Terceiro Projeto – Árvore Vermelho-Preta

Algoritmos e Estruturas de Dados I

Responsável: Prof. Fernando Vieira Paulovich
Aluna PAE: Glenda Botelho

25 de novembro de 2010

1. Descrição do Trabalho

Devido sua eficiência, uma das estruturas de dados mais utilizadas para buscas são as *Árvores Binárias (AB)*. Contudo, a eficiência de uma AB está intimamente ligada a sua altura, sendo que não é qualquer árvore que apresenta estrutura que leva a buscas eficientes. Para que uma AB seja eficiente, essa deve ser balanceada, ou aproximadamente balanceada. Isso porque uma árvore aproximadamente balanceada terá altura **c * $\log_2(n)$**, onde **c** é uma constante pequena e **n** é o número de elementos na árvore. Isso leva a complexidade computacional da ordem $O(\log_2 n)$, reduzindo bastante o número de acessos a serem feitos para a localização de um determinado elemento.

Entre as árvores ditas aproximadamente balanceadas, as *Árvores Vermelho-Pretas* são bem conhecidas, com constante **c** igual a **2**, portanto, sendo, no máximo, duas vezes mais altas do que as árvores perfeitamente balanceadas.

Nesse trabalho você deve implementar os métodos de **INSERÇÂO** e **REMOÇÂO** de *Árvores Vermelho-Pretas.* Uma boa fonte informação sobre esse tipo de estrutura é o livro **CORMEN, T.H.; LEISERSON, C.E.; RIVEST, R.L.; STEIN, C. Algoritmos: Teoria e Prática. Editora Campus.2002.**

2. Formato de Entrada e Saída

A entrada será composta por pares de elementos, separados por espaço, em linhas distintas (e consecutivas). O primeiro elemento indica a operação, **I** para inserção ou **R** para remoção, e o segundo elemento é a chave inserida (o valor associado à chave deve ser igual à chave). Por exemplo:

I 5
I 8
I 3
I 10
R 3
I 7

Representa a inserção das chaves **5, 3, 8** e **10** seguida pela remoção da chave **3**, e posterior inserção da chave **7**.

Observação Importante: quando um nó com sub-árvores a esquerda e a direita é removido existem duas possíveis ações envolvidas nesse processo de remoção (vide material de aula) que seriam trocar esse nó com o maior nó da sub-árvore esquerda, ou com o menor nó da sub-árvore direita. Nesse trabalho considere que a troca ocorrerá com o **maior nó da sub-árvore esquerda**.

As entradas devem ser lidas da entrada padrão (p.ex. usando o **scanf(...)**) e a árvore resultante de cada operação deve ser impressa na saída padrão (p. ex. usando **printf(...)**).

A árvore deve ser escrita na saída na forma de **chaves aninhadas**, imprimindo a árvore resultante de cada operação aplicada. Por exemplo, a saída referente a entrada apresentada seria:

{5}
{5,{8}}
{5,{3},{8}}
{5,{3},{8,{10}}}
{8,{5},{10}}
{8,{5,{7}},{10}}

O programa deve ser implementado em apenas um arquivo (.c).

O tempo máximo para apresentar a solução para qualquer entrada será de **10 segundos**.

2. Correção

O trabalho deve ser submetido ao sistema SQTPM (http://infoserver.lcad.icmc.usp.br/cgi-bin/glenda/alg1/sqtpm.pl) para correção automática.

3. Grupo

Grupo deve ser formado por no **máximo 2 alunos**. O nome e número USP dos alunos do grupo deve estar no início do arquivo submetido (como um comentário)

4. Entrega

Entrega dia **8 de Dezembro**. Entregas fora do prazo não serão aceitas. O sistema SQTPM estará no ar 24 horas antes do prazo final de entrega. **Mas façam antes o projeto!**

Em caso de dúvidas entre em contato com a estagiária PAE pelo e-mail: glenda@icmc.usp.br