Parcial 2

1. $T(n) = \begin{cases} C_1 & , n=1 \\ 2T(n/2) + C_2 + \sum_{j=1}^{n} C_3 & , n>1 \end{cases}$

Paso 1: $2T(n/2) + C_3 \cdot n + C_2$

Paso 2: $2[2T(n/4) + (n/2) \cdot C_3 + C_2] + C_3 \cdot n + C_2$

$= 2^2 T(n/2^2) + n \cdot C_3 + 2C_2 + C_3 \cdot n + C_2 = 2^2 T(n/2^2) + 2C_3 \cdot n + 3C_2$

Paso i: $2^i T(n/2^i) + i \cdot C_3 \cdot n + (2^i - 1) \cdot C_2$

$n/2^i = 1$ $\quad \Rightarrow$ Reemplazo i y n:

$n = 2^i$

$\log_2 n = i$ $\quad T(n) = 2^{\log_2 n} \cdot T(2^i/2^i) + \log_2(n) \cdot n \cdot C_3 + (2^{\log_2 n} - 1) \cdot C_2$

$T(n) = n \cdot T(1) + \log_2(n) \cdot n \cdot C_3 + (n-1) \cdot C_2 = n \cdot C_1 + \log_2(n) \cdot n + n \cdot C_2 - C_2 :- O(n \cdot \log_2(n))$

2. El $O(n)$ es 2^n ya que crece mucho más rapido que las funciones cúbicas, y que las funciones constantes.

3. $O(\log_{10}(n))$

1 HORA — $\log_{10}(1000) = 3$ operaciones para $n = 1000$

3 HORAS — $3 \cdot 3/1 = 9$ operaciones para $n = ?$

Se debe hallar n y el planteo sería el siguiente:

$\log_{10}(n) = 9$ $\quad \Rightarrow$ Aplico propiedad $\log_a b = c \Rightarrow a^c = b$

$n = 10^9$

El máximo tamaño de entrada que podría ejecutar el algoritmo si se dispone de 3 horas de CPU sería $n = 1.000.000.000$

4. 1) $T(n) = 0 + 2T(n/2) \Rightarrow$ Opción (b)

2) $(2^n + 2)(\log(n) + 5) = 2^n \cdot \log(n) + 5 \cdot 2^n + 2\log(n) + 10 :- O(2^n \cdot \log(n)) \Rightarrow$ Opción (a)

4) $T(n) = 4$ $\quad$ Si $n=1$ $\quad$ $T(n)$ para $n \geq 4$

$T(n) = 2 * T(n/2) + 5 * n + 1$ $\quad$ Si $n \geq 2$

$T(4) = 2 * T(4/2) + 5 * 4 + 1 = 2 * 19 + 20 + 1 = 59$

$T(2) = 2 * T(2/2) + 5 * 2 + 1$

$= 2 * 4 + 10 + 1 = 19$