

Tabela Hash

#Descrição Geral:

O trabalho envolve a implementação de uma Tabela Hash com foco na aplicação de funções hash e análise de desempenho. Os alunos deverão desenvolver uma solução para a inserção, busca e análise de colisões em diferentes cenários de dados.

#Requisitos:

1. **Estrutura de Dados:**

- Implementação de uma Tabela Hash para armazenar registros de empregados.
- Cada registro deve conter:
 - Matrícula (9 dígitos)
 - Salário
 - Código do Setor

2. **Função Hash:** modular com:

- $M = 1000$
- $M = 10000$
- $M = 100000$

3. **Cenários de Teste:**

- População da Tabela Hash com diferentes volumes de dados:
 - 5K,
 - 20K
 - 100K registros
- Realizar análises de:
 - Número de colisões
 - Tempo médio de busca (em milissegundos)
 - Tamanho médio das chaves

4. **Comparação de Desempenho:**

- Comparar o tempo médio de busca na Tabela Hash vs Vetor Sequencial contendo 100K registros.
- Rodar 10 vezes, com diferentes dados de teste. Computar a média de busca.

#Avaliação: Datas e Pontuações:

- **16/08 – até às 18:00 (2.0 pts):** Entrega do Protótipo de Implementação (Proof of Concept - POC).
 - Funções Hash implementada.
 - Dados gerados e Tabela Hash populada (Mínimo de 100 amostras na POC)
 - **Upload de código-fonte na aba da tarefa no tarefa no ava**
- **21/08 – até às 23:59 (6.0 pts):** Entrega final da Implementação.
 - Código-fonte (.c ou .py).
 - Apresentação contendo as Análises e Resultados (.ppt / pdf).
 - **Entregar um ZIP na aba da tarefa no tarefa no ava**
- **23/08 – Início da Aula (2.0 pts):**
 - Apresentação dos resultados durante a aula.
 - A ordem será definida no início da aula
 - **OBS: Os slides devem ser entregues previamente (21/08)**

#Rubricas de Avaliação:

1. Prova de Conceito (POC) - 2.0 pts – 16/08 – 18:00

- **2.0 - Excelente:** Função hash e tabela hash implementadas corretamente. Dados gerados e populados adequadamente com pelo menos 100 amostras. Tratamento de colisões
- **1.5 - Bom:** Implementação funcional, com dados populados e funcionais, mas com pequenas inconsistências ou melhorias possíveis.
- **1.0 - Satisfatório:** Implementação básica com problemas na função hash ou distribuição dos dados (indexação ou colisão), afetando o desempenho.
- **0.5 - Insatisfatório:** Implementação parcial ou incorreta, com falhas significativas na função hash ou manipulação dos dados.

2. Implementação Completa - 6.0 pts – 21/08 – 23:59

- **6.0 - Excelente:** Implementação completa e correta da tabela hash. Função hash eficiente, dados bem distribuídos, e tratamento de colisões. A tabela funciona adequadamente em todos os cenários testados. A função de busca funciona corretamente
- **4.0 - Bom:** Implementação funcional, mas com alguns problemas menores na distribuição de dados ou tratamento de colisões. A tabela é eficiente na maioria dos cenários, tanto inserção quanto busca.
- **2.0 - Satisfatório:** Implementação correta, mas com problemas de desempenho, distribuição de dados ou alta taxa de colisões devido a manipulação errada da hash, na inserção ou na busca
- **1.0 - Insatisfatório:** Implementação incompleta ou com erros significativos, resultando em baixa eficiência ou falha nos testes.

3. Apresentação e Análises - 2.0 pts – Apresentação em Sala (Início da Aula)

Critérios:

- **2.0 - Excelente:** Apresentação clara, organizada e concisa. Análises detalhadas e corretas, comparando eficientemente a tabela hash com outras abordagens.
- **1.5 - Bom:** Apresentação adequada, com boas análises, mas com pequenos pontos de melhoria na clareza ou organização.
- **1.0 - Satisfatório:** Apresentação básica, com análises incompletas ou menos detalhadas. Comparações realizadas de forma superficial.
- **0.5 - Insatisfatório:** Apresentação confusa ou mal organizada, com análises ausentes ou incorretas, comprometendo a interpretação dos resultados.

++++++FIM DA DOCUMENTAÇÃO++++++