

Estruturas de Dados

Trabalho para o 2º bimestre:

Trabalho: Implementação de uma Tabela de Hash com Tentativa Quadrática

Objetivo do Trabalho:

Desenvolver uma tabela de hash para armazenar nomes de pessoas. A tabela de hash será representada por uma matriz com duas colunas: a primeira para o estado (L para livre, O para ocupado e R para deletado) e a segunda para armazenar o elemento (neste caso, o nome da pessoa). A função de hash utilizará o método de tentativa quadrática para resolver colisões.

Descrição da Função de Hash:

A função de hash principal usa o método do resto da divisão. Dada uma chave (nome, neste caso), a função calcula o valor do hash como o resto da divisão da soma dos valores ASCII dos caracteres da chave pelo tamanho da tabela.

Tentativa Quadrática:

Em caso de colisões (quando uma posição calculada já está ocupada), a tentativa quadrática é usada. Isso significa que, se a posição $h(k)$ estiver ocupada, o programa tentará $h(k) + 1^2$, $h(k) + 2^2$, $h(k) + 3^2$, e assim por diante, até encontrar uma posição livre ou deletada.

O que deve ser feito:

1. Implementar a função de hash que converte o nome em um índice da tabela.
2. Implementar a lógica de tentativa quadrática para lidar com colisões.
3. Criar a tabela de hash com um tamanho fixo e predefinido.
4. Implementar funções para inserir, buscar e deletar nomes da tabela de hash.
5. Ler nomes de um arquivo e inseri-los na tabela de hash usando as funções criadas.
6. Garantir que a inserção só aconteça em posições com estado L (livre) ou R (deletado), e alterar o estado para O (ocupado) após a inserção.

Funções a Serem Implementadas e as demais de leitura de arquivo:

`funcao_hash(nome, tamanho_tabela)`: Calcula o índice para um dado nome.

`inserir(tabela, nome)`: Insere um nome na tabela de hash.

`buscar(tabela, nome)`: Busca um nome na tabela de hash.

`deletar(tabela, nome)`: Deleta um nome da tabela de hash.

Testar todas as funcionalidades devem ser implementadas em python e comentadas linha a linha.

Grupo de 3 pessoas.