

Universidade Estadual da Paraíba Centro de Ciência e Tecnologia Departamento de Computação Bacharelado em Ciência da Computação
Componente Curricular: <b>Técnicas de Análise de Algoritmos</b>
Semestre:
Professor: Fábio Luiz Leite Jr
Aluno (a):

### **Lista de exercícios sobre Análise de algoritmos recursivos**

1. Considere o algoritmo que calcula os números de fibonacci de um dado número  $n$  de entrada.
  - a. Faça um algoritmo iterativo para implementar o fibonacci
  - b. Analise o tempo de execução do algoritmo implementado em a)
  - c. Faça um algoritmo recursivo para implementa o fibonacci
  - d. Analise o tempo de execução do algoritmo implementado em b)

$$F(n) = \begin{cases} 0, & \text{se } n = 0; \\ 1, & \text{se } n = 1; \\ F(n-1) + F(n-2) & \text{outros casos.} \end{cases}$$

2. Mostre, pelo método da substituição, que o algoritmo merge sort é  $\Theta(n \lg n)$ .
3. Mostre, pelo método da substituição, que a solução de  $T(n) = 2T\left(\frac{n}{2} + 17\right) + n$  é  $O(n \lg n)$ .
4. Mostre, pelo método da substituição, que a solução de  $T(n) = T(n-1) + n$  é  $O(n^2)$ .
5. Resolva a recorrência  $T(n) = 4T\left(\frac{n}{2}\right) + n^2$  utilizando o teorema mestre.
6. Resolva a recorrência  $T(n) = 2T\left(\frac{n}{2}\right) + n^4$  utilizando o teorema mestre.
7. Resolva a recorrência  $T(n) = 7T\left(\frac{n}{3}\right) + n^2$  utilizando o teorema mestre.
8. Resolva pelo método da expansão a recorrência  $T(n) = 8T\left(\frac{n}{2}\right) + n^2$  ( $T(1) = 1$ )
9. Suponha que, para entradas de tamanho  $n$ , você tenha de escolher um entre dois algoritmos.
  - a. O algoritmo A resolve problemas dividindo-os em dois subproblemas de tamanho  $n-1$ , recursivamente resolve cada subproblema e então combina as soluções em tempo  $O(1)$ .
  - b. Algoritmo B resolve problemas dividindo-os em nove subproblemas de tamanho  $n/3$ , recursivamente resolve cada subproblema e então combina as soluções em tempo  $O(n^2)$ .
  - c. Qual o consumo de tempo desses algoritmos? Qual algoritmo é assintoticamente mais eficiente?

## 10. Questões do Poscomp:

**QUESTÃO 21** – Suponha que, ao invés de dividir em duas partes, foi criada uma versão do merge-sort que divida a entrada em quatro partes, ordene cada quarta-parte, e, finalmente, combine essas quatro partes usando um procedimento  $O(n)$ . A equação de recorrência que descreve o tempo de execução desse algoritmo é:

- A)  $T(n) = 4 \cdot T(n/4) + O(n)$
- B)  $T(n) = 4 \cdot T(n/2) + 2 \cdot O(n)$
- C)  $T(n) = T(n/4) + 4 \cdot O(n)$
- D)  $T(n) = 4 \cdot T(n/4) + 4 \cdot O(n)$
- E)  $T(n) = T(n/4) + O(n)$

**QUESTÃO 21** – Um algoritmo tem complexidade  $O(3m^3 + 2mn^2 + n^2 + 10m + m^2)$ . Uma maneira simplificada de representar a complexidade desse algoritmo é:

- A)  $O(m^3 + mn^2)$ .
- B)  $O(m^3)$ .
- C)  $O(m^2)$ .
- D)  $O(mn^2)$ .
- E)  $O(m^3 + n^2)$ .

---

**QUESTÃO 22** – O tempo de execução  $T(n)$  de um algoritmo, em que  $n$  é o tamanho da entrada, é dado pela equação de recorrência  $T(n) = 8T(n/2) + q \cdot n$  se  $n > 1$ . Dado que  $T(1) = p$ , e que  $p$  e  $q$  são constantes arbitrárias, a complexidade do algoritmo é:

- A)  $O(n)$ .
- B)  $O(n \log n)$ .
- C)  $O(n^2)$ .
- D)  $O(n^3)$ .
- E)  $O(n^n)$ .