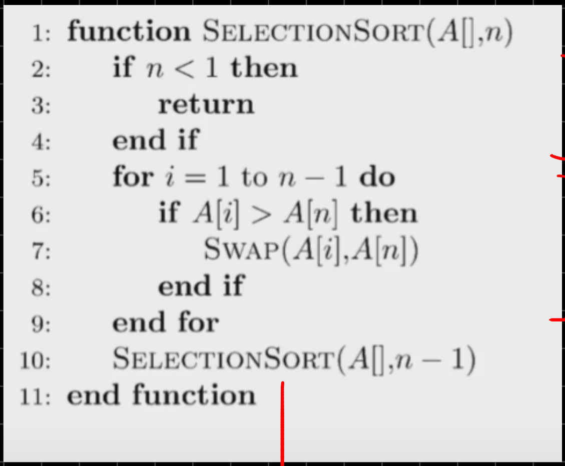


```
function SELECTIONSORT(A[],n)
    if $n < 1$ then
        return
    end if
    for $i = 1$ to $n - 1$ do
        if $A[i] > A[n]$ then
            SWAP(A[i],A[n])
        end if
    end for
    SELECTIONSORT(A[],n − 1)
end function
```

Caso Base

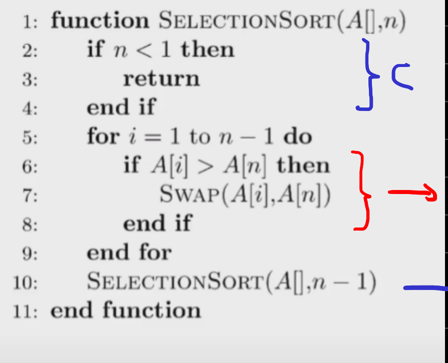
Comparamos el elemento i-esimo con el ultimo elemento y si el elemento i-esimo es más grande que el ultimo los cambiamos

Estos intercambios se hacen continuamente y terminado se tendra el maximo elemento del array en la posición final

En cada recursión estaremos encontrando el máx elemento de ese array que cada vez se esta acortando

Vamos a Analizarlo:

↳ $T(n) \rightarrow$ tiempo de ejecución del Selection Sort en un array de tamaño "n"



```
function SELECTIONSORT(A[],n)
    if $n < 1$ then
        return
    end if
    for $i = 1$ to $n - 1$ do
        if $A[i] > A[n]$ then
            SWAP(A[i],A[n])
        end if
    end for
    SELECTIONSORT(A[],n − 1)
end function
```

} C

→ Constante C

$C\sum_{i=1}^{n-1} 1 = C(n-1)$

→ $T(n-1)$

$T(n) = C(n-1) + T(n-1)$

Caso Base: $\boxed{T(0) = C}$

$T(n) = C(n-1) + T(n-1)$

Recorriendo la recursión

$T(n-1) = C(n-1-1) + T(n-1-1)$

• $T(n-1) = C(n-2) + T(n-2)$

$T(n) = C(n-1) + [C(n-2) + T(n-2)]$

$T(n-2) = C(n-3) + T(n-3)$

$T(n) = C(n-1) + C(n-2) + [C(n-3) + T(n-3)]$

$T(n) = C(n-1) + C(n-2) + C(n-3) + T(n-3)$

↳ $T(n) = C(n-1) + C(n-2) + \dots + C(n-k) + T(n-k)$ } Para algún $k \geq 1$

Obs:

Caso Base $T(0) = C$

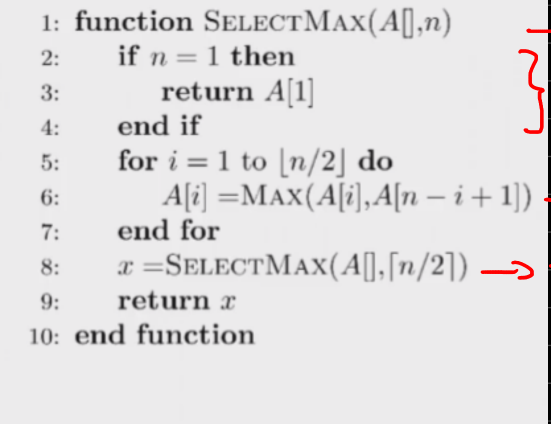
Eligiremos un k tq $n-k=0$

$\boxed{n=k}$

$T(n) = \underbrace{\overset{n-1}{C(n-1)} + \overset{n-2}{C(n-2)} + \dots + \overset{..1}{} \overset{0}{C(n-n)}}_{n \text{ terms}} + \cancel{T(0)} \; C$

↳ $T(n) = C[0 + 1 + \dots + n-2 + n-1]$ → Suma aritmetica

$T(n) = \frac{(n-1)(n)}{2}C + C \Rightarrow T(n) \in \Theta(n^2)$ //



```
function SelectMax(A[], $n$)
    if $n = 1$ then
        return $A[1]$
    end if
    for $i = 1$ to $\lfloor n/2 \rfloor$ do
        $A[i]$ = Max($A[i]$, $A[n-i+1]$)
    end for
    $x$ = SelectMax(A[], $\lceil n/2 \rceil$)
    return $x$
end function
```

→ Función Recursiva, para obtener el máx elemento de un array

→ Caso Base $T(1) = c$

$\rightarrow c \} \; c\left(\frac{n}{2}\right)$

$\rightarrow T\left(\frac{n}{2}\right)$

$$T\left(\frac{n}{2}\right) = \frac{cn}{4} + T\left(\frac{n}{4}\right)$$

$$T(n) = \frac{cn}{2} + T\left(\frac{n}{2}\right)$$

$$T(n) = \frac{cn}{2} + \frac{cn}{4} + T\left(\frac{n}{4}\right)$$

$$T(n) = \underbrace{\frac{cn}{2} + \frac{cn}{4} + \frac{cn}{8}} + T\left(\frac{n}{8}\right)$$

$$\frac{1}{2}\sum_{i=0}^{k-1} \frac{cn}{2^i} + T\left(\frac{n}{2^k}\right)$$

Queremos que

$$\frac{n}{2^k} = 1$$

$$n = 2^k$$

$$\boxed{k = \lg n}$$

$$\frac{cn}{2}\sum_{i=0}^{k-1} \frac{1}{2^i} + \cancel{T}(1) \quad c$$

$$T(n) = \underbrace{\frac{cn}{2}\sum_{i=0}^{k-1} \left(\frac{1}{2}\right)^i} + c$$

acotaremo esto a una serie geometrica

$$\sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i = \frac{1}{1-1/2}$$

$$T(n) \leq cn\underbrace{\sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i} + c$$

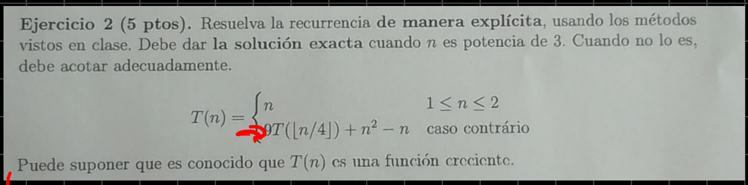
$$cn\left(\frac{1}{1-1/2}\right) + c$$

$$T(n) \leq 2cn + c \;\} \; T(n) \in O(n)$$

Inferior

$$cn(1)+c \leq Tn \;\} \; T(n) \in \Omega(n)$$

$$\therefore T(n) \in \Theta(n) \; \square$$



Ejercicio 2 (5 ptos). Resuelva la recurrencia **de manera explícita**, usando los métodos vistos en clase. Debe dar **la solución exacta** cuando n es potencia de 3. Cuando no lo es, debe acotar adecuadamente.

$$T(n) = \begin{cases} n & 1 \leq n \leq 2 \\ 9T(\lfloor n/4 \rfloor) + n^2 - n & \text{caso contrário} \end{cases}$$

Puede suponer que es conocido que $T(n)$ es una función creciente.

$\to n = 2^k$ $\quad \left\lfloor \frac{2^k}{4} \right\rfloor = \frac{2^k}{4} = 2^{k-2}$

$9T\left(\frac{2^k}{4}\right.$

$1°\quad 9T(2^{k-2}) + 2^{2k} - 2^k$ $\qquad \cdot\ T(2^{k-2}) = 9T(2^{k-4}) + 2^{2k-4} - 2^{k-2}$

$2°\quad 9\left[9T(2^{k-4}) + 2^{2k-4} - 2^{k-2}\right] + 2^{2k} - 2^k$ $\qquad k$

$2°\quad 81T(2^{k-4}) + 9 \cdot 2^{2k-4} + 2^{2k} - 9 \cdot 2^{k-2} - 2^k$

$\cdot\ T(2^{k-4}) = 9T(2^{k-6}) + 2^{2k-8} - 2^{k-4}$

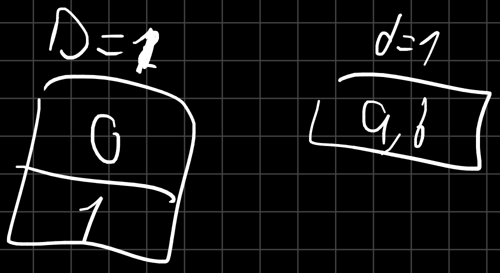
$\leq$

$3°\quad 9^3 T(2^{k-6}) + 9^2 \cdot 2^{2k-8} + 9\, 2^{2k-4} + \boxed{2^{2k}} - 9^2\, 2^{k-4} - 9\, 2^{k-2} - 2^k$

$i°\quad 9^i\, T(2^{k-2i}) + \sum_{i=1}^{k/2} 9^{i-1} \cdot 2^{2k-4(i-1)} - \sum_{i=1}^{k/2} 9^{i-1}\, 2^{k-2(i-1)}$

2^i

2^{k-2i} $\quad 2^{k-2i}$ $\quad k = 2i$ $\quad i = k/2$ $\quad i = 1$



$$F(n)\begin{cases} 1 & n=1 \\ 2F(n-1)+1 & c.c. \end{cases}$$

$F(1)=1$

$F(2)=3$

$F(3)=7$

$F(4)=15$

i	1	2	3	4	....	i	....	n
$F(i)$	1	3	7	15	$2F(i-1)+1$			$2F(n-1)+1$

- 1 Expandir la recurrencia
- 2 Llegar al caso Inicial

$$F(n) = 2F(n-1)+1 \qquad 1$$
$$= 2\left(2F(n-2)+1\right)+1$$
$$= 2^2F(n-2)+2+1 \qquad 2$$
$$= 2^2\left(2F(n-3)+1\right)+2+1$$
$$= 2^3F(n-3)+4+2+1 \qquad 3$$

$\underline{n=k}$ $\cdot 2^kF(n-k)+2^{k-1}+\dots 2^0$

$\underline{k=n-1}$ $= 2^{n-1}F(1)+2^{n-2}+2^{n-3}+\dots+2^0$

$$= 2^{n-1}\cdot 1+\sum_{i=0}^{n-2}2^i \quad \{ \quad 2^{n-1}-1$$

$$= 2^{n-1}+2^{n-1}-2 \implies 2.2^{n-1}-1 \implies \boxed{2^n-1}$$

$$\sum_{i=n}^{b} n_i - n_{i+1} = \cancel{n_a} - \cancel{n} - /$$

$$F(k) = 2F(k-1) + 1$$

$$F(k) - 2F(k-1) = 1$$

$$F(n) - 2\cancel{F(n-1)} = 1$$

$$\left(\cancel{F(n-1)} - 2\cancel{F(n-2)} = 1\right) \times 2 = 2^1$$

$$\left(\cancel{F(n-2)} - 2\cancel{F(n-3)} = 1\right) \times 4 = 2^2$$

$$\cancel{F(n-3)} - 2F(n-4) = 1 \;) \times 8 = 2^3$$

$$\vdots \qquad \times 16 = 2^4$$

$$\underset{k}{F(2)} - 2F(1) = 1$$

$$\left[F(k) - 2F(k-1) = 1\right] \times 2^{n-k}$$

$$\sum_{k=2}^{n} a_k - a_{k-1} = \sum_{k=2}^{n} 2^{n-k}$$

$$\boxed{a_k = 2^{n-k} F(k)}$$

telescopica

serie geometrica

$$a_n - a_1 = 2^{n-1} - 1$$

$$2^{n-n} F(n) - 2^{n-1} F(1) = 2^{n-1} - 1$$

$$F(n) = 2^n - 1 \quad //$$

2° Forma

- Formula Explicita (Expandir)

1° Forma

Tantear la formula y hacer Inducción

Define $G(k) = F(k)+1$

$$\to G(k) = F(k)+1 = \underbrace{2F(k-1)+1}+1 = 2\left(F(k-1)+1\right)$$

$$= 2G(k-1) \quad \forall k \geq 1$$

$$\wedge\ G(1) = F(1)+1 = 2 \quad \wedge\ G(k) = 2G(k-1)$$

$$\left.\begin{aligned} G(n) &= 2G(n-1) \\ G(n-1) &= 2G(n-2) \\ G(n-2) &= 2G(n-3) \\ &\vdots \\ G(2) &= 2G(1) \end{aligned}\right\} \begin{matrix} n-1 \\ \text{veces} \end{matrix} \longrightarrow\ \hookrightarrow G(n) = 2^{n-1}G(1) = 2^{n}$$

$$G(n) = 2.2.2.\ldots G(1)$$

$$F(n) = \begin{cases} 1 & : n=1 \\ 2F\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n & : \text{otros casos} \end{cases}$$

- 4°

3° Acotar $F(n)$ en general

$F(2) = 2F(1) + 2 = 4$

$F(3) = 2F(1) + 3 = 5$

2° Comentar que $f(n)$ es creciente

Restringiremos que n son potencias de 2 y lo que haremos es acotar para todos los valores de n

Calculando $f(n)$ para potencias de 2

$n = 2^k \rightarrow F(2^k) = 2F(2^{k-1}) + 2^k \quad \forall\ k \geq 1$

$F(2^0) = 1$ → caso base

telescopica

$k=$

$8T(1)+ \qquad 8+8 \qquad \boxed{15}$

F creciente: $x \le y \longrightarrow f(x) \le f(y)$

En general

cualquier
valor
estara
entre 2 potencias de 2 $\Rightarrow 2^{\alpha} \le n < 2^{\alpha+1}$

↳ $\alpha \le \lg(n) < \alpha + 1$

$$f(2^{\alpha}) \le f(n) < f(2^{\alpha+1})$$

del
ejercicio
anterior

$$\underline{\alpha . 2^{\alpha} + 2^{\alpha}} \le f(n) < (\alpha+1)2^{\alpha+1} + 2^{\alpha+1}$$

$$Cn\lg n < \alpha 2^{\alpha} + 2^{\alpha}$$

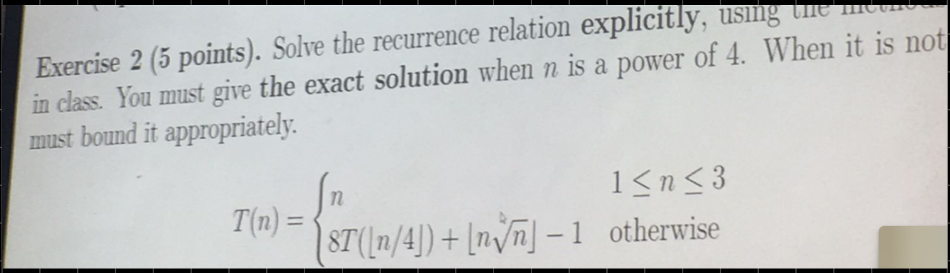
Sabemos que,

$$(\alpha + 1 > \lg n) \times 2^{\alpha}$$

$$(\alpha+1)2^{\alpha} > \underline{\lg n \, 2^{\alpha}} > \boxed{\lg n \left(\frac{n}{2}\right)}$$

$$(\alpha+2)\, 2^{\alpha+1} < (\lg n + 2)\underbrace{(2^{\alpha})}_{\le n} . 2 < (\underline{\lg n + 2})\, n . 2$$

$$\le \boxed{3\lg n} . n . 2$$



Exercise 2 (5 points). Solve the recurrence relation **explicitly**, using the metho
in class. You must give **the exact solution** when n is a power of 4. When it is not
must bound it appropriately.

$$T(n) = \begin{cases} n & 1 \leq n \leq 3 \\ 8T(\lfloor n/4 \rfloor) + \lfloor n\sqrt{n} \rfloor - 1 & \text{otherwise} \end{cases}$$

(2^2) $\quad 4^{\frac{3m}{2}} = (4^{1/2})^{3m}$

$\sqrt{n} = \sqrt{4^m} = 4^{\frac{m}{2}} \cdot 4^m$

Para potencias de 4 $\qquad 4^{\frac{3m}{2}} = 8^m$

$n = 4^m \qquad m \geq 1$

$T(n) = 8T(\lfloor n/4 \rfloor) + \lfloor n\sqrt{n} \rfloor - 1$

$\left\lfloor \frac{n}{4} \right\rfloor = \left\lfloor \frac{4^m}{4} \right\rfloor = \underline{4^{m-1}}$

$T(4^m) = 8T(4^{m-1}) + 2^{3m} - 1$

$\lfloor n\sqrt{n} \rfloor = 2^{3m}$

$n = 4^m$

$m = \frac{\lg(n)}{2}$

$T(4) = 15$

$k=m \quad \left(T(4^m) - 8T(4^{m-1}) = 2^{3m} - 1\right) \times 8^0$

$k=m-1 \quad \left(T(4^{m-1}) - 8T(4^{m-2}) = 2^{3m-3} - 1\right) \times 8^1$

$\left(T(4^{m-2}) - 8T(4^{m-3}) = 2^{3m-6} - 1\right) \times 64 = 8^2$

8^m

8^{m-1}

$\vdots$

$k=1 \quad \left(T(4^1) - 8T(4^0) = 2^{3(1)} - 1\right) \times 8^{m-1}$

$\left(T(4^k) - 8T(4^{k-1}) = 2^{3k} - 1\right) \times 2$

$k=m$ $\left(T(4^m) - 8\cancel{T(4^{m-1})} = 8^m - 1\right) \times 8^0$

$8^m - 8^0$

$k=m-1$ $\left(\cancel{T(4^{m-1})} - 8\cancel{T(4^{m-2})} = 8^{m-1} - 1\right) \times 8^1$

$8^m - 8^1$

$\left(\cancel{T(4^{m-2})} - 8\cancel{T(4^{m-3})} = 8^{m-2} - 1\right) \times 64 = 8^2$

$\vdots$

$\times 8^{m-k}$

$k=1$ $\left(\cancel{T(4^1)} - 8T(4^0) = 8^{(1)} - 1\right) \times 8^{m-1}$

$8^{m-k} \times \left(T(4^k) - 8T(4^{k-1}) = 8^k - 1\right)$ $k \geq 1$

$$8^{m-k}T(4^k) - 8^{m-k+1}T(4^{k-1}) = 8^m - 8^{m-k} \quad \Big\} \; k \geq 1$$

$$a_k - a_{k-1} = 8^m - 8^{m-k}$$

$$\boxed{a_k = 8^{m-k}T(4^k)}$$

$$\sum_{k=1}^{m} a_k - a_{k-1} = \sum_{k=1}^{m} (8^m - 8^{m-k})$$

a_m a_0

$$m8^m - \sum_{k=1}^{m} 8^{m-k}$$

$$a_m - a_0 = m8^m - \boxed{\sum_{k=0}^{m-1} 8^k} \rightarrow \frac{8^{m-1+1} - 1}{7}$$

$$T(4^m) - 8^m\underline{T(1)} = m8^m - \frac{8^m - 1}{7}.$$

$$T(4^m) - 8^m = m8^m - \frac{8^m - 1}{7}$$

$$m = \frac{\lg(n)}{2}$$

$$T(4^m) = 8^m + m8^m - \frac{8^m - 1}{7}$$

$$T(n) = n^{3/2} + \frac{\lg(n)n^{3/2}}{2} - \frac{n^{3/2} - 1}{7}$$

$$8^{\frac{\lg n}{2}} = \left(n^{\lg 8}\right)^{1/2}$$

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

Probar que $T(n) = O(n^2)$

$\exists\ c, n_0 > 0$ tq $\forall\ n \geq n_0$

$n \geq n_0 \longrightarrow T(n) \leq \_\_\ n^2$

Inducción Debil

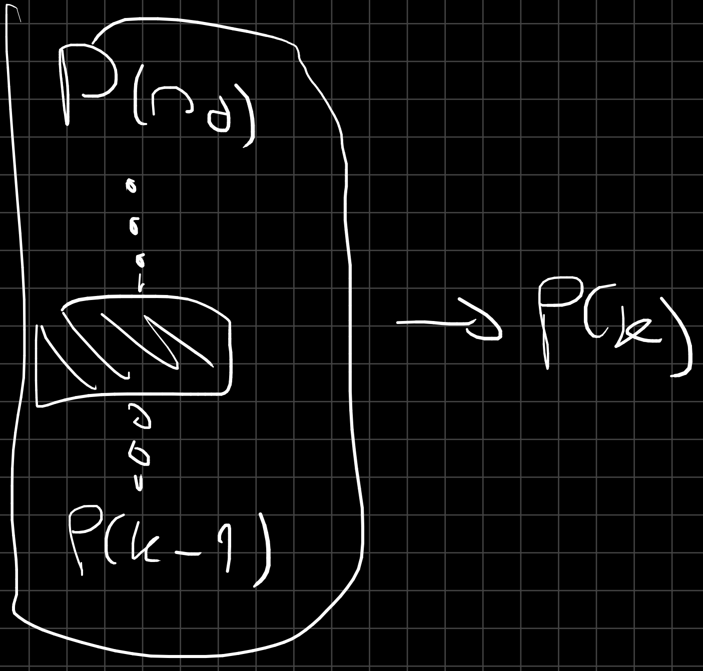
C.B. $\boxed{n=1}$

$P(k-1) \rightarrow P(k) \quad \forall\, k \notin$ C.B.

Inducción Fuerte

C.C.B.B $\boxed{1, 2, \ldots}$

No miras al inmediato anterior



Probar que

$$T(n) \le cn^2$$

$$\updownarrow$$

$$2\underline{T\left(\left\lfloor \frac{n}{2} \right\rfloor\right)} + n \le \underline{cn^2}$$

Probar:

$$T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \le c\left(\left\lfloor \frac{n}{2} \right\rfloor\right)^2$$

$$P(n) \xleftarrow[\text{Implica}]{} P\left(\left\lfloor \frac{n}{2} \right\rfloor\right)$$

Obs:

$x \le y \wedge x \le z$

$y \le z$

$$2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n \le \underline{2c\left(\left\lfloor \frac{n}{2} \right\rfloor\right)^2 + n}$$

$$2c\left(\left\lfloor \frac{n}{2} \right\rfloor\right)^2 + n \le cn^2$$

$$\underbrace{2c\left(\left\lfloor\frac{n}{2}\right\rfloor\right)^2+n} \leq cn^2$$

$$\longrightarrow 2c\left(\left\lfloor\frac{n}{2}\right\rfloor\right)^2+n \leq \frac{cn^2}{2}+n$$

$$\left(\left\lfloor\frac{n}{2}\right\rfloor\right)^2 \leq \left(\frac{n}{2}\right)^2$$

$$\left(\left\lfloor\frac{n}{2}\right\rfloor\right)^2 \leq \frac{n^2}{4} \quad (\times 2c)$$

$$2c\left(\left\lfloor\frac{n}{2}\right\rfloor\right)^2 \leq \frac{cn^2}{2} \quad (+n)$$

$$2c\left(\left\lfloor\frac{n}{2}\right\rfloor\right)^2+n \leq \frac{cn^2}{2}+n$$

Si:

$$\frac{cn^2}{2}+n \leq cn^2$$

Para $c=2$

$$n^2+n \leq 2n^2$$

$\forall n \geq 2$

$$2^{100}+\lg 20w$$

$$c(k-2)(\lg k-2)+k$$

$$c(k\lg k-2k-2\lg k+4)+k$$

$\lg k c$

$$ck\lg k+k+4c+c(-2k-2\lg k)$$

$$c\lg k+k+4c-2ck-2c\lg k$$

$c\lg k + k + 4c - 2ck - 2c\lg k \quad \longrightarrow \lg k < k$

$c\lg k + k + 4c - 4c\boxed{k} >$

$\searrow$

$c\lg k + k + 4c - 2ck - 2c\lg k > c\lg k + \underline{k} + 4c - \underline{4ck}$

$c\lg k + k + k\underline{(1-4c)} + 4c > ck\lg k$

$1 - 4c > 0$

$1 > 4c$

$\boxed{\frac{1}{4} > c} \longrightarrow \boxed{c \leq \frac{1}{8}}$

Teorema Maestro

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

$\log_b^a \quad vs \quad k$

$\log_b^a \neq k \longrightarrow \Theta\left(n^{\max\{\log_a^b,\, k\}}\right)$

$\log_b^a = k \longrightarrow \Theta\left(n^k \lg n\right)$

$$T(n) \begin{cases} n & 1 \le n \le 2 \\ 6T\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + n^2 - n \end{cases}$$

$6+9-3$

$T(3) = 12$

$n = 3^m$

$$T(n) = 6T(3^{m-1}) + 9^m - 3^m$$

$$T(3^m) - 6T(3^{m-1}) = 9^m - 3^m \quad) \times 6^0 \quad k=m$$

$$T(3^{m-1}) - 6T(3^{m-2}) = 9^{m-1} - 3^{m-1} \quad) \times 6^1 \quad k=m-1$$

$$T(3^{m-2}) - 6T(3^{m-3}) = 9^{m-2} - 3^{m-2} \quad) \times 6^2 \quad k=m-2$$

$\vdots$

$) \times 6^{m-k}$

$$T(3^1) - 6T(3^0) = 9 - 3 = 6 \quad) \times 6^{m-1} \quad k=1$$

$$T(3^k) - 6T(3^{k-1}) = 9^k - 3^k \quad) \times \qquad k \geq 1$$

$$6^{m-k} T(3^k) - 6^{m-k+1} T(3^{k-1}) = 6^{m-k}(9^k - 3^k)$$

$$\underbrace{6^{m-k}T(3^k)}_{\downarrow} - 6^{m-k+1}T(3^{k-1}) = 6^{m-k}(9^k-3^k)$$

$$a_k - a_{k-1} = 6^{m-k}(9^k-3^k) \qquad \left\{ \boxed{a_k = 6^{m-k}T(3^k)} \right.$$

$$\sum a^k - \sum a_{k-1} = \sum_{k=1}^{m} 6^{m-k}(9^k-3^k)$$

$$a_m - a_0 = \sum_{k=1}^{m} 6^{m-k}\cdot 9^k - \sum_{k=1}^{m} 6^{m-k} 3^k$$

$$= 6^m \sum_{k=1}^{m} \left(\frac{9}{6}\right)^k - 6^m \sum_{k=1}^{m} \left(\frac{3}{6}\right)^k$$

$\alpha = \frac{9}{6} = \frac{3}{2}$ $\qquad \beta = \frac{3}{6} = \frac{1}{2}$

Serie Geometrica

$$\sum_{k=1}^{m} a^k = \frac{a^{m+1}-a}{a-1}$$

$$a_m - a_0 = 6^m \sum_{k=1}^{m} \alpha^k - 6^m \sum_{k=1}^{m} \beta^k$$

$$= 6^m \left(\frac{\left(\frac{3}{2}\right)^{m+1} - \frac{3}{2}}{\frac{3}{2}-1} - \frac{\left(\frac{1}{2}\right)^{m+1} - \frac{1}{2}}{\frac{1}{2}-1} \right)$$

$$\sum_{k=0}^{m} a^k \hookrightarrow \frac{a^{m+1}-1}{a-1}$$

$=$

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

Probar que $T(n) = \Omega(n \lg n) \rightarrow T(n) \geq C n \lg n$

$$P\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \longrightarrow P(n)$$

$$P(n) \Rightarrow \qquad C n \lg n \leq 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$$

$$P\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \Rightarrow \qquad \left(T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \geq C\left[\left\lfloor \frac{n}{2} \right\rfloor \lg\left(\left\lfloor \frac{n}{2} \right\rfloor\right)\right] \right) \times 2$$

$$\left(2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \geq 2C\left[\left\lfloor \frac{n}{2} \right\rfloor \lg\left(\left\lfloor \frac{n}{2} \right\rfloor\right)\right] \right) + n$$

$$2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n \geq 2C\left[\left\lfloor \frac{n}{2} \right\rfloor \lg\left(\left\lfloor \frac{n}{2} \right\rfloor\right)\right] + n$$

Obs:
Sabemos que
$\lfloor x \rfloor \geq x - 1$

$$2C\left[\left\lfloor \frac{n}{2} \right\rfloor \lg\left(\left\lfloor \frac{n}{2} \right\rfloor\right)\right] + n \geq 2C\left(\frac{n}{2} - 1\right) \lg\left(\frac{n}{2} - 1\right) + n$$

Quiero que esto sea: mayor a $\lg\left(\frac{n}{4}\right)$

↓ más fácil manipul

Si:

$$\cancel{\lg}\left(\frac{n}{2} - 1\right) \geq \cancel{\lg}\left(\frac{n}{4}\right)$$

$$\frac{n}{2} - 1 \geq \frac{n}{4}$$

$$\frac{n-2}{\cancel{2}} \geq \frac{n}{\cancel{4}\,2} \rightarrow \boxed{n \geq 4}$$

$\therefore$ Para $\forall\, n \geq 4$ tenemos

$$2C\left(\frac{n}{2} - 1\right) \lg\left(\frac{n}{2} - 1\right) + n \geq 2C\left(\frac{n}{2} - 1\right) \lg\left(\frac{n}{4}\right) + n$$

$$2c\left(\frac{n}{2}-1\right)\lg\left(\frac{n}{4}\right)+n = \cancel{2}c\left(\frac{n-2}{\cancel{2}}\right)(\lg n-\lg 4)+n$$

$$\Rightarrow c(n-2)(\lg n-2)+n = c(n\lg n-2n-2\lg n+4)+n$$

$$\Rightarrow cn\lg n+4c-2cn-2c\boxed{\lg n}+n \longrightarrow \lg n < n$$

si quitas lgn va ser mayor a si quitas n

Por lo que

$$cn\lg n+4c-2cn-2c\lg n+n > \underbrace{cn\lg n+4c+n-2cn-2cn}$$

$$cn\lg n+4c+n-4cn$$

$$\underline{cn\lg n+4c+n(1-4c)} > cn\lg n$$

Esto sera mayor a cnlgn siempre y cuando $1-4c>0$

$\therefore$ $1-4c>0$

$$\boxed{c > \frac{1}{4}}$$

Para un $\boxed{c=\frac{1}{8}}$

$T(n) = 3n^2 + n - n.(9n)^{\log_3 2}$

$T(3) = 12$

$T(3) = 30 - 24$

$= 6$

$27^{\log_3 2}$

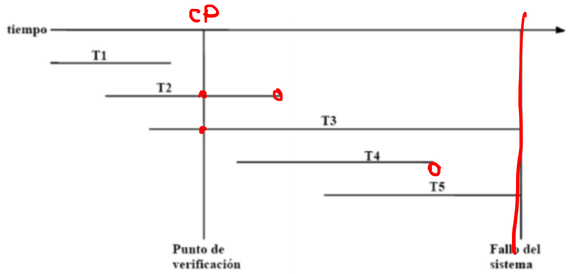
$2^{\log_3 3^3}$

$$6T\left(\left\lfloor \frac{n}{3} \right\rfloor\right) + n^2 - n = T(n)$$

$n = 3^m$

$T(n)$

Considere que el proceso de recuperación de bases de datos utiliza puntos de verificación (checkpoints). Aplique el algoritmo de recuperación e indique cuáles de las siguientes transacciones deben deshacerse (UNDO) y cuáles volver a ejecutarse (REDO) ante una falla del sistema.



1. CP[T2,T3]
2.
3.
4.
5.

1° CP[T2, T3]

2° REDO = [ ]
UNDO = [T2, T3] ← Activas

3°

4° guardamos en el stack, la ultima pos del CP

5° REDO [T2, T4]
UNDO [T3, T5]

Undo (UNDO) Flush T1
redo (REDO)

A

```
<T1, BT>
<T1, A, 10, 15> //Update A con 15.
<T1, B, 20, 15> //Update B con 15.
<T2, BT>
<T2, C, 11, 12> // Update C con 12.
<T1, D, 17, 18> //Update D con 18.
<T2, COMMIT>
<CHECKPOINT>
<T1, COMMIT>
<T3, BT>
<T3, A, 15, 12> //Update A con 12.
<T3, B, 15, 12> //Update B con 12.
<T4, BT>
<T4, C, 12, 22> // Update C con 22.
<T4, COMMIT>
Falla del sistema
```

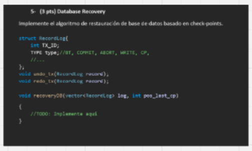
a) Genere un gráfico temporal de ejecución de las transacciones, con referencia al tiempo de checkpoint y al momento de la falla.

UNDO

W W

T1 BT	T1 A→15	T1 B→15	T2 BT	T2 C→12	T1 D→18	T2 Comm	CP	T1 Comm	T3 BT	T3 A→12	T3 B→12	T4 BT	T4 C→22	T4 Comm
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

←



5- (3 pts) Database Recovery

Implemente el algoritmo de restauración de base de datos basado en check-points.

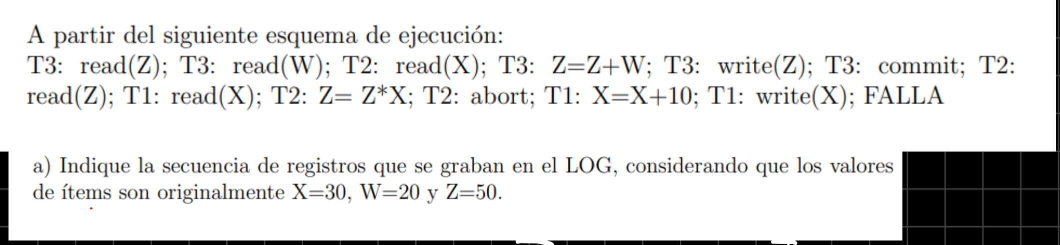
```
struct RecordLog{
    int TX_ID;
    TYPE type;//BT, COMMIT, ABORT, WRITE, CP,
    //...
};
void undo_tx(RecordLog record);
void redo_tx(RecordLog record);

void recoveryDB(vector<RecordLog> log, int pos_last_cp)
{
    //TODO: Implemente aqui
}
```

```
struct RecordLog{
int TxId;
string flag // BT , COMMIT ,
ABORT , WRITE , CP
int IdRes
Tv IA , ID;
vector<RecordLog> activetx;

}
```

→ T1, t2, t3, t4

REDO = [t2, T1, t4]

```
void recoverDB(vector<RecordsLog> log , int pos_last_cp)
{
   set<int> REDO;
   set<int> UNDO;
   for(int i = pos_last_cp ; i < log.size(); i++){
      if (log[i].flag == "COMMIT"){
         REDO.add(log[i].Txid);
      }else{
         if (!REDO.find(log[i].Txid)){
            UNDO.add(log[i].Txid);
         }
      }

   }
}
```



A partir del siguiente esquema de ejecución:
T3: read(Z); T3: read(W); T2: read(X); T3: Z=Z+W; T3: write(Z); T3: commit; T2: read(Z); T1: read(X); T2: Z= Z*X; T2: abort; T1: X=X+10; T1: write(X); FALLA

a) Indique la secuencia de registros que se graban en el LOG, considerando que los valores de ítems son originalmente X=30, W=20 y Z=50.

	T3	T2	T1
	Z X W	Z X W	Z X W
T_3 = Read (Z) →	Z → 50 X W		
T_3 = Read (W) →	Z → 50 X W → 20	Z X W	
T_2 = Read (x) →		Z X → 30 W	
T_3 = Z = Z + W →	Z → Z+W Z → 70		
T_3 = Write (Z) →	Z $\overset{*}{\rightarrow}$ 70		
T_3 = commit →	commit (Z)		
T_2 = read (Z) →		Z → 70 X → 30 W	
T1 = read (x) →			X → 30
T_2 = Z = Z * X →		Z → Z*X Z → 2100	
T_2 = abort →		ABORT	
T_1 = X = X + 10 →			X → X+10 X → 40
T_1 : Write X →			X $\overset{*}{\rightarrow}$ 40

$m = 3$

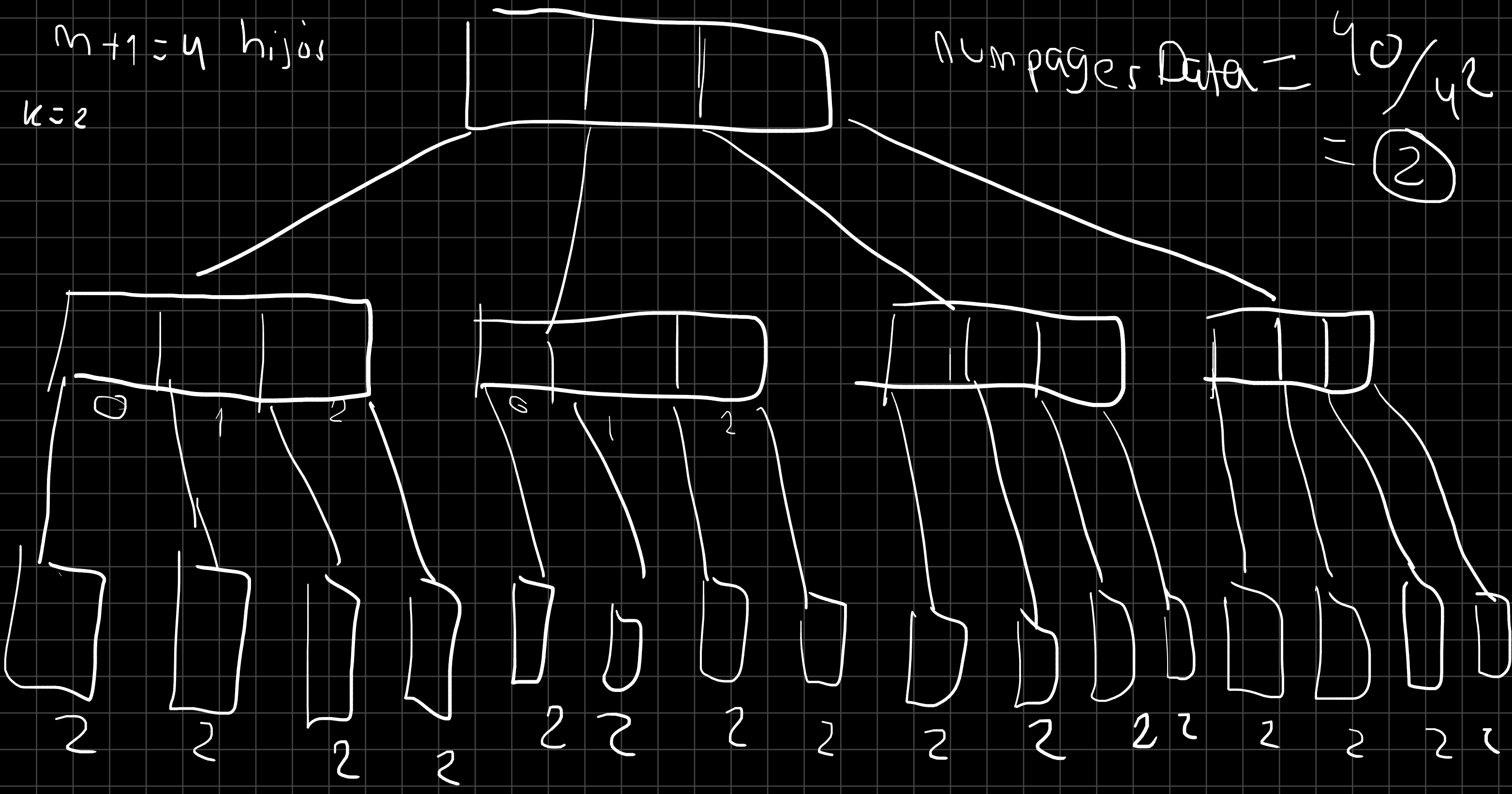
$n = 40$

$m+1 = 4$ hijos

$k = 2$

$\text{numpagesData} = 40/4^2$

$= \boxed{2}$


0
1
2
0
1
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2