

Relatório TDE 3 – Tabela Hash

Integrantes: Claudio Colombo e Eric Simões

Método de tratamento de colisões utilizado: Encadeamento Externo

HashTableFunction1

Função Hash: A função hash utilizada nesta tabela é baseada na soma dos valores ASCII de cada caractere na chave. Para cada caractere da string, o hash é atualizado multiplicando o hash atual por 31 e somando o valor ASCII do caractere. Isso gera um valor de hash que é posteriormente reduzido para o tamanho da tabela usando a operação módulo (%).

Resultados Tabela 1:

Número total de colisões: 4004

Tabela 1 - Tempo total de inserção: 3.161 ms

Tabela 1 - Tempo total de busca: 2.4964 ms

Total de posições ocupadas: 997

Total de clusters (posições com múltiplas chaves): 961

Total de posições vazias: 3

HashTableFunction2

Função Hash: A função hash desta tabela é uma implementação mais simples que calcula a soma dos valores ASCII de todos os caracteres da chave. O hash resultante é então ajustado para se adequar ao tamanho da tabela usando a operação módulo (%). Essa abordagem, embora mais direta, pode resultar em uma maior quantidade de colisões em comparação com a primeira função.

Resultados Tabela 2:

Número total de colisões: 4429

Tabela 2 - Tempo total de inserção: 1.5181 ms

Tabela 2 - Tempo total de busca: 2.9182 ms

Total de posições ocupadas: 572

Total de clusters (posições com múltiplas chaves): 448

Total de posições vazias: 428

Conclusão Final:

Considerando todos os fatores, a HashTableFunction1 se destaca como a melhor implementação entre as duas, apesar de ter um tempo de inserção ligeiramente maior. Ela apresenta um menor número de colisões, um melhor tempo de busca e uma distribuição de chaves mais eficiente, com menos posições vazias e uma maior ocupação. Essa análise mostra que, em muitos casos, uma função hash mais complexa (que resulta em menos colisões) pode ser mais vantajosa do que uma abordagem mais simples, que pode causar mais colisões e, conseqüentemente, afetar o desempenho geral da tabela hash.