Projetos e Análise de Algoritmos

Prova 1

Data: 09/04/2025
Duração: 100 minutos
Valor: 20 pontos

1) **Utilizando as definições formais** para as notações assintóticas, **prove** se são verdadeiras ou falsas as afirmativas abaixo:

   a) $n^3 + 2^n + n \log n = O(n^3)$
   b) $5n^2 + n (\log n)^2 \neq \Theta(n^2)$
   c) $n^2 = \omega (n (\log n)^2)$
   d) $\frac{1}{2} n^2 + n\log n = \Omega(n^2)$
   e) $2^{n+4} = \Theta(2^n)$

2) Considere o problema de encontrar o 2o maior elemento de um vetor com n elementos. Escreva um pseudo-código para resolver este problema. **Prove** que seu algoritmo está correto usando invariantes. Qual a **função de complexidade** do seu algoritmo? **Elabore sua resposta**.

3) Ordene as funções abaixo em **ordem crescente de complexidade assintótica:**

$n (\log n)^2$ – 2^{3n} – $n^2(\log n)^2$ – $n^{2/4}$ – $n\sqrt{n}$ – 4^{n+3} – $2^n(\log n)$ – n^2 – $(\log n)^3$

4) Indique uma equivalência assintótica (Θ) para cada uma das funções recursivas apresentadas abaixo. Em todos os casos, considere que os casos bases das recursões são todos iguais a 1. Utilize para cada letra o método indicado. Apresente cálculos ou justifique sua resposta.

   a) $T(n) = 7T(n/2) + n^2$, utilizando o Teorema Mestre
   b) $T(n) = 7T(n/4) + \sqrt{n}$, utilizando o Teorema Mestre
   c) $T(n) = 3T(n/2) + cn$, utilizando o método da expansão de termos.
   d) $T(n) = T(n-1) + \log n$, utilizando qualquer técnica dada em sala (especifique a técnica escolhida na sua resposta)

5) Seja o seguinte algoritmo (em pseudo-código) que determina o elemento máximo de um vetor A com n elementos:

Variável X_i: 1 se A[i] > max, 0 caso contrário

Indica se a comparação retorna verdadeiro e se a linha 4 é executada.

```
Maximo (vetor A, inteiro n)
(1)   max = -∞
(2)   para i = 1 até n faça
(3)         se A[i] > max então
(4)               max = A[i]
(5)   retorne max
```

$$E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i] = \sum_{i=1}^{n} \frac{1}{i} \approx \ln n - C_1$$

Considere que todos os elementos de A sejam distintos e derive o **número médio** de vezes que a linha (4) será executada.
Dica: considere uso de variáveis indicadoras aleatórias. O que cada variável X_i representa?

Obs.: A linha (4) executa se o rank i for maior que o rank do max atual. Supondo distribuição uniforme, ocorre com probabilidade $\frac{1}{i}$.

INFORMAÇÕES ÚTEIS

1. Teorema Mestre: Sejam $a \geq 1$ e $b > 1$ constantes, $f(n)$ uma função e

$$T(n) = aT(\frac{n}{b}) + f(n),$$

Então, para algum $\epsilon > 0$

Se $f(n) = O(n^{\log_b a - \epsilon})$ $T(n) = \Theta(n^{\log_b a})$
Se $f(n) = \Theta(n^{\log_b a})$ $T(n) = \Theta(n^{\log_b a} \log n)$
Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ e $af(\frac{n}{b}) \leq cf(n)$ então $T(n) = \Theta(f(n))$

2. $\sum_{i=1}^{n} \frac{1}{i} \approx \ln n$

Nome: Thales Henrique Silva

10/12

(1)

a) Falso. A função não pode ser limitada por $O(n^3)$ porque o termo 2^n cresce mais rápido que n^3:

$$\lim_{n\to\infty} \frac{2^n}{n^3} = \infty \text{ (o numerador ganha)}$$

Logo, não existem constantes tal que $2^n < c_1 n^3$ para $n > n_0$.

b) Falso. A função é $\Theta(n^2)$:

$$c_1 n^2 \leq 5n^2 + n(\log n)^2 \leq c_2 n^2$$
$$c_1 n^2 - 5n^2 \leq n(\log n)^2 \leq c_2 n^2 - 5n^2$$
$$(c_1 - 5) n^2 \leq n(\log n)^2 \leq (c_2 - 5) n^2$$

O lado esquerdo se satisfaz com $c_1 = 4$, supondo n positivo. Para o lado direito, $c_2 = 6$:

$$n(\log n)^2 \leq (c_2 - 5) n^2$$
$$(\log n)^2 \leq (c_2 - 5) n$$

A função linear ganha do logaritmo, mesmo que este tenha um expoente maior.

5/5

c) Verdadeiro.

$$\lim_{n\to\infty} \frac{n^2}{n(\log n)^2} = \frac{(n)'}{((\log n)^2)'} = \frac{1}{2(\log n)\cdot\frac{1}{n}} = \frac{(n)'}{(2\log n)'} = \frac{1}{2\cdot\frac{1}{n}} = n \to \infty$$

n^2 cresce mais rápido que $n(\log n)^2$ tendendo ao infinito.

d) Verdadeira.

$$c_1 . n^2 \le \frac{1}{2} n^2 + n \log n$$

$$(c_1 - 1/2) n^2 \le n \log n$$

Fazendo $c_1 = 1/2$, o lado esquerdo se cancela e o direito será sempre maior ou igual.

e) Verdadeira.

$$c_1 2^n \le 2^{n+4} \le c_2 2^n$$

$$c_1 2^n \le 2^4 \cdot 2^n \le c_2 2^n$$

$$c_1 \le 16 \le c_2$$

Basta usar $c_1 = c_2 = 16$.

2) Encontra2oMaior(vale vetor A[1..n]):

```
Se n == 2:
    retorna (min(A[1], A[2]), max(A[1], A[2]))  // tupla
Senão:
    (a, b) = Encontra2oMaior(A[2..n])  // tupla (a, b)
    Se A[1] > b:
        retorna (b, A[1])
    Se A[1] > a:
        retorna (A[1], b)
    Senão
        retorna (a, b)
```

2/4

O algoritmo retorna uma tupla (a, b) em que o 2o maior elemento é o a. Vamos provar por indução. O caso base n = 2 funciona, pois a tupla retornada contém o min e o max, e logo o 2o maior só pode ser o min. Suponha que o código funcione para n elementos.

② Observe o que acontece com uma lista de $n+1$ elementos. O elemento $A[0]$ é comparado com o 2o maior e o maior de uma lista com n elementos (supondo a hipótese indutiva). Se o elemento $A[0]$ é maior que o max b, então b é 2o maior e $A[0]$ é o maior: retorna $(b, A[0])$. Mas se $A[0] > a$, $A[0]$ se torna o 2o maior: retorna $(A[0], b)$. Por fim, se $A[0] \leq a \leq b$, então nada muda: retorna (a, b). Veja que o 2o maior elemento sempre está na primeira posição da tupla e o algoritmo está correto. Sua função de complexidade obedece à recorrência

$T(n) = T(n-1) + c$, $T(2) = 1$

Essa é uma clássica recorrência $O(n)$, pois fazemos n chamadas de custo constante.

③ $(\log n)^2 < n^{3/4} < n(\log n)^2 < n\sqrt{n} < n^2 < n^2(\log n)^2 <$
$< 2^n(\log n) < 4^{n+3} < 2^{3n}$

④

a) $T(n) = 7T(n/2) + n^2$

$a = 7,\ b = 2 \Rightarrow n^2 = O(n^{\log_2 7 - \varepsilon})$ $\hookrightarrow \log_2 7 > 2$

Caso 1: $T(n)$ é $\Theta(n^{\log_2 7})$

b) $T(n) = 7T(n/4) + n^{0.5}$ $\hookrightarrow \log_4 7 > 0.5$

$a = 7,\ b = 4 \Rightarrow n^{0.5} = O(n^{\log_4 7 - \varepsilon})$

Caso 1: $T(n)$ é $\Theta(n^{\log_4 7})$

(4)

c) $T(n) = 3T(n/2) + cn$

$T(n) = 3[3T(n/4) + cn/2] + cn$

$T(n) = 3[3[3T(n/8) + cn/4]] + 3c\frac{n}{2} + cn$

$T(n) = 3^3 T(n/8) + 3^2 c\frac{n}{4} + 3\frac{cn}{2} + cn$

O padrão é:

$nc \sum_{i=0}^{k} \left(\frac{3}{2}\right)^i$, em que $k = \log_2 n$, $T(1) = c$ por conveniência

$nc \cdot \frac{a_1(q^{k+1} - 1)}{q - 1} = nc \cdot \frac{1(3/2^{k+1} - 1)}{0.5} = 2nc \cdot \left(\frac{3}{2}^{\log_2 n + 1} - 1\right) =$

$= nc\, 3^{\log_2 n + 1} - 2nc \approx \Theta(n^{\log_2 3})$ ignorando constantes

d) $T(n) = T(n-1) + \log n$ (expansão)

$T(n-1) = T(n-2) + \log(n-1)$

$T(n-2) = T(n-3) + \log(n-2)$

$\vdots$

$T(1) = 1$

soma de logaritmos

$T(n) = 1 + \sum_{i=1}^{n-1} \log(n-i) = \overbrace{\log(n) + \log(n-1) + \ldots + \log(1)}^{\text{soma de logaritmos}}$

$T(n)$ é $\Theta(n \log n)$

Questão 5

4/4

Resolução no enunciado