CÓDIGO

2 0 2 0 1 0 3 1 1

NOMBRE DEL ALUMNO

~~[illegible]~~ MATOS — APELLIDO PATERNO

CANGALAYA — APELLIDO MATERNO

JEREMY JEFFREY — NOMBRES

CARRERA: CS

SECCIÓN: 1

FIRMA

PRÁCTICA CALIFICADA DEL CURSO:

Nº 1

NOTA

16

NOMBRE DEL PROFESOR

Juan Contreras

FIRMA

IMPORTANTE

- El alumno deberá completar los datos indicados en la carátula antes de iniciar la práctica.
- La práctica deberá ser desarrollada únicamente con lapicero negro o azul.
- El trabajo en limpio se desarrollará en la cara derecha de cada hoja, el lado izquierdo podrá ser usado como borrador.
- Se considerará importante para la calificación el orden, la ortografía y limpieza de la práctica.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

1-2

$$T(n) = \begin{cases} 1 \\ 4T(\lfloor n/4 \rfloor) + n^2 \end{cases}$$

$n = 4^K$

$n = 4^i$

$\lg_4 n = i$

$$T(n) = 4T(n/4) + n^2$$
$$= 4\left(4T(n/4^2) + \left(\frac{n}{4}\right)^2\right) + n^2$$
$$= 4^2 T(n/4^2) + \frac{n^2}{4} + n^2$$
$$= 4^2\left(4T(n/4^3) + \left(\frac{n}{4^2}\right)^2\right) + \frac{n^2}{4} + n^2$$
$$= 4^3 T(n/4^3) + \frac{n^2}{4^2} + \frac{n^2}{4^1} + \frac{n^2}{4^0}$$
$$\vdots$$
$$= 4^i T(n/4^i) + n^2\left(\frac{1}{4^0} + \frac{1}{4^1} + \dots + \frac{1}{4^{i-1}}\right)$$

$i = \lg_4 n$

$$4^{\lg_4 n} T(1) + n^2 \underbrace{\sum_{K=0}^{\lg_4 n - 1} \frac{1}{4^K}}_{\text{geometrica}} = n + n^2\left(\frac{\left(\frac{1}{4}\right)^{\log_4 n} - 1}{1/4 - 1}\right)$$
$$= n + n^2\left(\frac{4-4n}{-3n}\right)$$
$$T(n) = n\left(\frac{4n-1}{3}\right) \blacksquare$$

2

$4^K \leq n < 4^{K+1}$

- $T(4K) = 4^K\left(\frac{4^{K+1}-1}{3}\right) = \frac{4^{2K+1} - 4^K}{3} = \frac{(4^K)^2 \cdot 4 - 4^K}{3} = \frac{4n^2 - n}{3} \geq \frac{4n^2 - n}{4} = n^2 - \underbrace{\frac{n}{4}}_{\leq 1/2 n^2}$

$$T(4^K) \geq n^2 - \frac{n}{4} \geq n^2 \cdot \frac{1}{2} \quad \forall n \geq 1$$

- $T(4^{K+1}) = 4^{K+1}\left(\frac{4^{K+2}-1}{3}\right) \leq 4^{K+1}\frac{(4^{K+2})}{3} \leq \frac{4^{2K+3}}{3} = \frac{(4^K)^2 \cdot 4^3}{3} = \frac{64}{3}n^2 \quad \forall n \geq 1$

0

Entonces:

$$0 \leq \frac{1}{2}n^2 \leq \underbrace{n\left(\frac{4n-1}{3}\right)}_{} \leq \frac{64}{3}n^2 \quad \forall n \geq 1$$

$$T(n) = \Theta(n^2)$$

Probar $T(n) = \Omega(n \lg n)$ $\quad C = \frac{1}{2}$

Caso base: $T(1) = 1 \geq 1 \lg(1) \cdot \frac{1}{2} = 0$

Paso inductivo: Suponga que $T(n/4) \geq \frac{1}{2}\frac{n}{4}\lg\frac{n}{4}$

1,5

Luego: $T(n) = 4T(n/4) + n^2$

$$\geq 4 \cdot \frac{1}{2}\frac{n}{4}\lg(n/4) + n^2 \quad \text{por (H.I)}$$
$$\geq \frac{n}{2}(\lg(n) - \underbrace{\lg(4)}_{2}) + n^2$$
$$\geq \frac{n}{2}\lg(n) \underbrace{- n + n^2}_{\geq 0 \ \forall n \geq 1} \geq \frac{n}{2}\lg(n) \geq 0 \quad \forall n \geq 1$$

$\therefore \quad 0 \leq \frac{1}{2} n \lg(n) \leq T(n) \quad \forall n \geq 1 \quad T(n) = \Omega(n \lg n) \blacksquare$

E-6

- Koratsula sean 2 números X, Y

$$\left.\begin{array}{l} X = X_l \cdot 10^{n/2} + X_r \\ Y = Y_l \cdot 10^{n/2} + Y_r \end{array}\right\} XY = \underbrace{(X_l \cdot Y_l)}_{1} 10^{n} + (\underbrace{X_l Y_r}_{2} + \underbrace{Y_l X_r}_{3}) 10^{n/2} + \underbrace{X_r Y_r}_{4}$$

Este algoritmo haría 4 llamadas recursivas

```
Koratsula (X, Y)
    p = Koratsula(Xl, Yl)
    q = Koratsula(Xl, Yr)
    r = Koratsula(Xr, Yl)
    s = Koratsula(Xr, Yr)
    return (p)10^n + (q+r) 10^(n/2) + s
```

$T(n) = 4T(n/2)$

Sin embargo:

$$(X_l + X_r)(Y_l + Y_r) = \underbrace{X_l Y_l + X_r Y_r}_{\text{son 2 conocidas}} + X_l Y_r + X_r Y_l$$

$$X_l Y_r + X_r Y_l = \underbrace{(X_l + X_r)(Y_l + Y_r)}_{1} - \underbrace{X_l Y_l}_{2} - \underbrace{X_r Y_r}_{3}$$

Lo anterior concluye que son necesarias solo 3 llamadas.

Resultando en:

```
Koratsula (X, Y)
    p = Koratsula(Xl, Yl)
    q = Koratsula(Xr, Yr)
    s = Koratsula(Xl + Xr, Yl + Yr)
    return p 10^n + (s - q - p) 10^(n/2) + s
```

$T(n) = 3T(n/2) + n^2$

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

E-1

$$n^2 - 100n - 200\sqrt{n} + 50\sqrt[3]{n} = \Theta(n^2)$$

Note que para $n \geq \max\{400, \sqrt[3]{800^2}\}$ se tiene que

con $n \geq 400$:

$$n \geq 400$$
$$n^2 \geq 400n$$
$$\frac{n^2}{4} \geq 100n$$

con $n \geq \sqrt[3]{800^2}$:

$$\sqrt[3]{800^2} \leq n$$
$$800^2 \leq n^3$$
$$800^2 n \leq n^4$$
$$n \leq \frac{n^4}{800^2}$$
$$\sqrt{n} \leq \frac{n^2}{800}$$
$$800\sqrt{n} \leq n^2$$
$$200\sqrt{n} \leq n^2/4$$

3

Luego

$$n^2 - 100n - 200n + 50\sqrt[3]{n} \geq n^2 - 100n - 200n \geq n^2 - 2\frac{n^2}{4} = \frac{1}{2}n^2$$

Con $n \geq \sqrt[5]{50^3}$:

$$\sqrt[5]{50^3} \leq n$$
$$50^3 \leq n^5$$
$$n \cdot 50^3 \leq n^6$$
$$n \leq \frac{n^6}{50^3}$$
$$\sqrt[3]{n} \leq \frac{n^2}{50}$$
$$50\sqrt[3]{n} \leq n^2$$

Luego

$$n^2 - 100n - 200n + 50\sqrt[3]{n} < n^2 + 50\sqrt[3]{n} \leq n^2 + n^2 = 2n^2$$

Entonces:

$$0 \leq \frac{1}{2}n^2 \leq n^2 - 100n - 200\sqrt{n} + 50\sqrt[3]{n} \leq 2n^2 \quad \forall n \geq \max\{400, \sqrt[3]{800^2}, \sqrt[5]{50^3}\}$$

$\blacksquare$

E-4

$\lg(n) - \lg(i)$ → profundidad del nodo

↓

altura del arbol = profundidad

¿Cual es el parametro de la inducción, n o i?

Caso base: 1 nodo $\quad \lg(1/1) = 0$

Paso inductivo: Suponga que altura del nodo $n/2$ (ultimo con hijos) es

$$\lg(n/n_{/2}) = \lg(2n/n) = \lg(2) = 1$$

Luego si tomamos un hijo del nodo $n/2$ (ultimo con hijos) este sería una hoja con altura 0

altura del padre -1 = altura del hijo $\blacksquare$

0

E-3

max-subsegment (A, l, r) basado en el max subarray problem

if l<r: mid = l + (r-l)/2 $\Theta(1)$

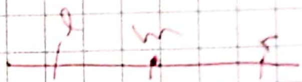
S₁ = max-subsegment(l, mid) $T(n/2)$

S₂ = max-subsegment(mid+1, r) $T(n/2)$

S₃ = max-crossing-subsegment (A, l, mid, r) $\Theta(n)$

return max (S₁, S₂, S₃) $\Theta(1)$

```
max-crossing subsegment (A, l, mid, r)
    i = mid
    while i > l and A[i-1] ≤ A[i]:
        i--
    j = mid
    while j < r and A[j] < A[j+1]:
        j++
    return j - i + 1
```



Recurrencia : $2T(n/2) + \Theta(n) = \Theta(n \lg n)$

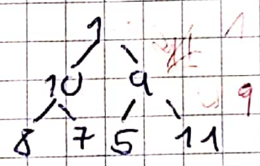
$\lg 2 / \lg 2 = 1$ teorema maestro

4

E-5

• No, porque hacer intercambios hacia arriba no garantiza que los subárboles se mantengan como máx heap.

Contraejemplo : correr en heap



• Cambio

línea 1: ir de j ← 2 up to n   ya que después de cada iteración garantiza que lo de arriba es un max-heap

```
Algo (A)
    for j ← 2 to n
        i ← j
        while i ≥ 2 and A[⌊i/2⌋] < A[i]
            A[⌊i/2⌋] ↔ A[i]
            i ← ⌊i/2⌋
```

3