

## **Trabalho II – Comparação entre buscas**

Escreva um programa que leia um número inteiro  $n$  e em seguida leia  $n$  pares  $p$   $s$ , onde  $s$  indica o nome de um personagem fictício (nome e sobrenome, sendo que todos os sobrenomes são compostos por apenas uma palavra) e  $p$  indica um número de ordem (identificador) associado a esse personagem. Depois, leia um número inteiro  $m$  e mais  $m$  nomes (nome e sobrenome). Para cada um desses nomes, mostre o número de ordem associado ao personagem. Escreva as seguintes três versões de estruturas de busca e as compare em relação ao tempo necessário para a inserção, para a busca e o tempo total.

- (1) Use um vetor ou lista e realize buscas lineares
- (2) Use uma estrutura de árvore balanceada (ex: rubro negra ou AVL)\*
- (3) Use uma estrutura de Hash encadeado com árvore balanceada\*\*

\* Pode-se utilizar as estruturas prontas da linguagem escolhida (ex: map/set em C++ ou TreeSet no Java), ou utilizar estruturas próprias (implementadas em EDA, por exemplo).

\*\* A estrutura de Hash deve utilizar um tamanho de vetor de hash modificável. Deve-se utilizar pelo menos dois valores diferentes para este tamanho na realização dos testes.

Os casos de teste e as respostas se encontram na página da disciplina: `exercício-hash.zip`. As respostas devem ser comparadas com os gabaritos fornecidos (`saida-n.txt`). Tratam-se de cinco casos de teste distintos:

- (1) 5.000 nomes / 1.000 consultas
- (2) 25.000 nomes / 10.000 consultas
- (3) 50.000 nomes / 10.000 consultas
- (4) 50.000 nomes / 25.000 consultas
- (5) 100.000 nomes / 75.000 consultas

**Linguagens permitidas:** C, C++, Java.

### **Material a ser apresentado:**

- Códigos das implementações;
- Gráfico comparativo de tempo de execução para as operações de busca, inserção e tempo total de execução dos algoritmos implementados;
- Gráfico de lotação (histograma) dos valores pelo hash;
- Análise de complexidade de tempo e espaço para o melhor e pior caso do algoritmo de *hashing* utilizado e das estruturas de busca implementadas (considerando as duas versões da estrutura de hash com tamanhos diferentes);

### **Restrições:**

- Este trabalho poderá ser realizado no máximo em duplas
- Este trabalho deverá ser entregue/apresentado no dia **18-19/10/2018**
- Perguntas serão realizadas para validar a autoria dos códigos