

Trabalho parcial G2 – Árvore B+

Um índice é uma estrutura de dados que facilita a recuperação de informações presentes, por exemplo, em uma tabela do banco de dados. Naturalmente, tais estruturas adicionam espaço adicional para manter a estrutura atualizada.

Como exemplo, suponha uma tabela contendo um texto constituído como abaixo:

id	title	authors	venue	year
304586	The WASA2 object-oriented workflow management system	Gottfried Vossen, Mathias Weske	International Conference on Management of Data	1999
304587	A user-centered interface for querying distributed multimedia databases	Isabel F. Cruz, Kimberly M. James	International Conference on Management of Data	1999
304589	World Wide Database-integrating the Web, CORBA and databases	Athman Bouguettaya, Boualem Benatallah	International Conference on Management of Data	1999
304590	XML-based information mediation with MIX	Chaitan Baru, Amarnath Gupta, Bertram L	International Conference on Management of Data	1999
304582	The CCUBE constraint object-oriented database system	Alexander Brodsky, Victor E. Segal, Jia C	International Conference on Management of Data	1999
304583	The Cornell Jaguar project: adding mobility to PREDATOR	Philippe Bonnet, Kyle Buza, Zhiyuan Cha	International Conference on Management of Data	1999
304584	The active MultiSync controller of the cubetree storage organization	Nick Roussopoulos, Yannis Kotidis, Yanni	International Conference on Management of Data	1999

Supondo os identificadores 30458X, o índice criado utilizando os 5 primeiros caracteres do 2 atributo (Title) é:

“The a” 304584

“The C” 304582, 304583

“The Q” 304586

....

Note que a lista de palavras chave está em ordem alfabética. O programa, a ser desenvolvido, deve receber como parâmetro o atributo que se deseja indexar e o número de caracteres. Por exemplo, atributo 2 (title) e os 10 primeiros caracteres.

Projete um sistema para produzir um índice utilizando as estruturas de dados de Árvore B+. Para facilitar, este sistema deve receber um tabela no formato .csv.

Detalhamento do trabalho:

A) O programa deve ler a entrada de um arquivo .csv.

B) Será considerado o tempo como um dos critérios para a avaliação.

C) Não poderá ser utilizado estruturas prontas, presentes em alguma biblioteca.

D) A leitura dos dados deve ser feita em massa ou Bulk Loading (não será aceito outra forma de carregamento).

E) O algoritmo deve considerar a possibilidade de redistribuição de chaves antes de realizar uma fragmentação. Ou seja, durante uma inserção, se um nó irmão tiver posições disponíveis, estas devem ser ocupadas;

F) Será realizada uma aula para apresentação do andamento do trabalho, com no mínimo a função de inserção em pleno funcionamento;

G) O programa deve implementar uma função para imprimir a árvore atual de forma visualmente legível

H) Deve ser implementado uma função para receber uma lista de ID que serão removidos

I) Pode ser utilizadas outras linguagens de programação. No entanto, o código deve simular a utilização de ponteiros para construção da árvore.

Ex:

Arquivo.txt

304582

304583

304584

304585

I) A entrada do programa deve ser da forma:

`./a.out numero_do_atributo numero_caracteres_indexados ordem_arvore`

numero_do_atributo: número do atributo a ser indexados

numero_caracteres_indexados: número de caracteres a serem indexados

ordem_arvore: ordem da árvore.

numero_caracteres_indexados ordem_arvore

Avaliação:

1. *Compilação: 10%*
2. *Execução correta: 40%*
3. *Estilo de programação: 10%*
 - *Código bem indentado, comentado (sem excesso), bem estruturado, utilizando as operações do TAD corretamente, nomes de variáveis significativos, modularização, etc.*
4. *TAD bem definido no código: 10%*
5. *Documentação: 20%*
6. *Apresentação do andamento com a estrutura do código pronto e a lógica para resolver o problema: 10%*

Atrasos:

Será descontado $2^d \cdot 5$ do trabalho (onde d representa o número de dias). Por exemplo, atrasou 3 dias terá um desconto de 40% da nota total.