

Pesquisa, Ordenação e Técnica de Armazenamento

Título: Gestão Universitária

Professor Luis Gustavo Araujo

Objetivo: Desenvolver, em Java, os conceitos vistos sobre Tablea Hash.

O estudante deve implementar apenas o caso 4, visto em sala, com função hash baseada em letras e suas posições. O estudante pode criar sua própria função hash, *caso ela tenha um desempenho melhor do que a anteriormente mencionada*. Devem ser implementadas, além da função hash, o método de inserção (insert) e busca (find). O estudante deve implementar os tratamentos de colisão vistos em sala: Linear, Quadrática (com vetor) e Separada (com lista).

Além das funções referentes à Tabela Hash, os estudantes devem implementar uma central de estatística para contabilizar a quantidade de colisões e quantidade de espaços em branco existentes na tabela hash. Um arquivo com a base de dados será disponibilizado pelo professor. Como testes use as seguintes buscas: “Zana”, “Jarrett”, “Aaron”.

Elementos obrigatórios são:

TableHash:

É um modelo abstrato da tabela hash, nela deve conter, além da tabela, os métodos: insert, find e hash. Esse modelo pode ser dividido em 3 classes distintas, implementado os métodos de colisão.

Statistic:

Esta classe é responsável por contabilizar as colisões das operações (insert e find). Deve conter o método getStatistic, que apresentará: a) número de colisões; b) espaços em branco na tabela.

Ex: “Você teve 10.000 colisões e possui 100 espaços em branco”.

Outras classes podem ser criadas para facilitar a implementação.

Avaliação do projeto:

1. O desenvolvimento do grupo de classes do modelo apontado anteriormente valerá, ao todo, 3 pontos.
2. Um relatório simplificado explicando a função hash utilizada e o funcionamento dos métodos de colisões, assim como a tabulação dos dados do Statistic dos três métodos utilizados, com uma breve discussão das diferenças, valerá 2,0 pontos.

Estrutura do relatório:

1. Capa;
2. Introdução (Breve resumo e Objetivo);
3. Função Hash Utilizada;
4. Tratamento de Colisões:
 1. Linear (resumo);
 2. Quadrática (resumo);
 3. Separada (resumo).
5. Comparação entre os métodos:
 1. Tabela com os dados estatísticos (colisões e espaços em branco);

2. Discussão sobre as diferenças dos métodos.

Entrega:

1. A entrega do trabalho deverá ser, obrigatoriamente, via e-mail (luisaraujo.ifba@gmail.com) até 17h do dia 06/06. Ou no dia 06/06, no horário da aula, em sala (através de um pen-drive ou similar);
2. Verifiquem, antes de enviar ou entregar o projeto, a integridade dos arquivos. Projetos com 100% dos arquivos corrompidos não serão avaliados, ficando o estudante como nota 0,0, na avaliação II.
3. Projetos que utilizam pacotes externos devem conter, na pasta do projeto, as classes necessárias ou arquivo .jar. Na impossibilidade de rodar o sistema, o item 3 da avaliação não será realizado.