Relatório Algoritmos e Estruturas de Dados 3

Trabalho 2.

Antonio da Ressurreição Filho.

Matheus Gastal Magalhães.

1. Objetivo principal do trabalho:

O sistema desenvolvido tem como objetivo principal gerenciar a inserção e organização de chaves inteiras em duas estruturas de tabelas hash com tratamento de colisões. O sistema recebe comandos do tipo 'i <chave>' ou 'ir <chave>' via entrada padrão, insere ou retira as chaves em uma tabela hash (tabela1) e, em caso de colisão, aplica um algoritmo diferente (outra função hash) para armazenar a chave antiga em uma segunda tabela (tabela2).

2. Estruturas Utilizadas:

Estrutura tabela: Utilizada para representar uma posição em uma tabela hash.

Contém os campos: int chave (valor armazenado), int vazio (indicador de slot livre), int excluido (marcador de exclusão lógica).

Estrutura saida: Serve para gerar e ordenar a saída final das chaves inseridas.

Contém os campos: int chave (valor da chave inserida); int posicao (posição da chave dentro da tabela); char tabela[3] (string identificadora da tabela (ex: "T1" ou "T2")).

3. Lógica do Programa:

Duas tabelas hash de tamanho fixo 11 são criadas dinamicamente com cria_tabela(). O programa lê comandos no formato 'i <chave>' ou 'r<chave>'. Calcula a posição h1 = chave % 11 para inserção em tabela1. Se a posição estiver **vazia** ou **marcada como excluída**, a chave é inserida diretamente. Caso contrário, chama-se uma nova função hash para inserir a chave contida na tabela 1 para a tabela 2 e sobreescreve o novo valor na tabela 1.

A função gera_saida() percorre tabela1 e tabela2, coletando todas as chaves não vazias. Cada chave coletada é adicionada a um vetor saída. O vetor é ordenado por: valor da chave (ordem crescente), nome da tabela (T1 antes de T2), posição (menor posição primeiro). O vetor ordenado é impresso, mostrando em qual tabela e posição cada chave foi armazenada.

4. Conclusão:

O sistema implementado demonstra o uso de técnicas básicas de hashing com tratamento de colisões por redispersão entre duas tabelas. Nós aprendemos imensamente sobre o básico de tabela hasing, para que no futuro possamos aperfeiçoar essas técnicas em trabalhos mais complexos.