

PANC: Projeto e Análise de Algoritmos

Aula 09: Divisão e Conquista - Subarranjo Máximo e Multiplicação de Matrizes

Lista de Exercício – Listex 08

Breno Lisi Romano

<http://sites.google.com/site/blromano>

Instituto Federal de São Paulo – IFSP São João da Boa Vista
Bacharelado em Ciência da Computação – 3º Semestre



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus São João da Boa Vista



Instruções Gerais para a Listex

■ Instruções:

- Todos os exercícios desta Listex devem ser realizados **individualmente**
- Estimula-se a **discussão** com outros **colegas** de **turma** para o **desenvolvimento**, mas **difícilmente soluções individualizadas** serão iguais → **Trabalho Idênticos serão penalizados**: Oportunidade de Aprender Errando
- Todos os **exercícios** desta Listex precisam ser **entregues na Plataforma Google Classroom**, na **atividade** criada, seguindo a data e hora definidos como **prazo de entrega**
- Quando os **exercícios envolverem programação**, **compactem o projeto com os arquivos** para solução dos exercícios, diferenciando cada um deles, o seguinte **padrão de nomes**:
 - Modelo: Listex01-Exercício01-NomeSobrenomeAluno.zip
 - Exemplo: Listex01-Exercício01-BrenoRomano.zip
- Quando os **exercícios envolverem pesquisar, textos escritos, manipulações matemáticas ou outros casos semelhantes**, **entreguem** o exercício em um arquivo na **extensão .PDF**, seguindo o **padrão de nomes**:
 - Modelo: Listex01-Exercício01-NomeSobrenomeAluno.pdf
 - Exemplo: Listex01-Exercício01-BrenoRomano.pdf



Trabalhos para Casa (1)

■ Exercício 01 – Subarranjo Máximo:

- Execute o algoritmo FindMaxSubarray() para a instância:
 - $A = [20; -7; 12; -5; -22; 15; -4; 7]$
- Resolva a recorrência e determine a complexidade de FindMaxSubarray() aplicando-se o Teorema Mestre
- Responda: O que FindMaxSubarray() retorna quando todos os elementos de A são negativos?
- Justifique por que o algoritmo recursivo FindMaxSubarray() não faz sentido ser aplicado quando todos os elementos do array A são positivos?
- Explicar o funcionamento do Algoritmo Linear proposto por Jay Kadane e realizar a Análise da Complexidade, mostrando que o mesmo é $O(n)$

```
SEG-MAX-1 (A, n)
1  somamax ← 0
2  e ← 0    d ← -1    ▷ A[e..d] é vazio
3  i ← 1
4  soma ← 0
5  para f ← 1 até n faça
6      se soma + A[f] < 0
7          então i ← f + 1    soma ← 0
8          senão soma ← soma + A[f]
9      se soma > somamax então
10         somamax ← soma    e ← i    d ← f
11  devolva e, d e somamax
```

Algoritmo Proposto
por Jay Kadane



Trabalhos para Casa (2)

Exercício 02 – Multiplicação de Matrizes:

- a. Use o algoritmo de Strassen para calcular o produto de matrizes

$$\begin{pmatrix} 1 & 3 \\ 7 & 5 \end{pmatrix} \begin{pmatrix} 6 & 8 \\ 4 & 2 \end{pmatrix}$$

- b. Aplicar o Teorema Mestre para Resolver a Recorrência do Algoritmo de D&C de Multiplicação de Matrizes

$$T(n) = \begin{cases} \theta(1) & \text{se } n = 1, \\ 8T(n/2) + \theta(n^2) & \text{se } n > 1 \end{cases}$$

- c. Aplicar o Teorema Mestre para Resolver a Recorrência do Algoritmo de Strassen de Multiplicação de Matrizes

$$T(n) = \begin{cases} \Theta(1) & \text{se } n = 1; \\ 7T(n/2) + \Theta(n^2) & \text{se } n > 1. \end{cases}$$

- d. **DESAFIO (Opcional):** Implementar os 03 algoritmos de multiplicação de matrizes apresentados, em qualquer linguagem de programação:

- Algoritmo Tradicional
- Algoritmo de Divisão e Conquista
- Algoritmo de Strassen

```
"D:\Google Drive\AULAS-BRENO\IFSP\Acadêmico\Disciplinas\PANC (Superior - BC...
Multiplicacao de Matrizes: Algoritmo de Strassen
Entre com o tamanho da matriz: 3
Entre com os elementos da Primeira Matriz, por linha:
2 2 2
3 3 3
4 4 4
Entre com os elementos da Segunda Matriz, por linha:
1 2 3
4 5 6
7 8 9
Resultado da Multiplicacao:
24 30 36
36 45 54
48 60 72
Process returned 0 (0x0)   execution time : 14.354 s
Press any key to continue.
```



Trabalhos para Casa (3)

- **Exercício 03 – Pesquisa: O Problema da Multiplicação de Números Inteiros – Algoritmo de Karatsuba:**
 - Problema: Dados números inteiros positivos u e v muito grandes, com n dígitos cada, calcular o produto $u \cdot v$
 - Exemplo 01: $99998888 \cdot 77776666$
 - Exemplo 02: $99998888 \cdot 00076666$
 - a. Estudar, na literatura, sobre o problema de multiplicação de números inteiros grandes, apresentando a lógica proposta pelo Algoritmo de Karatsuba
 - b. Apresentar um pseudocódigo para o Algoritmo de Karatsuba
 - c. Apresentar a Análise da Complexidade do Algoritmo de Karatsuba (Fórmula de Recorrência e big-O)
 - d. Aplicar o algoritmo de Karatsuba para a multiplicação dos números:
 - $99998888 \cdot 77776666$
 - e. Existem outros algoritmos de multiplicação de números inteiros muito grande mais que o algoritmo de Karatsuba?