Ejercicio 7:

Caso 1: $\log_b a > c$, $T(n) = \Theta(n^{\log}$

Caso 2: $\log_b a = c$, $T(n) = \Theta(n^c \lg n)$

Caso 3: $\log_b a < c$, $T(n) = \Theta(n^c)$

a) $T(n) = 2T(n/2) + n^4 \qquad f(n) = n^4$

$n^{\log_b a} = n^{\log_2 2} = n$

Caso 2 $\rightarrow f(n) \neq n$

Caso 1 $\rightarrow n^{1-\varepsilon} \stackrel{?}{=} n^4$, nunca va a poder serlo

Caso 3 $\rightarrow n^{1+\varepsilon} \stackrel{?}{=} n^4$, cuando $\varepsilon = 3$ va a ser igual a $f(n)$

Regularidad, debemos comprobar $af(n/b) \leq cf(n)$, $c < 1$

$a = 2$, $b = 2$, $f(n) = n^4$

$$af(n/b) = 2f(n/2)$$
$$= 2\frac{n^4}{2}$$
$$= n^4 \leq n^4$$
$$= cf(n) \quad , c \stackrel{?}{=} 1$$

b) $T(n) = 2T(7n/10) + n$

$a = 2$, $b = \frac{10}{7}$, $f(n) = n$, $c = 1$

$n^{\log_{\frac{10}{7}} 2} = n^{1,94}$ Aplico el caso 1, ya que $\log_b a > c \rightarrow 1,94 > 1$

Caso 1: $O(n^{1,94})$

c) $T(n) = 16T(n/4) + n^2$

$n^{\log_4 16} = n^2$

Por lo tanto por el caso 2 $T(n) = \Theta(n^2 \lg n)$

d) $T(n) = 7T(n/3) + n^2$

$n^{\log_3 7} = n^{1,77}$, $f(n) = n^2$, $1,77 < 2$ caso 3

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

e) $T(n) = 7T(n/2) + n^2$

$a = 7$

$b = 2$

$f(n) = n^2$

$n^{\log_b a} = n^{\log_2 7} = n^{2,81} > n^2$ caso 1

Por el caso 1 $T(n) = O(n^{2,81})$

f) $T(n) = 2T(n/4) + n^{1/2}$

$a = 2$

$b = 4$

$f(n) = n^{1/2}$

$n^{\log_4 2} = X$

$4^X = 2$

$(2^2)^X = 2^1$

$2X = 1$

$\boxed{X = \frac{1}{2}}$

$n^{1/2} = f(n) \rightarrow$ caso 2 , ya que $n^{\log_b a} = c$

$T(n) = \Theta(n^{1/2} \lg n)$

Ejercicio 1.

Demostrar que $6n^3 \neq O(n^2)$

Demostramos utilizando la def formal de la notacion O

$f(n) \in O(g(n))$ si existe una cte positiva c y un valor n_0 tal que $|f(n)| \leq C.|g(n)|$ para todo $n \geq n_0$

Para demostrar que $6n^3 \notin$ a $O(n^2)$ hay que demostrar que no hay una cte positiva c tal que $6n^3 \leq C.|n^2|$, $\forall\, n \geq n_0$

Divido por n^2 , $6n \leq c$ y eso no siempre va a ser correcto porque puede haber un n lo suficientemente grande tal que $6n > C$

Por lo tanto concluimos que $6n^3 \neq O(n^2)$

Ejercicio 2 :

Array de 10 elementos para el mejor caso de Quicksort

Como funciona Quicksort

1° Se elije un pivote y se colocan los menores a el a la izq y los mayores a la derecha

3	6	7	8	9	13	20	21	30	32

Elijo como pivote el elemento del medio y verifico que hayan quedado menores a la izq y mayores a la der.

El mejor caso sería $O(n \log n)$

Ejercicio 3

Quick Sort → Elijira un pivote en cada partición y dividirá el array en 2 subarrays, y así repetidamente por lo tanto sera $O(n^2)$

Insertion Sort → Va comparando el segundo con el primero, y al ser todos = lo va a hacer n veces por lo tanto $T(n) = O(n)$

Merge Sort → Tecnica divide y venceras, y cuando divida la a quedar arrays con mismos valores, por lo tanto el $T(n) = O(n \lg n)$