CC371: 2.- Practica Calificada

Propiedades & Recurrencia

Brando Miguel Palacios Mogollon

bpalaciosm@uni.pe

Universidad Nacional de Ingenieria
Escuela Profecional de Ciencias de la Computacion

June 25, 2020

Overview

Contenido

Pregunta 1.b

Demostrar:

$$\mathcal{O}(f(n)) = \mathcal{O}(g(n)) \leftrightarrow f(n) \in \mathcal{O}(g(n)) \land g(n) \in \mathcal{O}(f(n))$$

Resolución:

$(\longrightarrow)$ Siendo

$$\mathcal{O}(f(n)) = \mathcal{O}(g(n)) \Rightarrow \mathcal{O}(f(n)) \subset \mathcal{O}(g(n)) \land \mathcal{O}(g(n)) \subset \mathcal{O}(f(n)) \quad (1)$$

Recordando, dado que $f(n) \leq c \cdot f(n),\ c \in \mathbb{R}^+$:

$$\exists c \in \mathbb{R}^+, \exists k \in \mathbb{N}, n \geq k, f(n) \leq c \cdot f(n) \Rightarrow f(n) \in \mathcal{O}(f(n)) \quad (2)$$

Demostración

Sabiendo esto, de (1) podemos decir:

- $f(n) \in \mathcal{O}(f(n)) \subset \mathcal{O}(g(n)) \Rightarrow f(n) \in \mathcal{O}(g(n))$
- $g(n) \in \mathcal{O}(g(n)) \subset \mathcal{O}(f(n)) \Rightarrow g(n) \in \mathcal{O}(f(n))$

$\longrightarrow \quad f(n) \in \mathcal{O}(g(n)) \land g(n) \in \mathcal{O}(f(n))$

$(\longleftarrow)$ Siendo que : $(f(n) \in \mathcal{O}(g(n)) \rightarrow \mathcal{O}(f(n)) \subset \mathcal{O}(g(n)))$

Podemos decir que:

- $f(n) \in \mathcal{O}(g(n)) \Rightarrow \mathcal{O}(f(n)) \subset \mathcal{O}(g(n))$
- $g(n) \in \mathcal{O}(f(n)) \Rightarrow \mathcal{O}(g(n)) \subset \mathcal{O}(f(n))$

$\longrightarrow \quad \mathcal{O}(f(n)) = \mathcal{O}(g(n))$

$$\therefore\ \mathcal{O}(f(n)) = \mathcal{O}(g(n)) \leftrightarrow f(n) \in \mathcal{O}(g(n)) \land g(n) \in \mathcal{O}(f(n))$$

FACULTAD DE CIENCIAS

Pregunta 1.d

Demostrar:

$$f(n) \in \mathcal{O}(g(n)) \wedge f(n) \in \mathcal{O}(h(n)) \rightarrow f(n) \in \mathcal{O}(min(g(n), h(n)))$$

Resolución:

Tomando un $\delta(n) = \min(h(n), g(n))$
Por definición de orden:

$$\exists c_1 \in \mathbb{R}^+, \exists k \in \mathbb{N}, \forall n \geq k, [f(n) \leq c_1 \cdot g(n)] \tag{1}$$

$$\exists c_2 \in \mathbb{R}^+, \exists k \in \mathbb{N}, \forall n \geq k, [f(n) \leq c_2 \cdot h(n)] \tag{2}$$

De la inecuación (1),(2) realizamos una suma:

$$f(n) \leq c_1 \cdot g(n) \ , \ f(n) \leq c_2 \cdot h(n)$$

Demostración

$$2f(n) \leq c_1 g(n) + c_2 h(n)$$

Pero siendo δ(n) minimo, podemos ver:

$$(c_1 + c_2)\delta(n) \leq c_1 g(n) + c_2 h(n) \tag{3}$$

De esto podemos ver:

$$2f(n) \leq (c_1 + c_2)\delta(n) \leq c_1 g(n) + c_2 h(n)$$

$$2f(n) \leq (c_1 + c_2) \cdot \delta(n)$$

$$f(n) \leq \underbrace{\frac{(c_1 + c_2)}{2}}_{c>0} \delta(n) \Rightarrow f(n) \in \mathcal{O}(\delta(n))$$

$$\therefore \quad f(n) \in \mathcal{O}(\min(h(n), g(n))$$

FACULTAD DE CIENCIAS

Pregunta 1.f

Demostrar:

$$f_1(n) \in \mathcal{O}(g(n)) \land f_2(n) \in \mathcal{O}(h(n)) \to f_1(n) \cdot f_2(n) \in \mathcal{O}(g(n) \cdot h(n))$$

Resolución:

Por definición, sabemos que:

$$\text{Por: } f_1(n) \in \mathcal{O}(g(n)) \to f_1(n) \leq c_1 \cdot g(n) \quad , c_1 \in \mathbb{R}^+ \qquad (1)$$

$$\text{Por: } f_2(n) \in \mathcal{O}(h(n)) \to f_2(n) \leq c_2 \cdot h(n) \quad , c_2 \in \mathbb{R}^+ \qquad (2)$$

Multiplicamos las inecuaciones (1) y (2):

$$f_1(n) \cdot f_2(n) \leq \underbrace{c_1 c_2}_{c} \cdot (g(n) \cdot h(n))$$

$$\therefore \; f_1(n) \cdot f_2(n) \in \mathcal{O}(g(n) \cdot h(n))$$

Contenido

Demostración

Pregunta 2.b

Demostrar la siguiente propiedad. Sean las funciones f,g: $\mathbb{N} \to \mathbb{R}^+ \cup \{0\}$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \to \mathcal{O}(g(n)) \subset \mathcal{O}(f(n))$$

Recordando la definición de limites:

$$\forall M > 0, \exists A > 0, n \in \mathbb{N} \wedge A < n \Rightarrow M < \frac{f(n)}{g(n)} \tag{1}$$

Demostración

Despejando la inecuación (1):

$$M \cdot g(n) < f(n) \Rightarrow \; g(n) < \frac{1}{M} f(n)$$

$$\Rightarrow g(n) \in \mathcal{O}(f(n))$$

De la propiedad: $(f(n) \in \mathcal{O}(g(n)) \rightarrow \mathcal{O}(f(n)) \subset \mathcal{O}(g(n)))$

$$g(n) \in \mathcal{O}(f(n)) \rightarrow \mathcal{O}(g(n)) \subset \mathcal{O}(f(n))$$

$\therefore$ Proposición Verdadera

Contenido

Las torres de Hanói

Problema 3.

Siendo el algoritmo de Las Torres de Hanoi anexa:

```
void hanoiTower(int n, int inic, int tmp,int final){
 if(n > 0){
  hanoiTower(n-1,inic,final,tmp);
  cout<<"De poste "<<inic<<" a "<<final<<endl;
  hanoiTower(n-1,tmp,inic,final);
 }
}
```

$$f(n) = \begin{cases} 0 & n = 0 \\ 2f(n-1)+1 & n > 0 \end{cases} \tag{2}$$

Analizando:

$$f(n) = 2 \cdot f(n-1) + 1 = 2(2 \cdot f(n-2) + 1) + 1 = 2^2 f(n-2) + (2^2 - 1)$$

Podemos decir:

$$f(n) = 2^i \cdot f(n-i) + (2^i - 1) = 2^i \cdot (f(n-i) + 1) - 1$$

Para $n - i = 0 \rightarrow i = n$:

$$f(n) = 2^n \cdot (f(0) + 1) - 1 = 2^n - 1 \leq 2^n$$

$$\therefore \; f(n) \text{ es } \mathcal{O}(2^n)$$

Contenido

FACULTAD DE CIENCIAS

Problema 4.

Dada la siguiente función recursiva:

```
int Rec3(int n){
if(n<=1)
    return(1);
else
    return(Rec3(n-1)+Rec3(n-1));
}
```

determine su complejidad.

De la función, podemos obtener la siguiente función:

$$f(n) = \begin{cases} 1 & n \leq 1 \\ 2 \cdot f(n-1) & n > 1 \end{cases}$$

Función Recursiva

Analizando, peor caso:

$$f(n) = 2 \cdot f(n-1) = 2 \cdot (2 \cdot f(n-2)) = 2^2 \cdot f(n-2) = \cdots$$

Entonces podemos decir:

$$f(n) = 2^i \cdot f(n-i)$$

Tomando el caso de $n - i = 1 \rightarrow i = n - 1$:

$$f(n) = 2^{n-1} \cdot f(1) = 2^{n-1} \leq \frac{1}{2} \cdot 2^n$$

$$\Rightarrow \exists c = \frac{1}{2}, \exists k = 1, \forall n \geq k, [f(n) \leq \frac{1}{2} \cdot 2^n]$$

$$\therefore f(n) \text{ es } \mathcal{O}(2^n)$$