Ricardo Montiel Manriquez

30 de Junio del 2021

①

Algoritmo 1 misterio(n)

```
1: total=0;
2: for i=0 to n-1 do
3:     for j=n-1 to i down by 1 do
4:         total=total+1
5:     end for
6: end for
7: return total);
```

En la linea 1 tenemos un acceso y 1 una asignación = 2.

En la linea 2 el ciclo se repite n veces. = n

En la linea 3 el ciclo se repite $(n)+(n-1)+(n-2)+\ldots=0$ y esto nos da $n(n+1)/2$

En la linea 4 tenemos 2 accesos, 1 asignación y 1 suma = 4

En la linea 7 tenemos un acceso y un retorno de valor = 2

$$\therefore f(n)=\left(\left(\frac{n(n+1)}{2}\right)*4\right)+4$$

②

a) $2^{n+1} \in \Omega(2^n)$

Verdadero

$\exists c \in \mathbb{R}^+$ y $\exists n_0 \in \mathbb{N} \cup \{0\} \mid \forall n \geqslant n_0$,

$0 \leq c\,g(x) \leq f(x)$

$g(n) = 2^n$

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

Tomamos $n_0 = 1$

P.D: $0 \leq (2^n) \leq 2^{n+1}$

$2^n \geqslant 0$

$c(2^n) \leq 2^{n+1}$

$c = 1$

$\Rightarrow 1 \cdot (2^n) \leq 2$

b) $\log_2 2^n \in O(n \log 2^n)$

Para $n \geq 2$ tenemos que $n \log 2^n \geq \log_2 2^n$

$\therefore$ Tomando $n_0 = 1$, se cumple que.

$\log_2 2^n \geq n \log 2^n$

3

a) $f(n) = (n^2 - 2n + 1)/2$; $g(n) = 4n$.

$g(n) = 4n \in O(n)$

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

$n^2 \in O(n^2)$, $2n \in O(n)$, $1 \in O(1)$

$O(1) \in O(n)$ y

$O(n) \in O(n^2)$

$\therefore$ $(n^2 - 2n + 1)/2 \in O(n^2)$

y por dominancia

$g(n) \in O(f(n))$ pero $f(n) \notin O(g(n))$.

b) $f(n) = n \log_2 n$ $g(n) = n^3\sqrt{n}/2$

$f(n) = n \log_2 n$

$n \in O(n^2)$ por propiedad 5

$\log_2 n \in O(n)$

$n \cdot \log_2 n \in O(n^2)$ por regla del producto

$g(n) = n^3\sqrt{n}/2$

$n^3 \in O(n^3)$, $\sqrt{n} \in O(2^n)$

$1/2 \in O(1)$ por regla del producto

$n^3\sqrt{n}/2 \in O(2^n \cdot n^3)$

Por dominancia $f(n) \in O(g(n))$