

TRABALHO FINAL

Comparação do Desempenho de Árvores Binárias

1 Objetivo

Este trabalho tem por objetivo o desenvolvimento de uma análise comparativa do desempenho de diferentes tipos de árvores binárias (balanceadas e não balanceadas).

2 Especificação

O objetivo do trabalho é comparar o desempenho das Árvores Binárias de Pesquisa e de um tipo de árvore balanceada a ser escolhida pela dupla (AVL, Rubro-Negra, ou Splay) em diversas funções.

Ambas as árvores irão armazenar números inteiros que devem ser processados na ordem em que aparecerem no arquivo de entrada.

As seguintes funções devem ser implementadas para ambos os tipos de árvores:

- **Inserir (I)** – para inserir um número inteiro na árvore
- **Consultar (C)** – para consultar se um determinado valor existe na árvore
- **Remover (R)** – para remover um número inteiro na árvore

Além disso, funções auxiliares para calcular diversas **estatísticas (E)** sobre a árvore devem ser criadas.

- **ContaNodos** – retorna o número de nodos da árvore (serve para checar se todos os números foram corretamente inseridos/removidos).
- **Altura** – retorna a altura da árvore.
- **Fator** – retorna o fator de balanceamento da árvore (i.e., o maior dos fatores de balanceamento encontrados na árvore).
- É preciso também contar o **tempo gasto** (em milissegundos) e o **número de comparações** (i.e., o número de vezes que um valor de chave é comparado com o valor do nó corrente na função analisada – inserção/consulta/remoção) e **rotações** (número de vezes que uma operação de rotação é chamada) efetuadas para processar o total de valores em um arquivo.

3 Relatório

É necessário elaborar um relatório **detalhado** com a análise comparativa do desempenho em termos de tempo de processamento, número de comparações, fator de balanceamento, altura, etc. das duas árvores em cada uma das funções implementadas.

Utilize recursos como tabelas e gráficos para dar suporte às suas conclusões.

O relatório deve analisar qual árvore é mais eficiente para cada uma das operações e tipo de entrada e explicar os motivos.

Além disso, avalie como cada árvore se comporta à medida que o número de valores a serem inseridos/removidos/consultados aumenta.

4 Requisitos

- A aplicação deve ser chamada **a partir da linha de comando** (passando parâmetros para o main). Por exemplo, o comando:

C:\minhaaplicacao roteiro.txt

significa que é necessário processar as operações especificadas no arquivo roteiro.txt. Este arquivo será formado por comandos (representados pelas suas siglas, seguida dos nomes dos arquivos a serem processados, quando apropriado).

- Os arquivos com os valores a serem inseridos/consultados/removidos serão compostos apenas pelos números inteiros (um por linha).
- Serão disponibilizados arquivos para teste da aplicação. Para a apresentação, um novo conjunto de arquivos será utilizado.
- O trabalho deve ser feito, preferencialmente, em duplas. A linguagem de programação aceita é C (Não é C++ nem C#).

5 Exemplo de funcionamento

Entrada: roteiro.txt

```
I 100000.txt
E
C n10000.txt
E
R n1000.txt
E
```

Inserir na árvore os números que estão no arquivo n10000.txt
Imprimir estatísticas
Consultar os números que estão no arquivo n10000.txt
Imprimir estatísticas
Remover da árvore os números que estão no arquivo n1000.txt
Imprimir estatísticas

Saída (na tela ou redirecionada para arquivo) correspondente ao arquivo roteiro.txt acima.

```
*****VERSÃO COM ABP*****
Inserindo dados do arquivo n100000.txt
*****ESTATÍSTICAS ABP*****
Tempo: 323 ms
Nodos: 100000
Altura:480
Fator: 464
Comparações: 16893479
Rotações: 0
-----
Consultando dados do arquivo n10000.txt
*****ESTATÍSTICAS ABP*****
Tempo: 26 ms
Nodos: 100000
Altura:480
Fator: 464
Comparações: 104156
Rotações: 0
-----
Removendo dados do arquivo n1000.txt
*****ESTATÍSTICAS ABP*****
Tempo: 2 ms
Nodos: 99000
Altura:480
Fator: 464
Comparações: 14911
Rotações: 0
-----
```

Tempo gasto na inserção de todos os valores do arquivo

Lembre de resetar a contagem do tempo

Será sempre zero para as ABPs

6. Entrega e Apresentação

- **7 de dezembro 2016** apresentação (horário da aula) e envio pelo Moodle

7. Critérios de Avaliação

O trabalho deve ser realizado em duplas e deverá ser apresentado e defendido na data prevista.

Para a avaliação serão adotados diversos critérios:

- funcionamento (Peso: 30%),
- organização e documentação do código (Peso: 30%).
- relatório (Peso: 40%);

Importante:

É permitido reusar funções dos slides ou encontradas na internet (sempre referenciando a fonte), mas será necessário demonstrar conhecimento detalhado sobre o seu funcionamento.

Este trabalho deverá representar a solução da dupla para o problema proposto. O plágio é terminantemente proibido e a sua detecção incorrerá na divisão da nota obtida pelo número de alunos envolvidos. Para detectar o plágio, usaremos o software MOSS (<http://theory.stanford.edu/~aiken/moss/>).