



Universidade de São Paulo – USP
Instituto de Matemática e Estatística – IME
Tópicos de Programação – Verão 2025 (EAD Noturno)
Prof. M.Sc. Rodrigo Hagstrom

Lista de Exercícios – Aula 18

Tentar fazer a lista e trazer para a Aula 19, amanhã!!!

Exercício 1: Implementação do Pseudocódigo em uma Linguagem de Programação

Implemente o **pseudocódigo** de DFS recursivo em **Python, Java ou C++**. Crie uma árvore binária e percorra-a usando as três variações da DFS:

- **Pré-ordem (Root, Left, Right)**
- **Em ordem (Left, Root, Right)**
- **Pós-ordem (Left, Right, Root)**

Requisitos:

- Criar uma estrutura para representar os nós da árvore.
- Implementar a DFS recursiva para as três variações.
- Testar o código com uma árvore de pelo menos **5 nós**.

Exercício 2: Análise de Complexidade

Calcule a complexidade de tempo e espaço da **DFS recursiva** aplicada a uma árvore binária.

Perguntas:

1. Qual é a complexidade de tempo no **melhor, pior e caso médio**?
2. Qual é a complexidade de espaço, considerando o uso da pilha de recursão?

Exercício 3: Comparação entre DFS Recursiva e Iterativa

Reimplemente a DFS de forma **iterativa** usando uma **pilha explícita** (stack).

Perguntas:

1. Qual das duas abordagens (recursiva ou iterativa) é mais eficiente em termos de **uso de memória**?
2. Em quais cenários a **DFS iterativa** é preferível à recursiva?
3. Como garantir que a DFS iterativa funcione corretamente sem **estouro de pilha** para árvores muito profundas?