ANÁLISE DE ALGORITMOS – 2022.2 (Avaliação 1) **2,2 de 3,5**

1. (1,0 pontos) Para a sentença a seguir, verifique se ela é verdadeira ou falsa justificando sua resposta: $\log_2(n)$ é $\Theta(\log_{10}(n))$.

$\log_2 n$ é $\Theta(\log_{10} n)$?

Existem c_1, c_2 e n_0 positivos, tal que, para todo $n \geq n_0$

$c_1 = 10^{-4}$

$c_2 = 10$

$n = 64$

$$c_1 \cdot \log_{10} n \leq \log_2 n \leq c_1 \cdot \log_{10} n$$
$$10^{-4} \cdot \log_{10} 64 \leq \log_2 64 \leq 10 \cdot \log_{10} 64$$
$$10^{-4} \cdot 2 \leq 6 \leq 10 \cdot 1$$

forçando para cima — forçando para baixo

$2 < \log_{10} 64 > 1$,
já que $\log_{10} 100 = 2$

$$0{,}0002 \leq 6 \leq 10.$$

$k > n$
$k = 1024$

$\log_{10} 1000 = 3$
logo,
$4 < \log_{10} 1024 > 3$

$$10^{-4} \cdot \log_{10} 1024 \leq \log_2 1024 \leq 10 \cdot \log_{10} 1024$$
$$10^{-4} \cdot 4 \leq 10 \leq 10 \cdot 3$$

forçando para cima — forçando para baixo

$$0{,}0004 \leq 10 \leq 30$$

A medida que o n cresce, considerando n = 64 e $c_1 = 10^{-4}$ e $c_2 = 10$, observe que a distância aumenta entre $c_1 \cdot g(n) \leq f(n)$ e $f(n) \leq c_2 \cdot g(n)$. Logo, $\log_2 n$ é $\Theta(\log_{10} n)$.

2. (1,0 pontos) Justificando sua resposta, determine se a sentença a seguir é verdadeira ou falsa: $(\log_2 n)^3$ é $O(n)$.

$(\log_2 n)^3$ é $O(n)$?

para todo

Existe c e n_0 positivos, com $n \geq n_0$, tal que:

$c = 10^2$
$n = 64$

$$\log_2 n \cdot \log_2 n \cdot \log_2 n \leq c \cdot n$$
$$\log_2 2^6 \cdot \log_2 2^6 \cdot \log_2 2^6 \leq 10^2 \cdot 2^6$$
$$6 \cdot 1 \cdot 6 \cdot 6 \cdot 1 \leq 10^2 \cdot 2^6$$
$$216 \leq 6400$$

$c = 100$
$n_0 = 64$

k > n
k = 1024

$$\log_2 2^{10} \cdot \log_2 2^{10} \cdot \log_2 2^{10} \leq 10^2 \cdot 2^{10}$$
$$10 \cdot 10 \cdot 10 \leq 100 \cdot 1024$$
$$1000 \leq 102400$$

a partir de n = 64,
considerando $c = 10^2$.

A medida que n aumenta, a distância aumenta. Logo, é O(n) a função $(\log_2 n)^3$.

3. (1,5 pontos) Escreva o algoritmo INSERTION-SORT para ordenar de forma monotonicamente decrescente no lugar de monotonicamente crescente.

```
Insertion-Sort(A, n-1)
j = 0
for i = 0 to n-1; i++
        while j ≥ 0
                if A[ j ] < A[ j+1 ]
                        aux = A[ j ]
                        A[ j ] = A[ j+1 ]
                        A[ j+1 ] = aux
                        j++
    j = 0
```

não vai parar!

1 2 3 4	2 1 3 4	2 3 1 4
i : 0	i : 0	i : 0
j : 0	j : 1	j : 2

não é o insertion!