



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.

POKÉMON é uma franquia de mídia criada pela Nintendo, Game Freak e Creatures, que começou como uma série de jogos de RPG para o Game Boy em 1996. Criada por Satoshi Tajiri, a série POKÉMON rapidamente se tornou um fenômeno global, evoluindo para uma das franquias mais populares e lucrativas do mundo.

A premissa básica dos jogos gira em torno dos POKÉMON (Pocket Monsters), criaturas fictícias que os jogadores, conhecidos como Treinadores, capturam e treinam para lutar contra outros POKÉMON. Cada jogo geralmente começa com o jogador recebendo seu primeiro POKÉMON de um professor e então viajando por várias regiões para capturar novos POKÉMON, desafiar líderes de ginásio, e eventualmente competir na Liga POKÉMON.

Um elemento central dos jogos é a Pokédex, uma enciclopédia eletrônica que registra informações sobre todos os POKÉMON encontrados ou capturados. A Pokédex fornece detalhes como espécie, altura, peso, e uma descrição única para cada POKÉMON. O objetivo de completar a Pokédex, capturando todos os POKÉMON disponíveis, é uma das principais motivações dos treinadores.

Os POKÉMON possuem individualidades marcantes, como tipos e habilidades específicas. Existem vários tipos, como Fogo, Água, Planta, Elétrico, Psíquico, Dragão, entre outros. Esses tipos determinam as fraquezas e resistências de um POKÉMON em batalhas, criando uma dinâmica estratégica.



Por exemplo, um POKÉMON do tipo Fogo é forte contra POKÉMON do tipo Planta, mas fraco contra POKÉMON do tipo Água.

Além dos tipos, cada POKÉMON tem habilidades especiais que conferem vantagens em batalha. Por exemplo, a habilidade “Levitate” torna um POKÉMON imune a ataques do tipo Terra, enquanto “Intimidate” reduz o poder de ataque do oponente ao início de uma batalha.

Dentro do universo POKÉMON, também existem os POKÉMON lendários, que são raros, poderosos e fundamentais para a história dos jogos. Esses POKÉMON são únicos, com habilidades e estatísticas superiores em comparação com os POKÉMON comuns. Exemplos de POKÉMON lendários incluem Mewtwo, Zapdos, e Rayquaza, cada um com seu próprio lore e importância dentro da série.

POKÉMON se tornou um fenômeno cultural, com impacto significativo em várias gerações. Os jogos não apenas entretêm, mas também promovem valores como amizade, estratégia e a importância de cuidar de outras criaturas. A série de jogos POKÉMON continua a crescer, lançando novos títulos regularmente e mantendo sua relevância na cultura pop global.

O arquivo POKEMON.CSV contém um conjunto de dados dos pokémons do jogo extraídos do site [Kaggle](#). Essa base contém 801 pokémons. 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/.

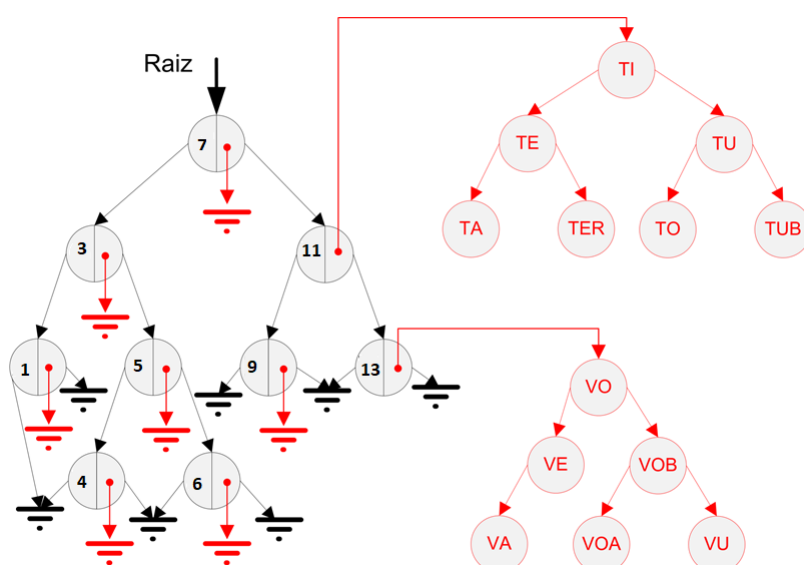
Árvores

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 **captureRate 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 **captureRate 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 em C. 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.
5. **Tabela Hash Direta com Reserva:** Refaça a primeira questão deste trabalho com Tabela Hash Direta com Reserva. A função de transformação será **(ASCII name) mod tamTab** onde tamTab (tamanho da tabela) é 21. A área de reserva tem tamanho 9, fazendo com que o tamanho total da tabela seja igual a 30. A saída padrão será a posição de cada elemento procurado na tabela (na *hash* ou na área de reserva). Se o elemento procurado não estiver na tabela, escreva a palavra NÃO. Além disso, o nome do arquivo de log será matrícula_hashReserva.txt.
6. **Tabela Hash Direta com Rehash:** Refaça a questão anterior com Tabela Hash Direta com Rehash. A primeira função de transformação será **(ASCII name) mod tamTab** onde tamTab (tamanho da tabela) é 21 e a outra, **(ASCII name + 1) mod tamTab**. O nome do arquivo de log será matrícula_hashRehash.txt.

7. Tabela *Hash* Indireta com Lista Simples em C: Refaça a questão anterior com Tabela *Hash* Indireta com Lista Simples. A função de transformação será **(ASCII name) mod tamTab** onde tamTab (tamanho da tabela) é 21. O nome do arquivo de log será matricula_hashIndireta.txt.