



Trabalho Prático III

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

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

1 Árvores e *Hash*

1. **Árvore Binária:** Crie uma Árvore Binária, fazendo inserções de registros conforme a entrada padrão. A chave de pesquisa é o atributo **nome**. 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 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:** 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 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 **idade mod 15** e, da outra, é o atributo **idade**.

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 **idade 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 **idade**. Antes de inserir qualquer

elemento, crie a primeira árvore, inserindo todos seus nós e respeitando a ordem 7, 3, 11, 1, 5, 9, 12, 0, 2, 4, 6, 8, 10, 13 e 14. O arquivo de log será matrícula_arvoreArvore.txt.

3. **Árvore AVL:** Refaça a primeira questão deste trabalho com Árvore AVL. O nome do arquivo de log será matrícula_avl.txt.
4. **Árvore Alvinegra:** 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 **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 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 **id mod tamTab** onde tamTab (tamanho da tabela) é 21 e a outra, **(id + 1) mod tamTab**. O nome do arquivo de log será matrícula_hashRehash.txt.
7. **Tabela *Hash* Indireta com Lista Simples:** Refaça a questão anterior com Tabela *Hash* Indireta com Lista Simples. A função de transformação será **id mod tamTab** onde tamTab (tamanho da tabela) é 21. O nome do arquivo de log será matrícula_hashIndireta.txt.