**Pontifícia Universidade Católica de Minas Gerais**
**Bacharelado em Ciência da Computação**
**Projeto e Análise de Algoritmos – 1º Semestre de 2019**
**Prof[a]. Raquel Mini**

# 1ª Prova

**Nome:**______________________________________________________________________

**Data: 01/04/2019**

**Valor: 35 pontos**

**1.** (8 pontos) Considerando que 0 < ε < 1 < c são constantes, prove se as seguintes afirmativas são verdadeiras ou falsas. Respostas sem justificativas não serão pontuadas.

a) $2^{n+1} = O(2^n)$

b) $2^{2n} = O(2^n)$

c) $n^2 = \Theta(n^3)$

d) $\dfrac{2n^3}{\log n + 1} = O(n^3)$

e) Se $f(n) = \Omega(g(n))$ então $f(n) = \omega(g(n))$

f) Se $T(n) = 9T\left(\dfrac{n}{3}\right) + n$ então $T(n) = \Theta(n^2)$

g) $\log_2 n = \Theta(\sqrt{\log_2 n})$

h) $c^{\varepsilon} = O(c+1)^{\varepsilon}$

**2.** (10 pontos) Considere o seguinte algoritmo recursivo, cujo parâmetro *n* é um inteiro positivo.

```
ASTERISCO(n)
   se n > 0
      então ASTERISCO(n − 1)
            para i ← 1 até n faça
                imprima "*"
            ASTERISCO(n − 1)
```

a) Considerando que a operação relevante seja o comando "imprima "*"", apresente a equação de recorrência para esse algoritmo.

b) Resolva a equação de recorrência, apresente a função de complexidade e sua ordem de complexidade utilizando a notação Θ (não precisa prova a notação Θ).

c) Para um dado valor de *n*, quantos asteriscos serão impressos em uma chamada ASTERISCO(*n*)?

**3.** (6 pontos) Utilize o teorema mestre, se possível, para resolver as recorrências abaixo. Caso não seja possível, explique o porquê.

a) $T(n) = 9T\left(\dfrac{n}{3}\right) + n$  b) $T(n) = 2T\left(\dfrac{5n}{2}\right) + n - 1$  c) $T(n) = 2T\left(\dfrac{n}{2}\right) + n\log n$

**4.** (11 pontos) Considerando que no algoritmo abaixo o vetor A é global e que a operação relevante seja o número comparações com os elementos do vetor A, responda às seguintes perguntas.

```
void OQUEEUFACO(int i, int j, int &m2, int &m1) {
   int m21,m22, m11, m12, mm;
   if ((j - i) <= 1) {
      if (A[i] < A[j]) {
         m2 = A[j];
         m1 = A[i];
      } else {
         m2 = A[i];
         m1 = A[j];
      }
   }
   else {
      mm = (i + j)/2;
      OQUEEUFACO(i,mm,m21,m11);
      OQUEEUFACO(mm+1,j,m22,m12);
      if (m21 > m22)
         m2 = m21;
      else
         m2 = m22;
      if(m11 < m12)
         m1 = m11;
      else
         m1 = m12;
   }
}
```

a) O que o algoritmo faz?
b) Qual é a situação que leva ao melhor caso e ao pior caso desse algoritmo?
c) Escreva a equação de recorrência que descreva o seu comportamento e indique a operação considerada relevante.
d) Converta esta equação de recorrência para um somatório.
e) Forneça a fórmula fechada para este somatório.

## Teorema Mestre

- Sejam as constantes $a \geq 1$ e $b > 1$ e $f(n)$ uma função definida nos inteiros não-negativos pela recorrência:

$$T(n) = aT(n/b) + f(n)$$

onde a fração $n/b$ pode significar $\lfloor n/b \rfloor$ ou $\lceil n/b \rceil$. A equação de recorrência $\mathrm{T}(n)$ pode ser limitada assintoticamente da seguinte forma:

1. Se $f(n) = \mathrm{O}(n^{\log_b a - \varepsilon})$ para alguma constante $\varepsilon > 0$, então $\boxed{\mathrm{T}(n) = \Theta(n^{\log_b a})}$
2. Se $f(n) = \Theta(n^{\log_b a})$, então $\boxed{\mathrm{T}(n) = \Theta(n^{\log_b a} \log n)}$
3. Se $f(n) = \Omega(n^{\log_b a + \varepsilon})$ para alguma constante $\varepsilon > 0$ e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e para $n$ suficientemente grande, então $\boxed{\mathrm{T}(n) = \Theta(f(n))}$

**Fórmulas:**

| | | | |
|---|---|---|---|
| $\log n = \log_2^n$ | $a^{\log_b^n} = n^{\log_b^a}$ | $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ | $2^{\log n} = n$ |
| $a = b^{\log_b^a}$ | $n^{\frac{1}{\log n}} = n^{\log_n^2} = 2$ | $\sum_{i=0}^{n} i^3 = \frac{n^2(n+1)^2}{4}$ | $\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ |
| $\sum_{i=1}^{n} 1 = n$ | $\sum_{i=1}^{n} ia^i = \frac{a-(n+1)a^{n+1}+na^{n+2}}{(a-1)^2}$ | $\sum_{i=0}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$ | $4^{\log n} = 2^{2\log n} = 2^{\log n^2} = n^2$ |
| $1 \prec \log\log n \prec \log n \prec n^{\varepsilon} \prec n \prec n^{c} \prec n^{\log n} \prec c^{n} \prec n^{n} \prec c^{c^{n}}$ <br> onde ε e $c$ são constantes arbitrárias com $0 < \varepsilon < 1 < c$ | | | |