

Algoritmos e Estruturas de Dados III
2016 – 1
Descrição do Primeiro Trabalho Prático

Objetivo

Implementar e comparar algoritmos para melhorar o desempenho de operações em árvores binárias de busca de acordo com o padrão de acesso.

Descrição

Você deverá utilizar o código fornecido pelo professor, consistindo dos arquivos `cliente.c`, `Item.c`, `Item.h`, `Dicionario.c` e `Dicionario.h`.

Você deverá apresentar três modificações dos arquivos `Dicionario.c` e `Dicionario.h`, cada uma correspondente à uma implementação diferente do TAD Dicionário. As implementações são descritas abaixo. O cabeçalho das funções disponíveis em `Dicionario.h` não deverá ser modificado.

Implementação 1

Você modificará os arquivos `Dicionario.c` e `Dicionario.h` de modo que o Dicionário seja implementado por meio de uma **árvore AVL**.

Implementação 2

Você modificará os arquivos `Dicionario.c` e `Dicionario.h` de modo que o Dicionário seja implementado por meio de uma árvore binária de busca comum, com a seguinte alteração no algoritmo de Busca:

Suponha que a chave c está sendo buscada

Se não houver na árvore um nó com chave c , então a busca será encerrada normalmente.

Se a chave c estiver na árvore, suponha que esta chave se encontra no nó N .

Se N for a raiz da árvore, então a busca será encerrada normalmente.

Se N não for a raiz da árvore, denote por P o seu pai.

Se N estiver à esquerda de P , deverá ser efetuada uma rotação à direita em P . Se N estiver à direita de P , deverá ser efetuada uma rotação à esquerda em P .

Implementação 3

Você modificará os arquivos `Dicionario.c` e `Dicionario.h` de modo que o Dicionário seja implementado por meio de uma árvore binária de busca comum, com a seguinte alteração no algoritmo de Busca:

Suponha que a chave c está sendo buscada

Se não houver na árvore um nó com chave c , então a busca será encerrada normalmente.

Se a chave c estiver na árvore, suponha que esta chave se encontra no nó N .

Deverá haver em cada nó da árvore um contador cujo valor representa o número de vezes em que aquele nó foi encontrado como o resultado de uma busca.

Se N for a raiz da árvore, então a busca será encerrada normalmente.

Se N não for a raiz da árvore, denote por P o seu pai.

Enquanto P não for nulo e o número de acessos à P for menor que o número de acessos à N :

Se N estiver à esquerda de P , deverá ser efetuada uma rotação à direita em P . Se N estiver à direita de P , deverá ser efetuada uma rotação à esquerda em P .

Seja P o novo pai de N .

Note que o objetivo das implementações 2 e 3 é tentar melhorar a eficiência do sistema manipulando a árvore de modo que nós acessados mais vezes ou mais recentemente apareçam mais próximos da raiz da árvore. Será dado um **bônus de 10%** ao grupo que sugerir uma estratégia não

proposta por nenhum outro grupo, que tenha o objetivo de melhorar a eficiência da árvore.

Execução

O programa será executado por meio de comandos. Estarão disponíveis os seguintes comandos

INSERIR <chave>

REMOVER <chave>

BUSCAR <chave>

IMPRIMIR

Os comandos INSERIR e REMOVER não deverão exibir nenhuma mensagem. O comando Buscar deverá exibir a mensagem “encontrado” caso a chave seja encontrada, e a mensagem “nao encontrado” caso a chave não seja encontrada.

O comando IMPRIMIR deverá imprimir as chaves dos nós da árvore em percurso **pré-ordem**.

A entrada de dados será tratada no arquivo `Cliente.c`, e será implementada pelo professor. Portanto, os únicos arquivos que serão manipulados pelos alunos no trabalho serão os arquivos `Dicionario.h` e `Dicionario.c`.

Entrega

Deverão ser entregue três modificações dos arquivos `Dicionario.c` e `Dicionario.h`. Estas versões deverão ser nomeadas:

`Dicionario1.c` e `Dicionario1.h` para a Implementação 1.

`Dicionario2.c` e `Dicionario2.h` para a Implementação 2.

`Dicionario3.c` e `Dicionario2.h` para a Implementação 3.

Além do código fonte, deverá ser entregue um relatório. O relatório deverá obrigatoriamente conter:

- Uma descrição de todas as estruturas de dados utilizadas e funções implementadas.
- Uma descrição EM ALTO NÍVEL (isto é, sem código) explicando a lógica do programa.

O trabalho deverá ser implementado utilizando a linguagem de programação C e ser passível de compilação em Linux.

O trabalho deverá ser feito em grupos de no máximo 3 alunos.

A data de entrega é 03/07/2016, até às 23:55.