

```

1: function SELECTIONSORT(A[], n)
2:   if n < 1 then
3:     return
4:   end if
5:   for i = 1 to n - 1 do
6:     if A[i] > A[n] then
7:       SWAP(A[i], A[n])
8:     end if
9:   end for
10:  SELECTIONSORT(A[], n - 1)
11: end function

```

Caso Base

Comparamos el elemento i-ésimo con el último elemento y si el elemento i-ésimo es más grande que el último los cambiamos

Estos intercambios se hacen continuamente y terminado se tendrá el máximo elemento del array en la posición final

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

Vamos a Analizarlo:

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

```

1: function SELECTIONSORT(A[], n)
2:   if n < 1 then
3:     return
4:   end if
5:   for i = 1 to n - 1 do
6:     if A[i] > A[n] then
7:       SWAP(A[i], A[n])
8:     end if
9:   end for
10:  SELECTIONSORT(A[], n - 1)
11: end function

```

} C

} Constante C

→ $T(n-1)$

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

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

Caso Base: $T(0) = C$

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

Recursiendo la recursión

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

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

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

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

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

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

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

Obs:

Caso Base $T(0) = C$

Elegiremos un k tq $n-k=0$
 $n=k$

$$T(n) = \overset{n-1}{C(n-1)} + \overset{n-2}{C(n-2)} + \dots + \overset{-1}{C(n-1)} + \overset{0}{T(0)}$$

n terms

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

Suma aritmética

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

```

1: function SELECTMAX(A[], n)
2:   if n = 1 then
3:     return A[1]
4:   end if
5:   for i = 1 to ⌊n/2⌋ do
6:     A[i] = MAX(A[i], A[n - i + 1])
7:   end for
8:   x = SELECTMAX(A[], ⌊n/2⌋)
9:   return x
10: end function

```

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

→ Caso Base $T(1) = C$

→ $C \left\{ C \left(\frac{n}{2} \right) \right.$

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

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

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

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

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

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

$$\frac{Cn}{2} \sum_{i=0}^{k-1} \frac{1}{2^i} + T(1)$$

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

Acotaremos esto a una serie geométrica

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

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

$$Cn \left(\frac{1}{1 - \frac{1}{2}} \right) + C$$

$$T(n) \leq 2Cn + C \quad \} \quad T(n) \in O(n)$$

Inferior

$$Cn(n) \leq Tn \quad \} \quad T(n) \in \Omega(n)$$

$$\text{ob } T(n) \in \Theta(n)$$

Ejercicio 2 (5 pts). 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.

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

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

$$\hookrightarrow n = 2^k$$

$$\lfloor \frac{2^k}{4} \rfloor = \frac{2^k}{4} = 2^{k-2}$$

$$\begin{aligned} 2^{k-2} & \cdot 2^{k-2} \\ 2^{k-2} & \cdot 2^{k-2} \\ 2^{k-2} & \cdot 2^{k-2} \\ 2^{k-2} & \cdot 2^{k-2} \end{aligned}$$

$$9T(\frac{2^k}{4})$$

$$T(2^{k-2}) = 9T(2^{k-4}) + 2^{2k-4} - 2^{k-2}$$

\downarrow

$$1^\circ \quad 9T(2^{k-2}) + 2^{2k} - 2^k$$

$$2^\circ \quad 9[9T(2^{k-4}) + 2^{2k-4} - 2^{k-2}] + 2^{2k} - 2^k$$

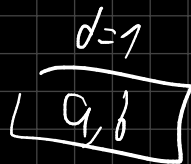
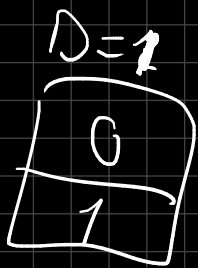
$$3^\circ \quad 81T(2^{k-4}) + 9 \cdot 2^{2k-4} + 2^{2k} - 9 \cdot 2^{k-2} - 2^k$$

$$T(2^{k-6}) = 9T(2^{k-8}) + 2^{2k-8} - 2^{k-3}$$

$$4^\circ \quad 9^3 T(2^{k-6}) + 9^2 \cdot 2^{2k-8} + 9 \cdot 2^{2k-4} + \underbrace{2^{2k}}_{2^k} - 9^2 2^{k-4} - 9 \cdot 2^{k-2} - 2^k$$

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

$$2^k$$



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

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

- 1 Expandir la recurrencia
- 2 Llegar al caso inicial

$$\begin{aligned}
 f(n) &= 2f(n-1) + 1 && 1 \\
 &= 2(2f(n-2) + 1) + 1 \\
 &= 2^2 f(n-2) + 2 + 1 && 2 \\
 &= 2^2 (2f(n-3) + 1) + 2 + 1 \\
 &= 2^3 f(n-3) + 4 + 2 + 1 && 3
 \end{aligned}$$

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

$$\begin{aligned}
 \underline{k=n-1} &= 2^{n-1} f(1) + 2^{n-2} + 2^{n-3} + \dots + 2^0 \\
 &= 2^{n-1} \cdot 1 + \sum_{i=0}^{n-2} 2^i \quad \left\{ \begin{array}{l} 2^{n-1} - 1 \end{array} \right.
 \end{aligned}$$

$$= 2^{n-1} + 2^{n-2} + \dots + 2^0 \Rightarrow 2 \cdot 2^{n-1} - 1 \Rightarrow \boxed{2^n - 1}$$

$$\sum_{i=1}^b a_i - a_{i+1} = a_1 - a_{b+1}$$

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

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

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

$$(F(n-1) - 2F(n-2) = 1) \times 2 = 2^1$$

$$(F(n-2) - 2F(n-3) = 1) \times 4 = 2^2$$

$$F(n-3) - 2F(n-4) = 1 \quad \times 8 = 2^3$$

$$\times 16 = 2^4$$

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

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

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

telescopica

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

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

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

2° Forma

- Formula Explicita
(Expandir)

1° Forma

Tantear la formula
y hacer Inducción

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

$$\rightarrow G(k) = f(k) + 1 = \underline{2f(k-1) + 1} + 1 = 2(f(k-1) + 1)$$

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

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

$$G(n) = 2G(n-1)$$

$$G(n-1) = 2G(n-2)$$

$$G(n-2) = 2G(n-3)$$

...

$$G(2) = 2G(1)$$

$$\rightarrow G(n) = 2^{n-1} G(1) = 2^n$$

$n-1$
times

$$G(n) = 2 \cdot 2 \cdot 2 \dots G(1)$$

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

2° 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

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

$$F(2^0) = 1 \rightarrow \text{caso base}$$

telescopia

$k =$

$$8T(1) + 8 + 8 \quad (15)$$

F creciente: $x \leq y \rightarrow f(x) \leq f(y)$

En general

cualquier

valor

estara

entre 2 potencias de 2

$$\Rightarrow 2^\alpha \leq n < 2^{\alpha+1}$$

$$\hookrightarrow \alpha \leq \lg(n) < \alpha+1$$

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

del
ejercicio
anterior

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

$$[\lg n < \alpha 2^\alpha + 2^\alpha]$$

Sabemos que,

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

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

$$\lg n \left(\frac{n}{2} \right)$$

$$(\alpha+2)2^{\alpha+1} < (\lg n + 2) \underbrace{2^\alpha}_{\leq n} \cdot 2 < (\lg n + 2) n \cdot 2$$

$$\leq 3 \lg n \cdot n \cdot 2$$

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

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

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

$$\sqrt{n} = \sqrt{4^m} = 4^{\frac{m}{2}} \cdot 4^0$$

Para potencias de 4

$$4^{\frac{3m}{2}} = 8^m$$

$$n = 4^m$$

$$m \geq 1$$

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

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

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

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

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

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

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

...

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

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

$$n = 4^m$$

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

$$T(4) = 15$$

$$8^m$$

$$8^{m-1}$$

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

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

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

$$\vdots$$

$$k=1 \quad T(4^1) - 8T(4^0) = 8^{(1)} - 1$$

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

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

$$a_k - a_{k-1} = 8^3 - 8^{m-k}$$

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

$$\sum_{k=1}^3 a_k - a_{k-1} = \sum_{k=1}^3 (8^3 - 8^{m-k})$$

$$3 \cdot 8^3 - \sum_{k=1}^3 8^{m-k}$$

$$a_m - a_0 = 3 \cdot 8^3 - \sum_{k=0}^3 8^k$$

$$T(4^m) - 8^3 T(1) = 3 \cdot 8^3 - \sum_{k=0}^3 8^k$$

$$T(4^m) - 8^3 = 3 \cdot 8^3 - \frac{8^3 - 1}{7}$$

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

$$T(4^m) = 8^3 + 3 \cdot 8^3 - \frac{8^3 - 1}{7}$$

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

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

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

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

$$\exists c, n_0 > 0 \text{ tal } \forall n \geq n_0$$

$$n \geq n_0 \longrightarrow T(n) \leq \text{---} n^2$$

Inducción Débil

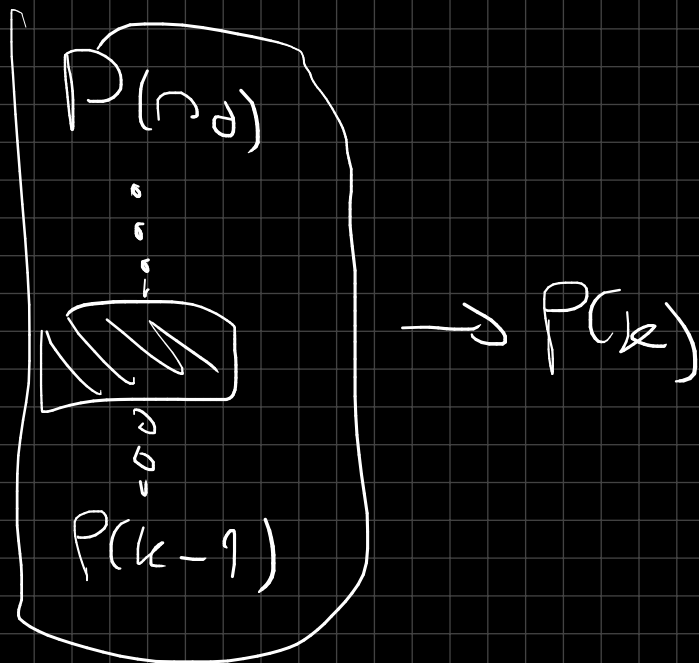
C.B. $n=1$

$$P(k-1) \rightarrow P(k) \quad \forall k \in \mathbb{N}$$

Inducción Fuerte

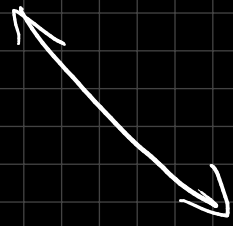
C.C.B.B $[1, k-1]$

No miras al inmediato anterior



Probar que

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



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

Probar:

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



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

Obs:

$$x \leq y \wedge x \leq z$$

$$y \leq z$$

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

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

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

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

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

Si:

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

para $c=2$

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

$\forall n \geq 2$

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

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

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

$$2^{\log k} + \log 2025$$

$$C(k-2)(\log k - 2) + k$$

$$C(k \log k - 2k - 2 \log k + 4) + k$$

$$\log k < k$$

$$Ck \log k + k + 4C + C(-2k - 2 \log k)$$

$$C \log k + k + 4C - 2Ck - 2C \log k$$

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

$$c \lg k + k + 4c - 4c \quad >$$



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

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

$$1-4c > 0$$

$$1 > 4c$$

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

$$c \leq \frac{1}{8}$$

Teorema Maestro

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

$$\log_b^a \quad \text{vs} \quad k$$

$$\log_b^a \neq k \quad \rightarrow \quad \Theta\left(n^{\max\{\log_b^a, k\}}\right)$$

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

$$T(n) \begin{cases} n & 1 \leq n \leq 2 \\ 6T(\lfloor \frac{n}{3} \rfloor) + n^2 - n & \end{cases} \quad 6+9-3$$

$$T(3) = 12$$

$$n = 3^m$$

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

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

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

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

⋮

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

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

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

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

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

$$a_k - a_{k-1} = 6^{m-k} (q^k - 3^k) \quad \left\{ \begin{array}{l} q_k = 6^{m-k} T(3^k) \end{array} \right.$$

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

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

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

$$q = \frac{9}{6} = \frac{3}{2}$$

$$p = \frac{3}{6} = \frac{1}{2}$$

$$a_m - a_0 = 6^3 \sum_{k=1}^3 q^k - 6^3 \sum_{k=1}^3 p^k$$

$$= 6^3 \left(\frac{\left(\frac{3}{2}\right)^{m+1} - \frac{3}{2}}{\frac{3}{2} - 1} - \frac{\left(\frac{1}{2}\right)^{m+1} - \frac{1}{2}}{\frac{1}{2} - 1} \right) \sum_{k=0}^3 q^k$$

$$\hookrightarrow \frac{a^{m+1} - a}{a - 1}$$

Serie Geometrie

$$\sum_{k=1}^3 a^k = \frac{a^{m+1} - a}{a - 1}$$

11

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

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

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

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

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

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

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

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
manipular

Para $\forall n \geq 4$
tenemos

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

s:

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

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

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

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

$$n \geq 4$$

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

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

$$\Rightarrow c n \lg n + 4c - 2cn - 2c \lg n + n$$

Por lo que

$\lg n < n$
 \uparrow
 si $\lg n$ va
 ser mayor a si
 quitas n

$$c n \lg n + 4c - 2cn - 2c \lg n + n > c n \lg n + 4c + n - 2cn - 2cn$$

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

$$c n \lg n + 4c + n(1 - 4c) > c n \lg n$$

Esto sera mayor a $c n \lg n$ siempre y cuando $1 - 4c > 0$

$$o.o \quad 1 - 4c > 0$$

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

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

$$T(n) = 3n^2 + n - n \cdot (\log_3 n)^2$$

$$T(3) = 30 - 24$$

$$= 6$$

$$T(3) = 12$$

$$27^{\log_3^2}$$

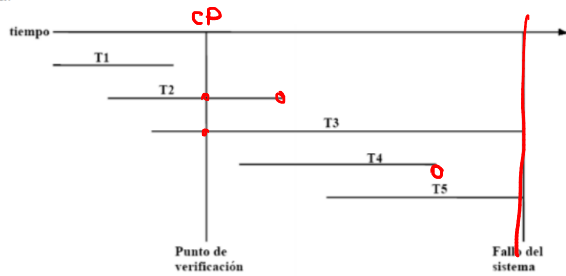
$$2^{\log_3^2}$$

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

$$n = 3^3$$

$$T(n)$$

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. $CP[T_2, T_3]$

- 2.
- 3.
- 4.
- 5.

1° $CP[T_2, T_3]$

2° $REDO = []$

$UNDO = [T_2, T_3] \leftarrow \text{Activas}$

3°

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

5° $REDO [T_2, T_4]$
 $UNDO [T_3, T_5]$

$undo(UNDO)$
 $redo(REDO)$

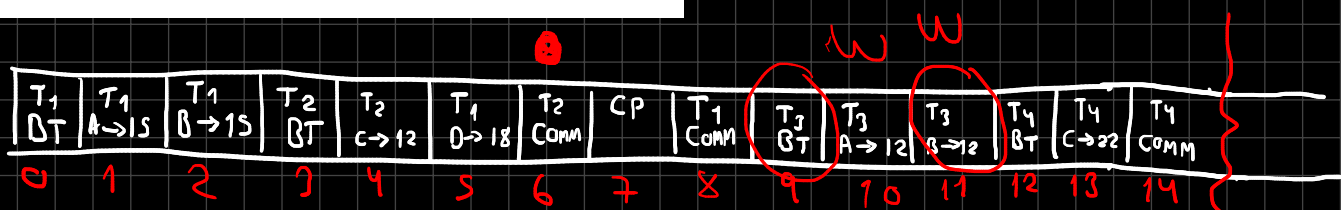
Flush T_1

UNDO

```
<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

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



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

void recoverDB(vector<RecordLog> 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);
            }
        }
    }
}
```

T_1, T_2, T_3, T_4

$REDO = [T_2, T_1, T_4]$

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

Transaction	Operation	Value
T3	Read(Z)	50
T3	Read(W)	20
T2	Read(X)	30
T3	Z = Z + W	70
T3	Write(Z)	70
T3	Commit	70
T2	read(Z)	70
T1	read(X)	30
T2	Z = Z * X	2100
T2	abort	ABORT
T1	X = X + 10	40
T1	Write X	40

$$m = 3$$

$$n = 40$$

$$m+1 = 4 \text{ hijos}$$

$$k = 2$$

$$\text{Num pages Data} = \frac{40}{4 \cdot 2} = 2$$

