

Universidade Federal da Bahia
Graduação em Ciência da Computação
MATA54 - Estruturas de Dados e Algoritmos II
Primeiro Trabalho Prático
Prof. Flávio Assis
Semestre 2013.2 - 26 de novembro de 2013

Hashing com Árvore Binária

1 Descrição Geral do Trabalho

Neste trabalho o aluno implementará um arquivo estruturado como uma tabela *hash* em que o método de resolução de colisão *Árvore Binária* é utilizado.

Cada registro do arquivo 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 uma constante:

- TAMANHO_ARQUIVO: indica o número máximo de registros do arquivo.

O valor inicial desta constante deve ser 11. O programa deve ser feito de forma que este valor possa ser modificado.

A função de *hashing* a ser utilizada, denominada h , é:

$$h(chave) = chave \bmod \text{TAMANHO_ARQUIVO}$$

Observação importante: o programa deve manter as atualizações em arquivo. A correção levará em consideração que o estado dos dados é persistente. Com isto, um teste pode ser feito, por exemplo, inserindo-se um registro, terminando a execução do programa e fazendo uma consulta ao registro em nova invocação do programa. Neste caso o registro deve ainda estar no arquivo.

O arquivo deve ser armazenado em formato binário.

2 Formato de Entrada e Saída

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

1. **insere registro:** esta operação conterà quatro linhas. A primeira linha conterà a letra 'i'. A segunda conterà um valor de chave. A terceira conterà uma sequência de até 20 caracteres. A quarta linha conterà um valor de idade. A sequência de caracteres da terceira linha conterà 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 no arquivo 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 no arquivo, sem gerar saída.

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

Se houver registro no arquivo com o valor de chave indicado, esta operação gera na saída a sequência de caracteres '*chave:*', seguida de um espaço, seguido do 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 no arquivo 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 conterá duas linhas. A primeira linha conterá a letra 'r'. A segunda conterá um valor de chave.

Se houver registro no arquivo com o valor de chave indicado, esta operação causará a remoção do registro e não gerará saída. Se não houver registro no arquivo 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 arquivo:** esta operação conterá apenas uma linha, contendo a letra 'p'. Esta operação imprimirá o formato do arquivo, da forma a seguir. Os registros serão apresentados, um em cada linha, em ordem, do registro de índice 0 até o registro de índice *TAMANHO_ARQUIVO* - 1. Cada linha conterá: o índice do registro, seguido de dois pontos (':'), seguido de um espaço. Se o registro estiver vazio, a sequência de caracteres '*vazio*' deverá ser apresentada. Se o registro contiver dados, deve ser apresentada a chave do registro, seguida de um espaço, seguida da sequência de caracteres (nome), seguida de um espaço, seguido da idade. Se o registro estiver marcado como apagado, deve ser apresentado o símbolo '*'.

5. **sequência de tentativa de inserção:** esta operação conterá duas linhas. A primeira conterá a letra 't'. A segunda conterá um valor de chave.

Esta operação lista, uma por linha, as tentativas de inserção de uma chave, *SEM INSERIR A CHAVE*. Garantidamente, *não haverá registro no arquivo com valor de chave igual ao indicado na operação*. As tentativas devem ser listadas na ordem gerada pelo método de árvore binária. Cada tentativa deve ser apresentada no seguinte formato: o valor do índice do registro consultado, seguido por dois pontos (':'), seguido de um espaço. Em seguida, se o registro consultado estiver ocupado, o programa deve

apresentar a chave do registro. Caso contrário, deve apresentar a sequência de caracteres *'vazio'*.

6. **média de acessos a registros do arquivo:** esta operação conterá apenas uma linha, contendo a letra 'm'. Esta operação apresenta, em uma linha, apenas o valor da média do número de acessos a registros do arquivo, considerando-se uma consulta a cada um dos registros armazenados no arquivo.
7. **término da sequência de comandos:** a sequência de comandos será terminada por uma linha com a letra 'e'.

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.

3 Observações

Trabalho individual.

Data de entrega: 20/12/2013

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 recente, disponibilizado pela Oracle.

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*).