



Pontifícia Universidade Católica de Minas Gerais  
Instituto de Ciências Exatas e Informática  
Departamento de Ciência da Computação  
Disciplina: Algoritmos e Estruturas de Dados II

**PUC Minas**

# Trabalho Prático VII

(Data de Entrega: 28/11)

---

## Regras Gerais

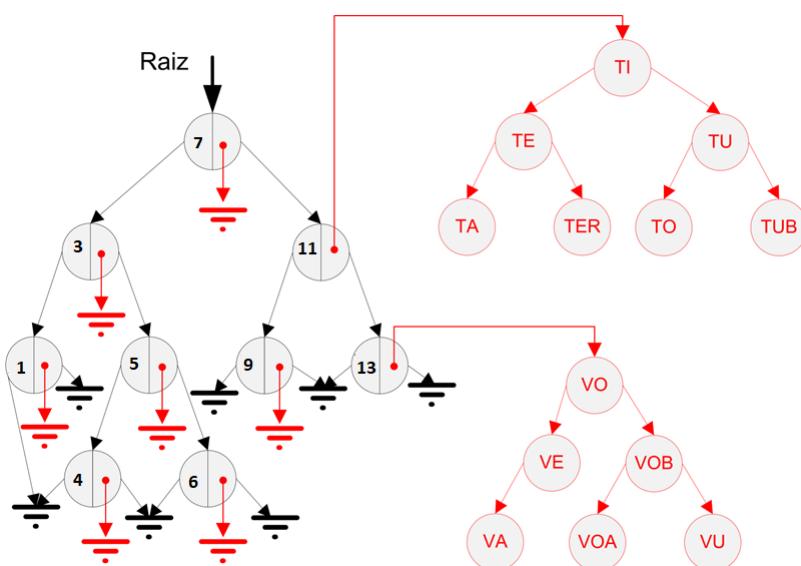
- A partir deste trabalho, você deve utilizar as soluções em **Java** ou **C**, implementadas no TP-06.
- Para cada questão, deve-se submeter apenas um arquivo (.java ou .c). Essa regra será necessária para a submissão de trabalhos no Verde e no identificador de plágio utilizado na disciplina.
- Para cada exercício: faça (análise, implementação e comentários), teste (várias vezes) e submeta no Verde. Os exercícios não testados/comentados serão penalizados.
- A correção será realizada automaticamente pelo sistema Verde. Entretanto, você poderá ser entrevistado durante as aulas de laboratório e deverá demonstrar pleno domínio sobre o código desenvolvido. A utilização de ferramentas de IA generativa na elaboração do trabalho implicará em nota zero.

## Questões

1. **Árvore Binária em Java:** Crie uma Árvore Binária, fazendo inserções de registros conforme a entrada padrão. A entrada padrão é composta por duas partes onde a primeira é igual a entrada do PESQUISA BINÁRIA do TP-05. A chave de pesquisa é o atributo NAME. Não insira um elemento se sua chave estiver na árvore. Em seguida, pesquise se alguns registros estão cadastrados na Árvore, mostrando seus respectivos caminhos de pesquisa. A saída padrão é composta por várias linhas, uma para cada pesquisa. Cada linha é composta pelo caminho ou sequência de ponteiros (**raiz**, **esq** ou **dir**) utilizados na pesquisa e, no final, pelas palavras SIM ou NAO. Além disso, crie um arquivo de log na pasta corrente com o nome matrícula\_arvoreBinaria.txt com uma única linha contendo sua matrícula, tempo de execução do seu algoritmo e número de comparações realizadas durante as pesquisas.

2.  **Árvore Binária de Árvore Binárias em Java:** Refaça a questão anterior, contudo, considerando a estrutura de árvore de árvore. Nessa estrutura, temos uma árvore binária tradicional na qual cada nó tem um ponteiro para outra árvore binária. Graficamente, a primeira árvore está no plano xy e a árvore de seus nós pode ser imaginada no espaço tridimensional. Temos dois tipos de nós. O primeiro tem um número inteiro como chave, os ponteiros ESQ e DIR (ambos para nós do primeiro tipo) e um ponteiro para nós do segundo tipo. O outro nó tem uma String como chave e os ponteiros ESQ e DIR (ambos para nós do segundo tipo). A chave de pesquisa da primeira árvore é o atributo **estimatedOwners mod 15** e, da outra, é o atributo **Name**. Conforme a figura abaixo.

Destaca-se que nossa pesquisa faz um “mostrar” na primeira árvore e um “mostrar” na segunda. Faremos um “mostrar” na primeira árvore porque ela é organizada pelo **estimatedOwners mod 15**, permitindo que o valor desejado esteja na segunda árvore de qualquer um de seus nós. Faremos o “mostrar” na segunda porque ela é organizada pelo atributo **Name**. Antes de inserir qualquer elemento, crie a primeira árvore, inserindo todos seus nós e respeitando a ordem **7, 3, 11, 1, 5, 9, 13, 0, 2, 4, 6, 8, 10, 12 e 14**. O arquivo de log será **matrícula\_arvoreArvore.txt**.



3.  **Árvore AVL em C:** Refaça a primeira questão deste trabalho com ÁRVORE AVL. O nome do arquivo de log será **matrícula\_avl.txt**.
4.  **Árvore Alvinegra em Java:** Refaça a primeira questão deste trabalho com ÁRVORE ALVINEGRA. O nome do arquivo de log será **matrícula\_avinegra.txt**.