

PANC: Projeto e Análise de Algoritmos

Aula 06: Algoritmos Recursivos: Problemas e Análise da Complexidade

Lista de Exercício – Listex 05

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 – Para os problemas apresentados em Sala de Aula, implementar, em projetos distintos, a solução para resolvê-los, na Linguagem C:**
 - a. Implemente a solução recursiva para resolução do Fatorial de um número n fornecido pelo usuário
 - b. Implemente a solução iterativa para resolução do Fatorial de um número n fornecido pelo usuário
 - c. Implemente a solução recursiva para resolver o problema da Torre de Hanoi, solicitando a quantidade n de discos a ser aplicado
 - d. Implemente a solução recursiva para resolução da Busca Binária, solicitando ao usuário o Array ordenado e o valor a ser procurado
 - e. Implemente a solução recursiva para resolução do Fibonacci de um número n fornecido pelo usuário
 - f. Implemente a solução iterativa para resolução do Fibonacci de um número n fornecido pelo usuário



Trabalhos para Casa (2)

- **Exercício 02 – Para os problemas apresentados abaixo, implementar, em projetos distintos, a solução para resolvê-los, na Linguagem C. Além disso, determinar a Recorrência associada a Recursão – $T(n)$:**
 - a. Implemente uma função recursiva para calcular 2^n , para um n fornecido pelo usuário
 - b. Implemente uma função recursiva para calcular o Máximo Divisor Comum (MDC) de dois números (a e b) fornecidos pelo usuário (Teorema de Euclides)
 - c. Implemente recursivamente uma função Max que retorne o maior valor armazenado em um array, contendo n números inteiros fornecidos pelo usuário
 - d. Implemente uma função recursiva para calcular o valor de n -ésimo número harmônico da Série Harmônica $H(n) = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$, para um valor de n fornecido pelo usuário e $n \geq 1$. Destaca-se que $H(1) = 1$ e $H(n) = H(n-1) + 1/n$



Trabalhos para Casa (3)

- Exercício 03 – Considere a função $\text{Comb}(n, k)$, que representa o número de grupos distintos com k pessoas que podem ser formados a partir de n pessoas. Por exemplo, $\text{Comb}(4,3) = 4$, pois com 4 pessoas (A, B, C, D), é possível formar 04 grupos distintos: ABC, ABD, ACD e BCD. Sabe-se que:

$$\text{Comb}(n, k) = \begin{cases} n & \text{se } k = 1 \\ 1 & \text{se } k = n \\ \text{Comb}(n-1, k-1) + \text{Comb}(n-1, k) & \text{se } 1 < k < n \end{cases}$$

- Implemente um projeto na linguagem C que utilize uma função recursiva $\text{Comb}(n, k)$, retornando a quantidade de grupos k que podem ser formados para n pessoas.

```
C:\Users\bv120133\Desktop\Aula06-Ex03-Comb\bin\D...
Entre com o valor de grupos distintos (k): 3
Entre com o valor de pessoas (n): 4
A quantidade de combinacoes C(n, k) = C(4, 3) = 4
Process returned 0 (0x0)   execution time : 3.847 s
Press any key to continue.
```