Pontifícia Universidade Católica de Minas Gerais
Curso de Ciência da Computação
Disciplina: Algoritmos e Estruturas de Dados II
Prof. Felipe Domingos da Cunha / Prof. Max do Val Machado

Trabalho Prático VII - Árvores e *Hash*

1 Regras Básicas

1. extends TP5RegrasBasicas;

2 Descrição

1. **Árvore Binária**: Crie uma Árvore Binária, fazendo inserções de objetos conforme a entrada padrão. A chave de pesquisa é o atributo **id**. Não insira um elemento se sua chave estiver na árvore. Em seguida, pesquise se alguns objetos estão cadastrados na Árvore, mostrando seus respectivos caminhos de pesquisa. A entrada padrão contém três partes. As duas primeiras são iguais as duas partes da questão "TP03Q03 - Pilha Sequencial" do Trabalho Prático III, contudo, no caso dos comandos de remoção, após a letra R, temos um valor indicando a chave de um objeto a ser removido. A terceira parte contém vários valores, um por linha, indicando a chave de objetos a serem pesquisados. 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 NÃO. 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**: Crie uma árvore de árvore. Nesse caso, 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 um int como chave e os ponteiros esq e dir (ambos para nós do segundo tipo). A chave de pesquisa da primeira árvore é o atributo **códigoUF** e, da outra, é o atributo **id**. Nesta questão, vamos inserir e remover alguns objetos e, em seguida, pesquisar a existência

de alguns **ids**. Destaca-se que nossa pesquisa faz um "mostrar" na primeira árvore e um "pesquisar" na segunda. Faremos um "mostrar" na primeira árvore porque ela é organizada pelo **códigoUF**, permitindo que o valor desejado esteja na segunda árvore de qualquer um de seus nós. Faremos o "pesquisar" na segunda porque ela é organizada pelo atributo **id**. Antes de inserir qualquer elemento, crie a primeira árvore, inserindo todos seus nós e respeitando a ordem 29, 21, 50, 14, 25, 35, 52, 12, 16, 23, 27, 32, 42, 51, 53, 11, 13, 15, 17, 22, 24, 26, 28, 31, 33, 41 e 43. A entrada e a saída padrão são iguais as da questão anterior e o arquivo de log será matrícula_arvoreArvore.txt. O "mostrar" da **primeira** árvore deve ser o central.

3. **Árvore AVL**: Refaça a primeira questão deste trabalho com Árvore AVL. O nome do arquivo de log será matrícula_avl.txt. Não insira um elemento se sua chave estiver na árvore.

4. **Árvore Alvinegra**: Refaça a primeira questão deste trabalho com Árvore Alvinegra. O nome do arquivo de log será matrícula_avinegra.txt. Não insira um elemento se sua chave estiver na árvore. **Não será necessário implementar a opção de remoção.**

5. **Tabela *Hash* Direta com Reserva**: Refaça a questão "TP04Q01 - Pesquisa Sequencial" do Trabalho Prático IV com Tabela *Hash* Direta com Reserva, fazendo com que a função de transformação seja **id 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 entrada padrão será igual as demais deste trabalho. A saída será a posição de cada elemento procurado na tabela. Se ele estiver na área de reserva considere que o tamanho total da tabela é a área da *hash* mais a de reserva. Caso o elemento procurado não esteja na tabela, escreva a palavra NÃO. Além disso, o nome do arquivo de log será matrícula_hashReserva.txt. **Não será necessário implementar a opção de remoção.**

6. **Tabela *Hash* Direta com Rehash**: Refazer a questão anterior com Tabela *Hash* Direta com *Rehash*, fazendo com que a função de rehash seja **(id mais um) mod tamTab** onde tamTab é 21. A entrada e saída são como na questão anterior. O nome do arquivo de log será matrícula_hashRehash.txt. **Não será necessário implementar a opção de remoção.**

7. **Tabela *Hash* Indireta com Lista Simples**: Refazer a questão anterior com Tabela *Hash* Indireta com Lista Simples, fazendo com que a função de transformação seja **id mod tamTab** onde tamTab é igual a 21 e corresponde ao tamanho da tabela. A entrada e saída são como na questão anterior. O nome do arquivo de log será matrícula_hashIndireta.txt.