

Universidade Federal da Bahia  
Graduação em Ciência da Computação  
MATA54 - Estruturas de Dados e Algoritmos II  
Primeiro Trabalho Prático  
Prof. Flávio Assis  
Semestre 2013.1 - 26 de junho de 2013

Árvore B+

## 1 Descrição Geral do Trabalho

Neste trabalho o aluno implementará uma árvore B+, de acordo com a descrição do livro de A.Tharp, *File Organization and Processing*. De acordo com a definição neste livro, cada nó da parte de índice da árvore conterá, com exceção da raiz, no mínimo  $d$  chaves e no máximo  $2d$  chaves, onde  $d$  é a *ordem* da árvore.

Cada registro da árvore terá os seguintes dados: uma chave, de valor inteiro não negativo; uma cadeia de, no máximo, 20 caracteres, que irá armazenar um nome; e um outro valor inteiro não negativo, que irá armazenar uma idade. O programa deverá conter duas constantes:

- ORDEM: ordem da parte de índice da árvore;
- REGS\_POR\_PAGINA: número de registros por página de nós terminais.

O programa deve ser feito de forma que os valores destas constantes possam ser modificados.

## 2 Formato de Entrada e Saída

A entrada constará de uma sequência de operações sobre a árvore. As operações e seus formatos estão descritos abaixo:

1. **insere registro:** esta operação conterá quatro linhas. A primeira linha conterá a letra 'i'. A segunda conterá um valor de chave. A terceira conterá uma sequência de até 20 caracteres. A quarta linha conterá um valor de idade. A sequência de caracteres da terceira linha conterá qualquer sequência de letras (minúsculas, sem acento, nem cedilha) e espaços, sendo que o primeiro e último caracteres não serão espaço.

Esta operação verifica se já há registro na árvore com o valor de chave indicado. Se sim, esta operação gera na saída, em uma mesma linha, a sequência de caracteres '*chave ja existente:*', seguida de um espaço, seguido do valor da chave. Se a chave não existir, a operação insere o registro na árvore, sem gerar saída.

2. **consulta registro:** esta operação conterà duas linhas. A primeira linha conterà a letra 'c'. A segunda conterà um valor de chave.

Se houver registro na árvore com o valor de chave indicado, esta operação gera na saída a sequência de caracteres '*chave:*', seguida de um espaço, seguido no valor da chave. Em seguida, na próxima linha escreve o valor do nome associado ao registro, e, na linha seguinte, o valor da idade associada ao registro. Se não houver registro na árvore com o valor de chave indicado, esta operação gera na saída a sequência de caracteres '*chave nao encontrada:*', seguida de um espaço, seguido do valor da chave.

3. **remove registro:** esta operação conterà duas linhas. A primeira linha conterà a letra 'r'. A segunda conterà um valor de chave.

Se houver registro na árvore com o valor de chave indicado, esta operação causará a remoção do registro e não gerará saída. Se não houver registro na árvore com o valor de chave indicado, esta operação gera na saída a sequência de caracteres '*chave nao encontrada:*', seguida de um espaço, seguido do valor da chave.

4. **imprime árvore:** esta operação conterà apenas uma linha, contendo a letra 'p'. O formato da saída desta operação está indicado na seção 3.

5. **imprime nós terminais:** esta operação conterà apenas uma linha, contendo a letra 'f'.

Se a árvore estiver vazia, esta operação não gera saída. Caso contrário, para cada nó, esta operação gera na saída: a sequência de caracteres '*No:*', seguida de um espaço, seguido de um número sequencial para o nó (o primeiro nó é o nó número 1, o segundo é o nó 2, etc.). Em seguida, a operação escreve os dados de cada registro armazenado no nó: em uma linha escreve o valor da chave, na linha seguinte, o valor do nome, e, na linha seguinte, o valor da idade.

As chaves dos nós devem aparecer em ordem crescente.

6. **término da sequência de comandos:** a sequência de comandos será terminada por uma linha com a letra 'e'.

### 3 Formato de Impressão da Árvore

A impressão da árvore será feita seguindo um percurso em largura. Cada nó receberá um número sequencial, seguindo a ordem do percurso (a raiz é o nó 1). A impressão de cada nó será feita em uma linha. Não deve haver espaço entre linhas.

A impressão de cada nó seguirá o seguinte formato. Primeiramente será escrita a sequência de caracteres '*No:*', seguida de um espaço, seguido do número sequencial do nó, seguido do caractere ':' (dois pontos), seguido de um espaço. Em seguida, na mesma linha, será impresso o conteúdo do nó.

Se o nó for um nó não-terminal (parte de índice), o nó será impresso da seguinte maneira. Os apontadores e chaves devem ser impressas seguindo a estrutura do nó. Cada apontador deve ser impresso da seguinte maneira: a sequência de caracteres '*apontador:*'; seguida de um espaço, seguido do número sequencial do nó para o qual o apontador aponta. Cada chave será impressa da seguinte maneira: a sequência de caracteres '*chave:*'; seguida de um espaço, seguido do valor da chave. As impressões de apontadores e chaves devem estar separadas por um espaço.

Se o nó for um nó terminal, a impressão de seu conteúdo será apenas a impressão das chaves armazenadas no nó. Cada chave será impressa como indicado no parágrafo acima.

**Importante:** o programa não deve gerar nenhum caractere a mais na saída, além dos indicados acima. Em particular, o programa não deve conter menus.

Não deve haver espaço entre linhas na saída. A saída deve apresentar os caracteres em letras minúsculas.

## 4 Observações

Trabalho individual.

Data de entrega: 26/07/2013

Linguagens de programação permitidas: C, C++, Java ou Python.

**Observação Importante:** Para as linguagens C, C++ e Java, somente trabalhos feitos utilizando-se os seguintes compiladores serão aceitos:

- C: gcc ou djgpp
- C++: g++ ou djgpp
- Java: compilador java do JDK (mais recente)

**Não serão compilados trabalhos em outros compiladores! Erros ocasionados por uso de diferentes compiladores serão considerados erros do trabalho!**

O aluno deverá armazenar seu trabalho, protegido por senha, na página da disciplina e enviar a senha para o email do professor (o aluno deve utilizar *zip*).