



## Práctica 1

1. Probar utilizando el método de sustitución que  $T(n) \in O(\lg(n))$ .

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

2. Utilizar un árbol de recurrencia para obtener una cota asintótica para

$$\begin{aligned} T(n) &= T(n-1) + T(a) + cn & \text{si } n > a \\ T(n) &= c' & \text{si } n \leq a \end{aligned}$$

donde  $a \geq 1$ ,  $c, c' > 0$  son constantes.

3. Sean  $a, b \in \mathbb{R}^+$ , utilizar el método de sustitución para encontrar funciones  $f(n)$  tales que  $T(n) \in \Theta(f(n))$  para las siguientes recurrencias.

1.

$$T(n) = \begin{cases} a & n = 1 \\ 2T(\lfloor n/2 \rfloor) + b & n > 1 \end{cases}$$

2.

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

Ayuda: Recuerde las propiedades:

- $\forall n, a, b \in \mathbb{Z}, a \neq 0 \wedge b \neq 0 \Rightarrow \lfloor \lfloor n/a \rfloor / b \rfloor = \lfloor n/(ab) \rfloor$
- $\forall n, a, b \in \mathbb{R}^+, a^{\log_b(n)} = n^{\log_b(a)}$

4. Utilizar el teorema Maestro para encontrar cotas asintóticas  $\Theta$  para las siguientes recurrencias (asumir que  $T(1) > 0$ ):

1.  $T(n) = 4T(n/2) + n$
2.  $T(n) = 4T(n/2) + n^2$
3.  $T(n) = 4T(n/2) + n^3$

5. Para cada una de las siguientes funciones, determinar si son suaves o no. Demostrar.

1.  $\ln(n)$
2.  $n^2$
3.  $n^n$

6. Utilice un árbol de recurrencia para encontrar una cota asintótica  $\Theta$  para la recurrencia

$$T(n) = \begin{cases} a & n = 1 \\ 4T(\lceil n/2 \rceil) + cn & n > 1 \end{cases}$$

donde  $a$  y  $c$  son constantes positivas. Verifique que la cota encontrada es correcta. Ayuda: demostrar por suavidad.

7. Encontrar cotas asintóticas  $\Theta$  para cada una de las siguientes recurrencias (asumir que  $T(1) > 0$ ). Demostrar.

- $T(n) = T(n/2) + 1$
- $T(n) = T(n-1) + n$
- $T(n) = T(\sqrt{n}) + 1$

8. Dadas las siguientes definiciones en pseudocódigo de  $exp_1$  y  $exp_2$ , calcular el trabajo de cada una de ellas y determinar qué función es más eficiente.

$$\begin{aligned} exp_1\ 0 &= 1 \\ exp_1\ (n+1) &= 2 \times exp\ n \\ \\ exp_2\ 0 &= 1 \\ exp_2\ n &= \text{case par } n \text{ of} \\ &\quad \text{true} \rightarrow \text{square}\ (exp_2\ (\text{div } n\ 2)) \\ &\quad \text{false} \rightarrow 2 \times (exp_2\ (n-1)) \end{aligned}$$

9. Dados los siguientes pseudocódigos que implementan distintos algoritmos para invertir los elementos de una lista, calcular el trabajo de  $reverse_1$  y  $reverse_2$  y determinar qué función es más eficiente.

$$\begin{aligned} reverse_1 &: [a] \rightarrow [a] \\ reverse_1\ [] &= [] \\ reverse_1\ (x \triangleleft xs) &= (reverse_1\ xs) @ [x] \\ \\ [] @ ys &= ys \\ (x \triangleleft xs) @ ys &= x \triangleleft (xs @ ys) \\ \\ revStack &: [a] \rightarrow [a] \\ revStack\ []\ ys &= ys \\ revStack\ (x \triangleleft xs)\ ys &= revStack\ xs\ (x \triangleleft ys) \\ \\ reverse_2 &: [a] \rightarrow [a] \\ reverse_2\ xs &= revStack\ xs\ [] \end{aligned}$$

10. Dado el siguiente pseudocódigo de un algoritmo que construye un árbol binario a partir de una lista:

$$\begin{aligned} \text{data Tree } a &= \text{Empty} \mid \text{Leaf } a \mid \text{Node } (\text{Tree } a) (\text{Tree } a) \\ \\ split &: [a] \rightarrow [a] \times [a] \\ split\ [] &= ([], []) \\ split\ [x] &= ([x], []) \\ split\ (x \triangleleft y \triangleleft xs) &= \text{let } (ys, zs) = split\ xs \\ &\quad \text{in } (x \triangleleft ys, y \triangleleft zs) \\ \\ toTree &: [a] \rightarrow \text{Tree } a \\ toTree\ [] &= \text{Empty} \\ toTree\ [x] &= \text{Leaf } x \\ toTree\ (x \triangleleft y \triangleleft xs) &= \text{let } (ys, zs) = split\ (x \triangleleft y \triangleleft xs) \\ &\quad (t1, t2) = toTree\ ys \parallel toTree\ zs \\ &\quad \text{in Node } t1\ t2 \end{aligned}$$

1. Expresar las recurrencias correspondientes al trabajo y a la profundidad de la función  $toTree$ , asumiendo que  $W_{split}(n) = S_{split}(n) = O(n)$ , siendo  $n$  la longitud de la lista que recibe.
2. Resolver la recurrencia encontrada en el apartado anterior utilizando el teorema Maestro. Expresar la solución utilizando la notación  $\Theta$ .