

INF01124 - Classificação e Pesquisa de Dados - Exercício 4

Professor João Comba

1 Implementação de Tabela Hash

O objetivo deste laboratório é implementar uma tabela Hash, com diferentes tamanhos de tabela (M). A tabela hash deve armazenar registros compostos de uma chave identificadora e dados satélites associados. Para a implementação da Tabela Hash, realize as seguintes tarefas:

1. escolha uma função de hash para mapear a chave de cada registro para um inteiro entre 0 e $M - 1$;
2. implemente o método de resolução de conflitos de endereçamento fechado com listas encadeadas;
3. implemente a operação de inserção de um registro na tabela hash; e
4. implemente a operação de busca de um registro por uma chave na tabela hash.

2 Arquivo de Testes

A implementação da tabela Hash deve ser testada com um arquivo .CSV contendo descrições de 18944 nomes de jogadores de futebol. O arquivo é disponibilizado junto ao enunciado deste trabalho com o nome de *players-fifa.csv*. A primeira linha do arquivo contém a descrição em cada coluna do registro a ser armazenado. A chave é o identificador de cada jogador (sofifa_id), e os dados satélites consistem no nome completo do jogador e um string contendo uma lista de posições que o jogador atua. Cada linha do arquivo contém os dados de um jogador.

Um exemplo deste arquivo é mostrado abaixo:

```
sofifa_id,name,player_positions
158023,Lionel Andres Messi Cuccittini,"RW, ST, CF"
20801,Cristiano Ronaldo dos Santos Aveiro,"ST, LW"
200389,Jan Oblak,GK
188545,Robert Lewandowski,ST
190871,Neymar da Silva Santos Junior,"LW, CAM"
192985,Kevin De Bruyne,"CAM, CM"
231747,Kylian Mbappe Lottin,"ST, LW, RW"
192448,Marc-Andre ter Stegen,GK
203376,Virgil van Dijk,CB
212831,Alisson Ramses Becker,GK
...
```

A lista de posições nas quais o jogador atua usa as seguintes abreviaturas:

```
GK goalkeeper
CB center-back
RB right back
LB left back
CDM central defensive midfielder
CM central midfielder
```

RM right midfielder
LM left midfielder
CAM central attacking midfielder
LW left wing
RW right wing
CF center forward
ST striker

3 Experimentos

Para testar a sua implementação da tabela hash vamos coletar estatísticas relacionadas à construção da tabela e consultas por registros armazenados na tabela. Como o tamanho da tabela é muito importante na performance, vamos realizar experimentos com os seguintes valores de tamanho da tabela: 3793, 6637, 9473, 12323 e 15149. Todos estes números são números primos próximos aos percentuais de 20%, 35%, 50%, 65% e 80% do total número de jogadores. Para cada tamanho de tabela, realize um experimento que consiste em inserir na tabela os dados de jogadores do arquivo *players-fifa.csv*, e após realize consultas por identificadores (sofifa.id) de jogadores descritas no arquivo *consultas.csv*. Para avaliar a performance para cada tamanho de tabela, colete as seguintes estatísticas sobre a construção da tabela e consultas realizadas:

- **estatísticas da tabela hash:**

- tempo de construção da tabela (em milissegundos): t3793, t6637, t9473, t12323 e t15149
- taxa de ocupação da tabela (número de entradas usadas divididas pelo tamanho da tabela): o3793, o6637, o9473, o2323 e o5149
- tamanho máximo de lista: M3793, M6637, M9473, M2323 e M5149
- média do tamanho das listas (para entradas não vazias): m3793, m6637, m9473, m2323 e m5149

- **estatísticas das consultas:**

- tempo de realização de todas as consultas (em milissegundos): a3793, a6637, a9473, a12323 e a15149
- Para cada chave encontrada, retorne o id do jogador (id_jogador), nome do jogador (nome_jogador) e número de testes (numero_testes) na tabela hash realizado até encontrar o jogador: c3793,c6637,c9473,c2323,c5149
- Para cada chave não encontrada, retorne 99999 (id_jogador), NAO_ENCONTRADO (nome_jogador) e número de testes (numero_testes) realizado até concluir que a chave não está na tabela: c3793,c6637,c9473,c2323,c5149

Os resultados devem ser gravados em dois arquivos: *estatisticas_construção.txt* e *estatisticas_consultas.txt*.

O arquivo de saída *estatisticas_construção.txt* deve ter rigorosamente o seguinte formato:

```
t3793,t6637,t9473,t12323,t15149
o3793,o6637,o9473,o2323,o5149
M3793,M6637,M9473,M2323,M5149
m3793,m6637,m9473,m2323,m5149
```

O arquivo de saída *estatisticas_consultas.txt* deve ter rigorosamente o seguinte formato:

```
a3793,a6637,a9473,a2323,a5149
id\_jogador,nome\_jogador,c3793,c6637,c9473,c2323,c5149
id\_jogador,nome\_jogador,c3793,c6637,c9473,c2323,c5149
id\_jogador,nome\_jogador,c3793,c6637,c9473,c2323,c5149
id\_jogador,nome\_jogador,c3793,c6637,c9473,c2323,c5149
....
id\_jogador,nome\_jogador,c3793,c6637,c9473,c2323,c5149
id\_jogador,nome\_jogador,c3793,c6637,c9473,c2323,c5149
id\_jogador,nome\_jogador,c3793,c6637,c9473,c2323,c5149
```

4 Opcional: Desafio Bônus - Problema Hash Tables

Resolva o problema BEE 1256 - Hash Tables

Este é um problema de maratona de programação. O problema será considerado completo se a solução for aceita no site da BeeCrowd. O comprovante da aceitação e o código deve ser entregue junto com a solução do laboratório. A entrega correta valerá um adicional de 25% pontos.

5 Entrega

A solução deve ser enviada pelo Moodle dentro de um arquivo .zip, contendo os seguintes arquivos:

- **integrantes.txt**: coloque o nome dos integrantes do grupo (até 2 pessoas) , com um nome por linha
- **estatisticas_construcao.txt**
- **estatisticas_consultas.txt**
- código fonte correspondente a solução
- opcional: código e comprovante de aceite do problema desafio