Prova 1

The respondent's email (**henrique.padula@ufv.br**) was recorded on submission of this form.

QUESTÃO 5. O seguinte algoritmo recursivo recebe como entrada uma matriz de ordem nxn (note que as linhas e colunas da matriz variam de 0 a n-1). Indique o que faz/determina o algoritmo, e quais são as complexidades do algoritmo no pior caso e melhor caso. Justifique suas respostas. *

```
MATRIZ(n, M[0..n-1][0..n-1]) //entrada: matriz de ordem nxn
    if n == 1: //se a matriz é de ordem 1x1
        if M[0][0] ≠ 1: return false
        else: return true
    else:
        if MATRIZ(n-1, M[0..n-2][0..n-2]):
            if M[n-1][n-1] ≠ 1: return false
            for i = 1 to n-1:
                if M[n-1][i] ≠ 0 or M[n-1][i] ≠ 0:  return false

            return true
        else:
            return false
```

- (x) O algoritmo verifica se a matriz é identidade. O algoritmo possui complexidade O(nˆ2) no pior caso e O(1) no melhor caso.
- ( ) O algoritmo verifica se todos os elementos da matriz são iguais a 1 (exceto os elementos da diagonal principal que são iguais a zero). O algoritmo possui complexidade O(nˆ2) no pior caso e O(n) no melhor caso.
- ( ) O algoritmo verifica se todos os elementos da matriz são iguais a zero, exceto os elementos das duas diagonais que são iguais a 1. O algoritmo possui complexidade O(nˆ2) no pior caso e O(n) no melhor caso.
- ( ) O algoritmo verifica se a matriz é identidade. O algoritmo possui complexidade O(2nˆ2) no pior caso e O(n) no melhor caso.
- ( ) O algoritmo verifica se todos os elementos da matriz são iguais a 0 (exceto os elementos da diagonal principal que são iguais a 1). O algoritmo possui complexidade O(nˆ2) no pior caso e O(n) no melhor caso.

QUESTÃO 6. Assinale a função fechada T(n) obtida após resolver a recorrência T(n) = 3T(n/3) + n, para n>1, com T(1) = 1 (considerando n potencia de 3). Justifique sua resposta. *

- (●) T(n) = n log_3 n + n
- ( ) T(n) = n^2 log_3 n + n
- ( ) T(n) = n log_3 n + n^2
- ( ) T(n) = n log_3 n + log_3 n
- ( ) T(n) = n log_3 n + 3n

QUESTÃO 2. Para provar que 2n^2 - 4n - 20 ∈ Ω(n^2) deve-se mostrar que existem duas constantes c ∈ R+ e no ∈ Z+ tal que, 2n^2 - 4n - 20 ≥ c.n^2, para todo n ≥ no. Assinale as constantes que podem ser utilizadas para fazer essa prova. Justifique sua resposta. *

- (●) Todas as alternativas são corretas.
- ( ) c = 1, no = 7
- ( ) c = 1/4, no = 5
- ( ) c = 5/9, no = 6
- ( ) c = 2/5, no = 5

QUESTÃO 1. Determine a complexidade do seguinte algoritmo (considere entradas n potências de 2). Justifique sua resposta. *

```
ALG (n):
  if n > 1:
     ALG(n/2)
     for i = 0  to n-1 do
          for j = 0 to i-1 do
              print('* *')
     ALG(n/2)
```

- $\Theta(n^2 \log_2 n)$
- $\Theta(2^n)$
- $\Theta(n \log_2 2^n)$
- $\Theta(n^3)$
- $\Theta(n \log_2 n)$

Individual feedback

Obtido $2T(n/2) + n(n-1)/2$, aplicando Teorema mestre obtém-se: $\Theta(n^2) = \Theta(n \log_2 2^n)$

QUESTÃO 4. Sejam f(n), g(n) e h(n) os tempos de execução de pior caso de, respectivamente, três algoritmos A, B e C propostos para um mesmo problema computacional, onde n é o tamanho da entrada. Dizemos que, assintoticamente no pior caso, o algoritmo A é mais eficiente que o algoritmo B, e o algoritmo C é pior que o algoritmo B quando (justifique sua resposta): *

- ○ $f(n) \in \theta(g(n))$ e $h(n) \in \theta(g(n))$
- ○ $f(n) \in o(g(n))$ e $g(n) \in \omega(h(n))$
- ○ $g(n) \in o(f(n))$ e $h(n) \in \omega(g(n))$
- ○ $g(n) \in O(f(n))$ e $h(n) \in O(g(n))$
- ◉ $g(n) \in \omega(f(n))$ e $g(n) \in o(h(n))$

QUESTÃO 3. Para cada uma das afirmações seguintes, diga se é Verdadeiro ou Falso. Justifique sua resposta. *

A) Se $f(n) \in \Theta(g(n))$, então $f(n) \in O(g(n))$
B) $f(n) \in O(g(n))$, então $f(n) \in o(g(n))$
C) Se $f(n) \in o(g(n))$, então $g(n) \in \Omega(f(n))$
D) Se $f(n) \in \omega(g(n))$, então $f(n) \in O(g(n))$
E) Se $f(n) \in O(g(n))$, então $\lim_{n\to\infty} f(n)/g(n) = 0$
F) Se $\lim_{n\to\infty} f(n)/g(n) = \infty$, então $f(n) \in \omega(g(n))$
G) $2^{2n} \in \Theta(2^n)$
H) $2^n \in o(n!)$
I) $n^2 \log_2 n \in o(n^2 \sqrt{n})$
J) $f(n) \in \omega(g(n))$ *se e somente se* existem as constantes $c \in R^+$ e $n_0 \in Z^+$ tal que, $f(n) > c.g(n)$, para todo $n \geq n_0$.

- ○ A, F e I
- ○ C, E, F, I e J
- ◉ A, C, E e F
- ○ A, C, F e I
- ○ A, C, F, I e J

Google Forms