Aula 10 - Exercício prático Tabela Hash

Aluno: Gian Franco Joel Condori Luna

November 6, 2024

Exercices

1 (1,0) Implemente um algoritmo para função Hash que:

a) Faz resolução de colisões por encadeamento com a função hash da divisão.

b) Faz resolução de colisões por endereçamento aberto usando sondagem linear.

c) Insira 100.000 elementos gerados aleatoriamente no intervalo [0-100.000] em uma tabela hash com m = 50.000 posições no encadeamento e m = 100.000 no endereçamento aberto e compute o tempo de inserção em cada caso. Depois pesquise o elemento 10.000:

Solução:

(O código está no arquivo python)

Hash por encadeamento	**Hash por endereçamento aberto**
Tempo Inserção Função hash divisão: 0.0372 s	Tempo Inserção Sondagem Linear: 2.1680 s
Tempo Busca Função hash divisão: 0.000038 s	Tempo Busca Sondagem Linear: 0.000032 s

Discussão:

- **Tempo de Inserção:** A função hash com encadeamento levou significativamente menos tempo para inserir os elementos (0,0372 s) comparado ao método de endereçamento aberto com sondagem linear (2,1680 s). Isso sugere que o encadeamento é mais eficiente para inserções quando a tabela hash tem um número elevado de elementos e há alta possibilidade de colisões.

- **Tempo de Busca:** Para a busca, ambos os métodos apresentaram tempos de execução muito pequenos, com uma diferença mínima entre eles

(0,000038 s para encadeamento e 0,000032 s para sondagem linear). Embora o tempo de busca seja similar, isso indica que ambos os métodos são eficientes para buscas rápidas, mas o encadeamento ainda pode ser uma escolha mais vantajosa para um número elevado de inserções.

Fontes Consultadas

- https://chatgpt.com/