

TECNOLOGIA E ANÁLISE EM DESENVOLVIMENTO DE SISTEMAS

Disciplina de Estrutura de Dados Lineares - 2013.2
Prof. Demétrios Coutinho

Data: 28/02/2014

- Exercícios Complementares de Tabela de Dispersão -

1. Suponha um conjunto de n chaves x formado pelos n primeiros múltiplos de número 7. Quantas colisões seriam obtidas mediante a aplicação das funções de dispersão seguintes.
 - $x \bmod 7$.
 - $x \bmod 14$.
 - $x \bmod 5$.
2. Considere o método da divisão para criar **funções de hash**. Considere que o universo das chaves, U , é o conjunto dos números inteiros não negativos. Se o tamanho da tabela for $m = 177$, encontre 3 chaves diferentes que tenham o mesmo valor de hash. Ache uma fórmula que, para qualquer chave k , lhe permita construir tantas chaves quantas desejar com o mesmo valor de hash de k . Dica: faça um teste para valores pequenos de m .
3. Demonstre a inserção das chaves 5, 28, 19, 15, 20, 33, 12, 7 e 10 numa tabela de hash com colisões resolvidas por encadeamento externo. Considere a tabela com $m = 9$ posições e a função hash como sendo $h(k) = k \bmod m$. Reconstrua a tabela para $m = 11$ (primo) e comente os resultados.
4. Implemente uma tabela de hash usando acesso direto (hashing fechado) para o seguinte problema. Uma companhia aérea tem 200 tripulantes que são identificados por um inteiro entre 1 e 200 (código de tripulante). Queremos uma tabela de hash que permita guardar a constituição da tripulação que segue num determinado avião. O programa deve ler do arquivo "pdf-linhas-aéreas" a tripulação para um voo e guardar a informação na tabela de hash. Deve depois permitir responder à questão "O tripulante k viaja neste voo?" (onde k é um código de tripulante). Se a resposta for afirmativa deve mostrar os seus dados. As estruturas que guardam a informação são do tipo seguinte:

```
Estrutura Tripulante  
| codigo, idade: Inteiro;  
| nome: String ;  
Fim-Estrutura
```

O arquivo deve ter um tripulante por linha. Ou seja, seguinte formato:

```
    codigo1;idade1;nome1  
codigo2;idade2;nome2  
codigo3;idade3;nome3  
:           :       :
```

5. Implemente o algoritmo de **busca, inserção e remoção** de uma tabela hash com tratamento de colisão por encademanento interno e externo.