# Unidade V:
# Árvore Binária: Estruturas Híbridas

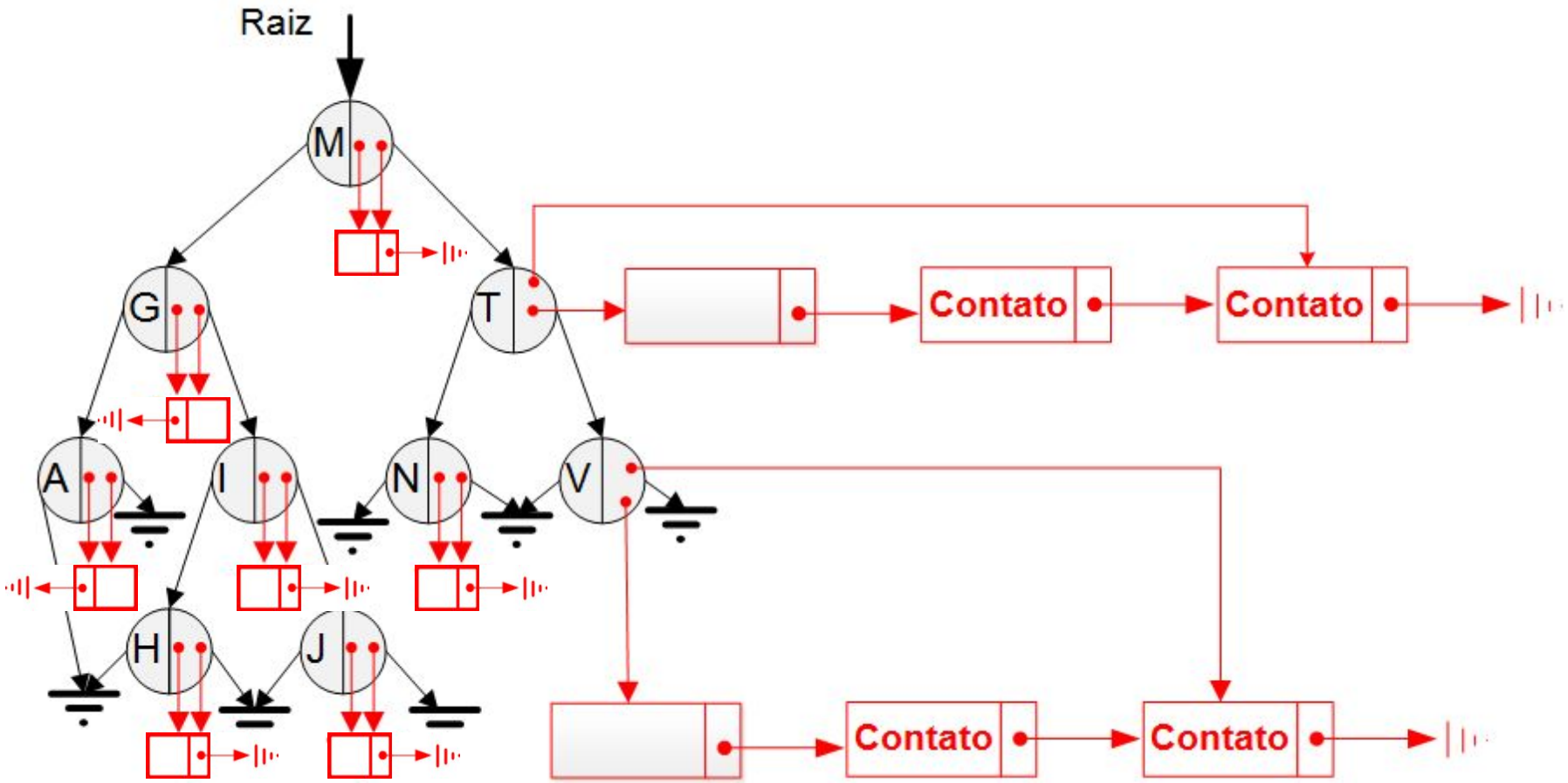

Instituto de Ciências Exatas e Informática
Departamento de Ciência da Computação

# Exercício (1)

- Você foi contratado para desenvolver uma agenda de contatos (atributos nome, telefone, email e CPF) para um escritório de advocacia

# Exercício (1)

- Um colega sugeriu implementar uma árvore de binária de listas em que a pesquisa na árvore acontece pela primeira letra do nome e, quando encontramos a letra, temos uma pesquisa em uma lista de contatos



# Exercício (1)

- Crie uma classe Contato contendo os atributos String nome, telefone e e-mail e int CPF

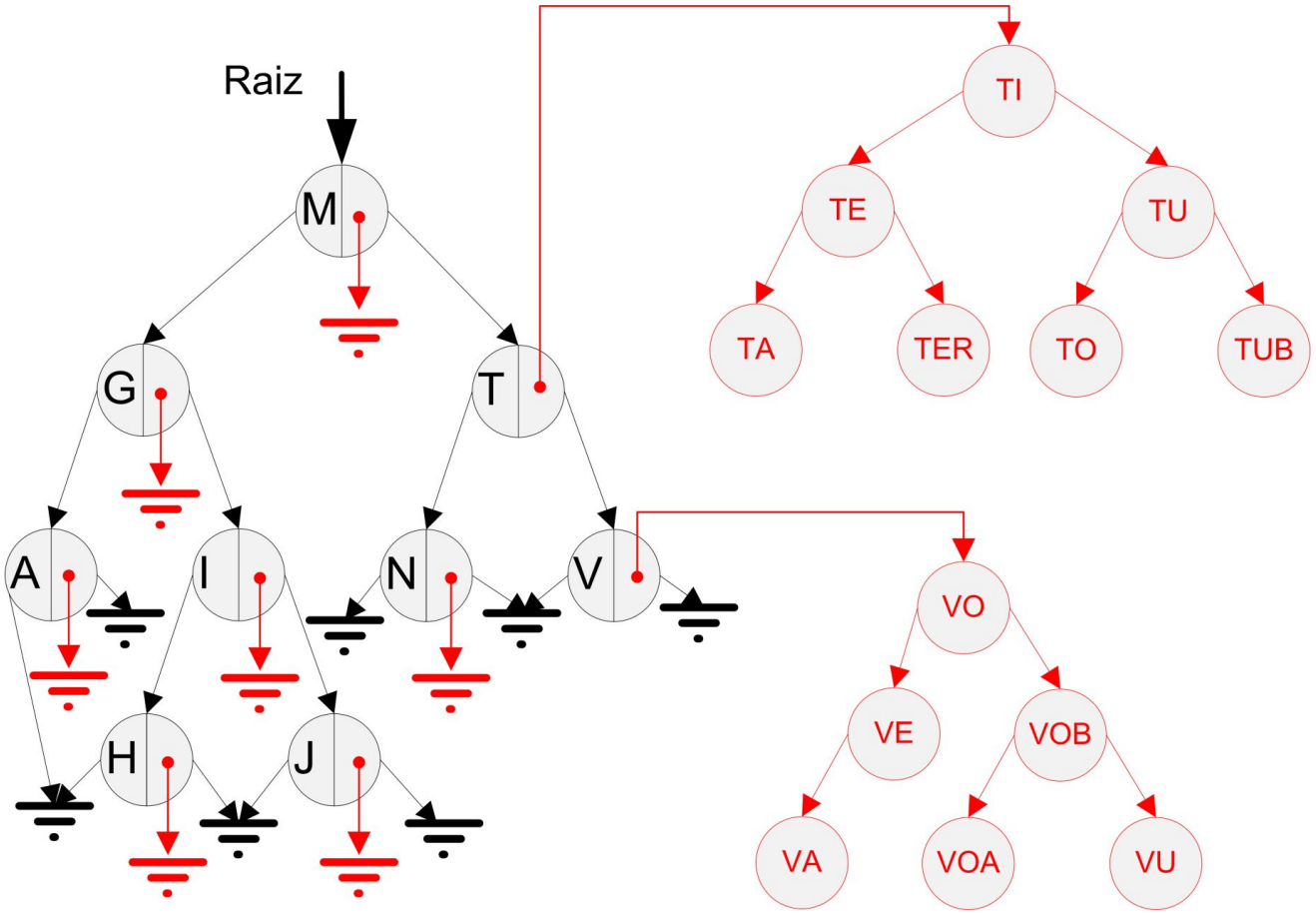
- Crie uma classe Celula contendo os atributos Contato contato e Celula prox

- Crie uma classe No contendo os atributos char letra, No esq e dir; e Celula primeiro e ultimo

- Crie uma classe Agenda contendo o atributo No raiz, os métodos inserir(Contato contato), remover(String nome), pesquisar(String nome) e pesquisar(int cpf). No construtor, insira todas as letras (Nós): somente letras maiúsculas e sem acento. Para cada método, mostre o melhor e pior caso

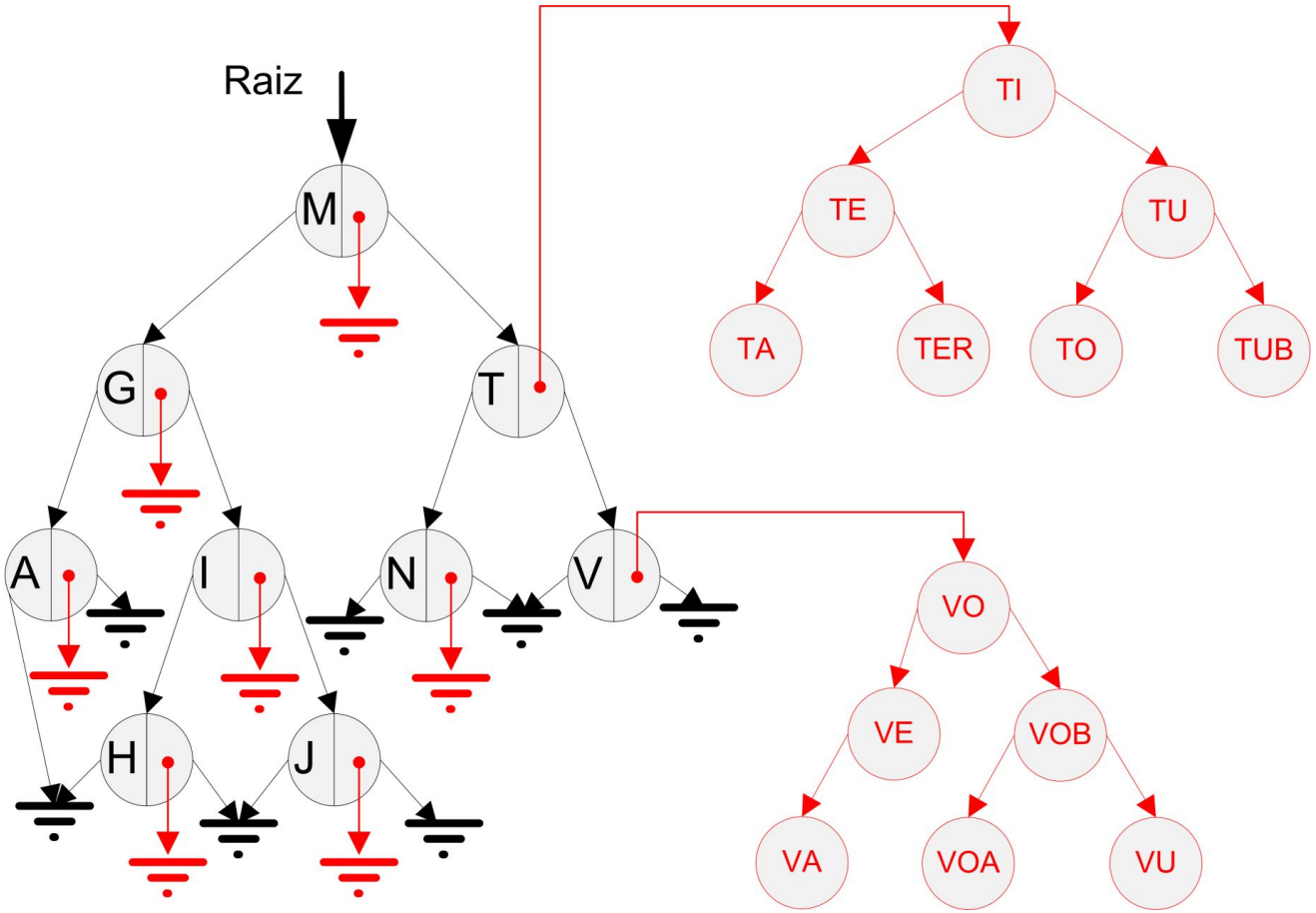
# Exercício (2)

- Implemente os métodos ***boolean pesquisar(String nome)***, ***void inserir(String nome)*** e ***void mostrar()*** para a estrutura abaixo:



# Exercício (3)

- Na árvore de árvore, implemente o método ***boolean hasStringTam10()***, que retorna true se tivermos uma string de tamanho 10:



# Exercício (4)

- Na árvore de árvore, implemente o método ***boolean hasStringTam10(char c)***, que retorna true se tivermos uma string de tamanho 10 e que começa com o caractere c:

# Exercício (5)

- Implemente os métodos **boolean pesquisar(int elemento)**, **boolean pesquisar(int i, int j, int elemento)**, **void inserir(int i, int j, int elemento)**, **void mostrar()** para a estrutura abaixo: