

ESTRUTURA DE DADOS

HASHING

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

1

BIBLIOGRAFIA

- WEISS, Mark A. **Data Structures & Algorithm Analysis in Java**. Addison Wesley Longman, 1999.
 - Chapter 5: Hashing

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

2

INTRODUÇÃO

- **Hashing** é uma técnica utilizada para realizar inserções, deleções e consultas em tempo médio constante;
- A estrutura de dados tabela hash ideal nada mais é do que um array de tamanho fixo contendo itens;

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

3

INTRODUÇÃO

- Se o tamanho da tabela hash for *size*, então cada chave é mapeada para um número no intervalo de 0 a $size - 1$;
- Este mapeamento é realizado através de uma **função hash**;

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

4

HASH FUNCTION

- Se as chaves de entrada forem inteiros, simplesmente retornando **$key \bmod size$** é uma estratégia aceitável;
- Se as chaves forem literais, podemos somar os códigos ASCII (ou Unicode) dos caracteres da literal;
- Uma outra possibilidade seria a de também considerar o posição do caracter na literal;

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

5

COLISÕES

- Uma **colisão** ocorre quando um item é mapeado para uma posição na tabela hash que já possui um item;
- Existem algumas técnicas para fazer o tratamento de exceções:
 - Encadeamento separado.
 - Encadeamento aberto.

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

6

ENCADEAMENTO SEPARADO

- Cada posição $A[i]$ armazena um referência para uma lista que armazena todos os elementos que a função hash mapeou para a posição $A[i]$;
- Precisa de uma estrutura de dados auxiliar;
- Pode ter mais elementos que posições na tabela hash, pois cada posição da tabela hash pode conter uma lista de itens;

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

7

ENCADEAMENTO ABERTO

- O tamanho da tabela hash deve ser maior que a quantidade de itens;
- Cada item é armazenado em uma posição diferente da tabela;
- Pode usar:
 - Teste linear.
 - Teste quadrático.
 - Hashing duplo.

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

8

ENCADEAMENTO ABERTO

- Exemplo:
 - Size: 10
 - Função Hash: $f(x) = x \bmod 10$
 - Colisão: encadeamento aberto

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

9

TESTE LINEAR

- Enquanto a tabela for grande o suficiente, terá pelo menos uma posição em aberto;
- Tenta inserir o elemento na posição definida pela função hash (i);
- Caso esta já tenha um item, segue sequencialmente ($i+1$) as demais posições até encontrar uma posição vazia;
- Pode demorar muito para encontrar uma posição vazia;

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

10

TESTE LINEAR

- Exemplo:
 - Size: 10
 - Função Hash: $f(x) = x \bmod 10$
 - Colisão: teste linear

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

11

TESTE QUADRÁTICO

- Usa uma segunda função, normalmente i^2 para encontrar uma posição vazia;
- Problemas com o endereçamento aberto:
 - A operação de deleção pode ficar comprometida.

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

12

TESTE QUADRÁTICO

- Exemplo:
 - Size: **10**
 - Função Hash: **$f(x) = x \bmod 10$**
 - Colisão: teste quadrático

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

13

REHASHING

- Quando uma tabela hash começar a ficar cheia, colisões são mais comuns;
- Neste caso é interessante aumentar o tamanho da tabela hash;
- Exemplo:
 - dobro do tamanho
 - em 50%
 - etc.

3/6/2008

Estrutura de Dados
Prof. Ademar Schmitz, M.Sc.

14