



## Trabalho Prático IV

---

### Regras Básicas

- extends Trabalho Prático 03
- Fique atento ao Charset dos arquivos de entrada e saída.

**Observação:**

**Nas questões de árvore, utilizamos o mostrar pré.**

**Não será necessário implementar a opção de remoção nas TADs abaixo.**

Harry Potter é uma série de sete romances de fantasia escrita pela autora britânica J.K. Rowling. A narrativa centraliza-se em Harry Potter, um jovem órfão que descobre, no seu décimo primeiro aniversário, que é um bruxo. Ele é convidado a estudar na Escola de Magia e Bruxaria de Hogwarts, onde aprende a prática da magia sob a orientação do gentil diretor Albus Dumbledore e de outros professores da escola. Harry descobre que é famoso no mundo dos bruxos por ter sobrevivido a um ataque letal do poderoso e maligno bruxo Lord Voldemort quando era apenas um bebê. Voldemort matou os pais de Harry, mas por algum motivo, sua maldição mortal não funcionou em Harry.



Ao longo dos livros, Harry é acompanhado por seus melhores amigos, Ronald Weasley e Hermione Granger. Juntos, eles enfrentam diversos desafios, incluindo o retorno de Voldemort, que busca não apenas conquistar o mundo bruxo, mas também destruir Harry, o único que pode impedir seus planos maléficos.

A série alcançou uma imensa popularidade, aclamação da crítica e sucesso comercial em todo o mundo. Além dos livros, a história de Harry Potter foi adaptada para uma série de filmes de sucesso,

peças de teatro, jogos e uma vasta gama de produtos. Tornou-se um significativo fenômeno cultural e uma das séries de livros mais vendidas da história.

O arquivo **characters.csv** contém um conjunto de dados de personagens da série extraídos do site <https://www.kaggle.com/>. Essa base contém registros de 405 personagens. Este arquivo **sofreu algumas adaptações** para ser utilizado neste e nos próximos trabalhos práticos. Tal arquivo deve ser copiado para a pasta /tmp/. **Quando reiniciamos o Linux, ele normalmente apaga os arquivos existentes na pasta /tmp/.**

## Árvores

**Observação: ATENÇÃO para os algoritmos de árvore que já estão implementados no [Github!](#)**

1. **Árvore Binária em Java:** Crie uma Árvore Binária, fazendo inserções de registros conforme a entrada padrão. 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 entrada padrão é igual a da questão de “Pesquisa Sequencial”. 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. Todas as informações do arquivo de log devem ser separadas por uma tabulação '\t'.
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 **yearOfBirth 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 **yearOfBirth 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.

