Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2283 - Diseño y Análisis de Algoritmos

Ayudantía 1

27 de agosto de 2021

Profesor Marcelo Arenas

Bernardo Barías

Pregunta 1 - Límites y complejidad asintótica

Sean $f, g : \mathbb{N} \to \mathbb{R}^+$. Suponga que $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ existe y es igual a ℓ. Demuestre las siguientes afirmaciones (pueden ser utilizadas en las evaluaciones):

a) Si $\ell = 0$, entonces $f \in O(g)$ y $g \notin O(f)$.

b) Si $\ell = \infty$, entonces $g \in O(f)$ y $f \notin O(g)$.

c) Si $\ell \in \mathbb{R}^+$, entonces $f \in \Theta(g)$.

d) Muestre que $\log n \in O(n^\epsilon)$ para todo $\epsilon > 0$.

Pregunta 2 - Teorema maestro

a) Aplique el teorema maestro en la ecuación de recurrencia

$$T(n) \ = \ 2T(n/4) + \sqrt{n}.$$

b) ¿Puede ser aplicado el teorema maestro a la ecuación de recurrencia $T(n)$ definida a continuación?

$$T(n) \ = \ \begin{cases} 1 & \text{si } n = 0 \\ 4T(n/2) + n^2 \log n & \text{si } n > 0 \end{cases}$$

Calcule una cota asintótica superior de $T(n)$. Asuma que $n = 2^k$, con $k \in \mathbb{N}$.

Pregunta 3 - MergeSort

Un ejemplo clásico de un algoritmo de tipo dividir para conquistar es MergeSort (o ordenamiento por mezcla), el cuál fue creado por John Von Neumann en 1945. MergeSort es un algoritmo de ordenación que está compuesto por dos subrutinas.

Algoritmo 1 MergeSort($A[1 \dots n]$)

```
if n > 1 then
    m ← ⌊n/2⌋
    MergeSort(A[1...m])
    MergeSort(A[m+1...n])
    Merge(A[1...n], m)
end if
```

Algoritmo 2 Merge($A[1 \dots n], m$)

```
i ← 1
j ← m + 1
for  k ← 1 to n do
    if  j > n then
    B[k] ← A[i]; i ← i + 1
    else if i > m then
    B[k] ← A[j]; j ← j + 1
    else if A[i] < A[j] then
    B[k] ← A[i]; i ← i + 1
    else
    B[k] ← A[j]; j ← j + 1
    end if
end for
for  k ← 1 to n do
    A[k] ← B[k]
end for
```

Explique detalladamente el algoritmo y demuestre que MergeSort tiene complejidad $O(n\ log(n))$