck + 1} \longrightarrow T(k) \leq Ck.$

$Ck \leq Ck + 1$

$P\lfloor \frac{k}{2} \rfloor \longrightarrow P(k)$

$2D + 1 \leq D$ $\qquad T(k) \leq Ck + D$.

$D + 1 \leq 0$

$\boxed{D \leq -1}$

$T(k) \leq 2T\lfloor \frac{k}{2} \rfloor + 1 \leq 2(C\lfloor \frac{k}{2} \rfloor + D) + 1 \leq Ck + 2D + 1$

$T(k) \leq Ck + 2D + 1 = Ck - 1 = Ck + D.$ $\qquad \leq Ck + D$

$\therefore\ T(n) \leq 2n - 1 \ \forall n \geq 1$

En particular $T(n) \leq 2n - 1 \leq 2n \ \forall n \geq 1$ $\quad \therefore\ T(n) = O(n)$

θ

$$T(n) = \underline{a}\, T\left(\frac{n}{b}\right) + \Theta\left(n^{k}\right)$$

$\boxed{\alpha = \log_b a}$ vs $\beta = k$

$\alpha == \beta$?

$T(n) = \Theta(n^{\alpha} \lg n)$ si $\alpha == \beta$

$T(n) = \Theta(n^{\max(\alpha,\beta)})$ $\alpha \neq \beta$

$$T(n) = 2\,T(\lfloor n/2 \rfloor) + \underbrace{cn + d}_{\Theta(n^1)}$$

$\alpha = \log_2 2 = 1$

$\beta = 1$

$\Theta(n^1 \cdot \lg n)$

Exercise 2 (4 points). Resolver la relación de recurrencia **explicitamente**, usando los métodos vistos en clase. Deberás brindar la **solución exacta** para infinitos valores de n. Usar esta información para encontrar y probar el crecimiento asintótico de $T(n)$.

$\boxed{\log_{45} 2025}$ 2

$$T(n) = \begin{cases} n^2 & 1 \leq n < 45 \\ 2025T(\lfloor n/45 \rfloor) + n^{3} - 2024 & \text{otherwise} \end{cases}$$

$\beta = 3$

Puedes asumir que la función $T(n)$ es creciente.

$T(n) = \Theta(n^{3} \lg n)$

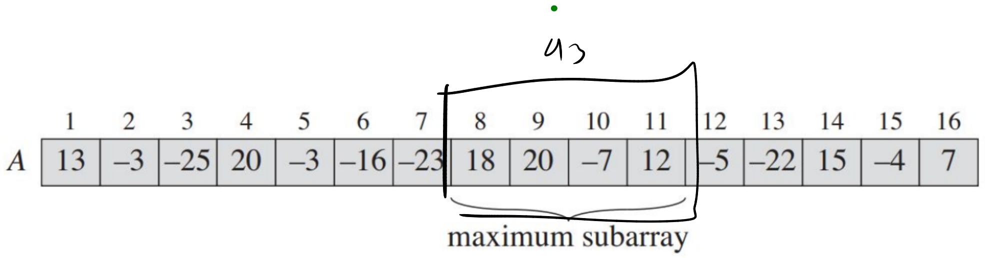


Figure 1: Cormen, Introduction to Algorithms

$$val(A[i;j]) = \sum_{k=i}^{j} A(k)$$

Dado A:

$$S = \max_{\emptyset \le i < j \le n} \{ val(A[i,j]) \}$$

(Asumiendo que consideramos al $\emptyset$)

Algo 1 (A):

```
maxi = 0
for i = 1 To n:
    for j = i To n:
        // A[i,j]
        sum = 0
        for k = i To j:
            sum = sum + A(k)
        // val(A_ij)
        maxi = max(maxi, sum)
return maxi.
```

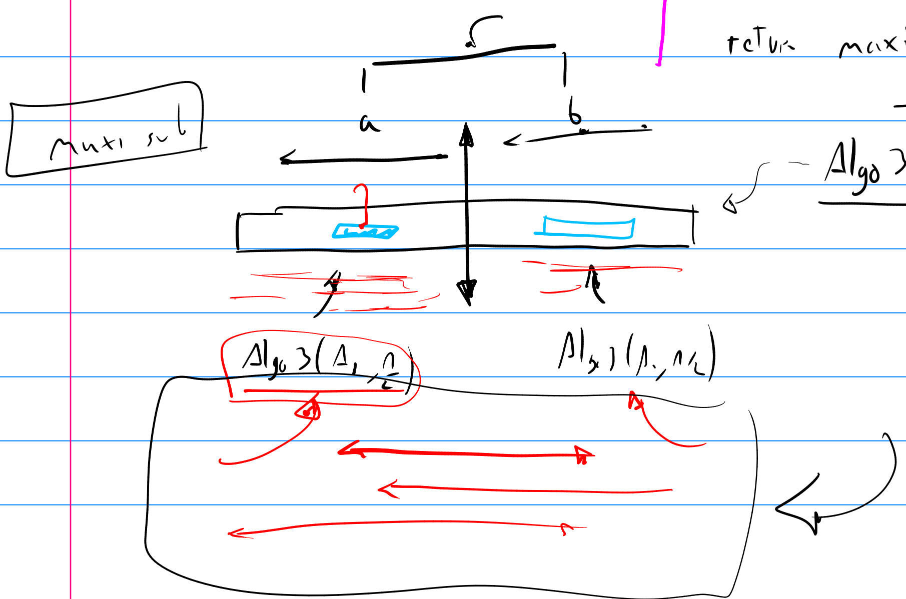
Algo 2 (A):

```
acum[0:n]    // acum[i] = sum_{k=1}^{i} A[k]
acum[0] = 0
For i = 1 To n:
    acum[i] = A(i) + acum(i-1)
// - - - # OBTENIENDO MAX - - -
maxi = 0
For i = 1 To n:
    for j = i To n:
        sum = acum(j) - acum(i-1)
        maxi = max(maxi, sum)
return maxi.
```

$\Theta(n)$

$\Theta(n^2)$

Algo 3 (A, n). max val subarray de A. T(n)



Algo 3 (A, i, j) // max val de un sub array de A[i:j]

CB → $\cdot$ if(i==j) return A[i] // si caso ∅ max(0, A_i)

D → mid = $\lfloor (i+j)/2 \rfloor$

C { T_1 = Algo3(A, i, mid)
  T_2 = Algo3(A, mid+1, j)

maxi = 0

for a = i To mid:

  for b = mid+1 To j:

    maxi = max(maxi, acum(b) - acum(a-1))

maxi = max(maxi, T_1, T_2)

return maxi

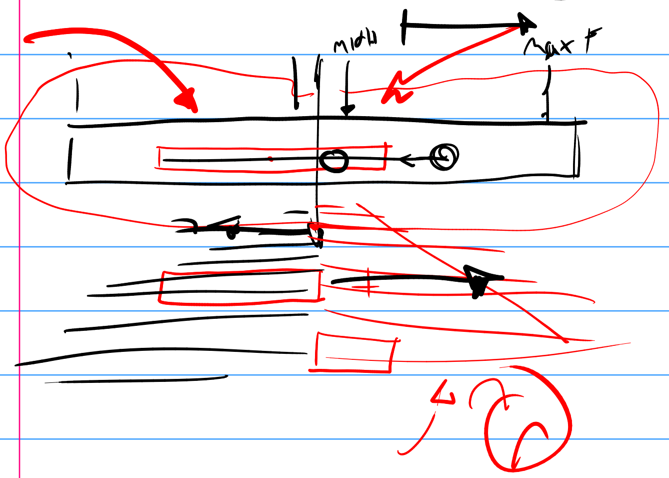
$$F(n) = 2F(\lfloor n/2 \rfloor) + \boxed{\Theta(n^2)}$$

$\alpha = \log_2 2 = 1$

$p(2)$

$\Theta(n^2)$

(Sn) → S_1 S_2 S_3 S_4



Algo (A, i, j)

  if(i==j) return A[i]

  $\cdot$ mid = $\lfloor (i+j)/2 \rfloor$

  T_1 = Algo(A, i, mid)

  T_2 = Algo (A, mid+1, j)

  T_3 = Cruzado(A, i, j, mid)

  return max(T_1, T_2, T_3)

$$T(n) = 2T(\lfloor n/2 \rfloor) + \Theta(n)$$

$\Theta(n \lg n)$,

Cruzado (A, i, j, mid)

  sumr = maxr = A[mid+1]

  for k = mid+2 To j:

    sumr = A[k] + sumr

    maxr = max(maxr, sumr)

  suml = maxl = A[mid]

  for k = mid-1 downTo i:

    suml = A[k] + suml

    maxl = max(maxl, suml)

return maxr + maxl

cnt 36

Algo(A, B, n) <- el resultado de $A \cdot B$

$n=6$

$(a_1 \times 10^k + a_2)$

$(b_1 \times 10^k + b_2)$

p

q

r

s

13	24
97	68

135	246
973	412

4 subprobl + D + C

$$T(n) = \underline{4}\,T(n/2) + \Theta(n) = \Theta(n^2)$$

$2^{\log_2 4}$

$\log_2(4)$

$p = a_2 b_2$

$q = a_1 b_2$

$r = a_2 b_1$

$s = a_1 b_1$

$$\boxed{p+q+r+s} = a_2 b_2 + a_1 b_2 + a_2 b_1 + a_1 b_1$$
$$= (a_1+a_2)(b_1+b_2)$$

$$q + r = (a_1+a_2)(b_1+b_2) - p - s$$

$$T(n) = 3T(n/2) + \Theta(n) \leftarrow \Theta(n^{\lg 3})$$

$$T(n) = 8T\left(\frac{n}{2}\right) + \Theta(n^2) = \Theta(n^3)$$

$2 = \lg_2^{8}$ vs 2

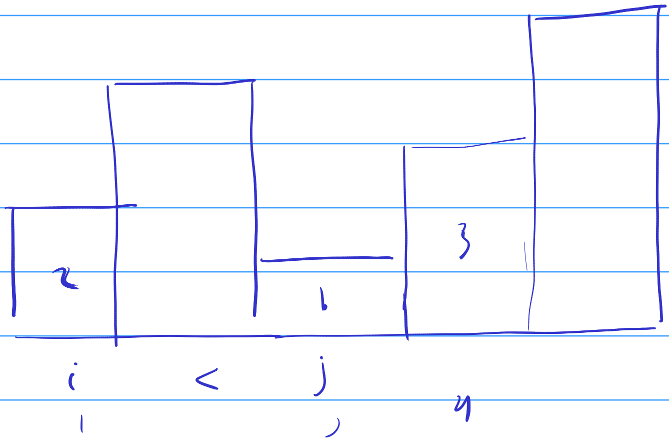
1 2 3 4

in ord ✓

1 2 4 3
2 1 3 4
1 3 2 4

4 3 2 1

2 4 1 3 5



(1, 3)
(2, 1)
(2, 4)

$\# I(A_{ord}) = 0$

$\# I = \binom{n}{2}$

n^2