GABARITO SEMANA 4
Monitora: Adriana Jacoto Unger

Questão 1 0.5 Pontos

Demonstrar que 2^{n+1} é $\Omega(2^n)$

Solução 1:

Dadas as funções $T(n) = 2^{(n+1)}$ e $f(n) = 2^n$, desejamos encontrar c = ? e n_0 = ?, tal que T(n) **>=** c * f(n)

$2^{n+1} >= 2 * 2^n$ para todo n>=0

logo c = 2 e $n_0 = 0$

Solução 2:

Dadas as funções $T(n) = 2^{(n+1)}$ e $f(n) = 2^n$, desejamos encontrar c = ? e n_0 = ?, tal que T(n) **>=** c * f(n)

$2^{n+1} >= 2 * 2^n >= 1 * 2^n$ para todo n>=0

logo c = 1 e $n_0 = 0$

Solução 3:

Será demonstrado por indução que 2^{n+1} é $\Omega(2^n)$ para c=1 e n_0 =1, isto é $2^{(n+1)} >= 1*2^n$ para todo n>=1.

(i) PASSO BASE: Para n=1, $2^2 >= 2^1$, 4>= 2 (VERDADEIRO)

(ii) PASSO INDUTIVO: Para k>=2, fazemos a suposição que a propriedade é válida para n=k-1, isto é

$2^k >= 2^{(k-1)}$

multiplicando ambos lados por 2, temos:

$2 * (2^k) >= 2 * (2^{(k-1)})$

$2^{k+1} >= 2^k$

Dessa forma, conclui-se que $2^{n+1)}>= 2^n$ para todo n>= 1 e, portanto, 2^{n+1} é $\Omega(2^n)$, com c = 1 e n_0 =1.

Observação: Pode-se usar também n_0 =0. Pode ser usado também c=2.

Questão 2 0.2 Pontos

Determine se a sentença é verdadeira(V) ou falsa(F):

Seja T(n) o número de operações de multiplicação realizadas pelo algoritmo que multiplica as matrizes A e B de tamanho nxn. Pode-se afirmar que $T(n) = n^3$, T(n) é $O(n^3)$, T(n) é $\Omega(n^3)$ e T(n) é $\Theta(n^3)$.

```
Multiplica(A, B, n)
para i=1 até n
        para j=1 até n
```

C[i,j] = 0
para k=1 até n
C[i,j] = C[i,j] + A[i,k] * B[k,j]
devolva C
(V)

0.3 Pontos

Questão 3

Determine se cada sentença é verdadeira(V) ou falsa(F):

(V) n + 1000 é O(n)

(F) $100n^2 + 10000000$ é $\Omega(n^2 \lg n)$

(F) 3^n é O(n)