

Algoritmos e Estruturas de Dados

Professor: Ciro Meneses Santos

Universidade Presidente Antonio Carlos – UNIPAC
Ciência da Computação

1

Hashing - Definição

- ♦ Em ciência da computação a **tabela hash** (de *hashing*, no inglês), também conhecida por **tabela de espalhamento** ou **tabela de dispersão**, é uma estrutura de dados especial, que associa chaves de pesquisa (*hash*) a valores. Seu objetivo é, a partir de uma chave simples, fazer uma busca rápida e obter o valor desejado.

2

Hashing - Definição

- ♦ Os registros armazenados em uma tabela são diretamente endereçados a partir de uma transformação aritmética sobre a chave de pesquisa.
- ♦ Método de pesquisa com o objetivo de melhorar a performance de busca e inclusão de chaves
- ♦ Metas para alcançar o objetivo:
 - uso de vetor: acesso direto a uma posição ($O(1)$)
 - uso de uma função (função hash – hash(chave)) que traduz um valor de chave para uma posição no vetor
 - hash(chave) deve ter fácil de computar: complexidade $O(1)$

3

Hashing - Definição

- ♦ Um método de pesquisa através da transformação de chave é constituído de duas etapas principais:
 - Computar o valor da função de transformação (também conhecida por função hashing), a qual transforma a chave de pesquisa em um endereço da tabela.
 - Considerando que duas ou mais chaves podem ser transformada em um mesmo endereço de tabela, é necessário existir um método para lidar com colisões.

4

Funções de Transformação

- ♦ Uma função de transformação deve mapear chaves em inteiros dentro do intervalo $[0..M-1]$, onde M é o tamanho da tabela.
- ♦ A função de transformação ideal é aquela que:
 - Seja simples de ser computada.
 - Para cada chave de entrada, qualquer uma das saídas possíveis é igualmente provável de ocorrer.

5

Funções de Transformação

$$N = \text{dim}(tabela)$$

$$h(K) = K \bmod N$$

- Vantajoso escolher para N um número primo.
- Não sendo N número primo, mas não tendo factores primos inferiores a 20 obtemos também bons resultados.
- Quando pouco se sabe sobre as chaves a divisão é muitas das vezes a função de dispersão escolhida

6

Funções de Transformação

- Funções hash típicas:
 - Operam sobre o resto da divisão por M ($ch \bmod M$)
 - M é um número primo (poucas divisões exatas!)
- Exemplo:

$$n = 2500 \quad M = 1031 \quad h(ch) = ch \bmod M$$

$$h(23) = 23 \quad h(1401) = 370$$

$$h(24) = 24 \quad h(1402) = 371$$

7

Chaves Não-Numéricas

- Chaves não-numéricas:
- Aplicar duas funções:
 - $f(ch)$: converte ch para uma representação numérica
 - $g(ch)$: converte $f(ch)$ para o intervalo $[0, \dots, M-1]$
- $h(ch) = g(f(ch))$

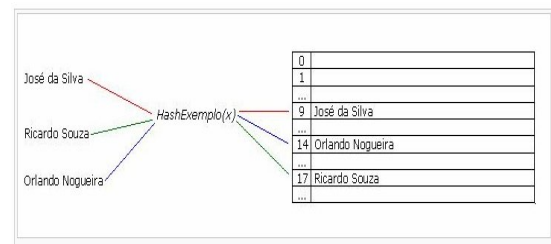
8

Chaves Não-Numéricas

- Técnicas para $f(ch)$:
 - Soma dos códigos numéricos dos caracteres
- Exemplo: A N A S I L V A
- $$f(ch) = \text{int}(A) + \text{int}(N) + \dots + \text{int}(A)$$
- $$g(ch) = f(ch) \bmod M$$

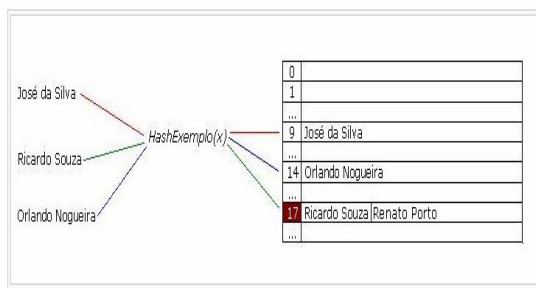
9

Chaves Não-Numéricas



10

Chaves Não-Numéricas



11

Resolução de Colisão

- Rehash
- Listas Encadeadas
- Endereçamento Aberto
- Indexação Perfeita

12

Resolução de Colisão Rehash

- ♦ É utilizada uma segunda função matemática para calcular a posição em que deve ser feita a próxima prova. Esta função é denominada *Função de Rehash*.

13

Resolução de Colisão Listas Encadeadas

- ♦ Uma das formas de resolver as colisões é simplesmente construir uma lista linear encadeada para cada endereço da tabela. Assim, todas as chaves com mesmo endereço são encadeadas em uma lista linear.

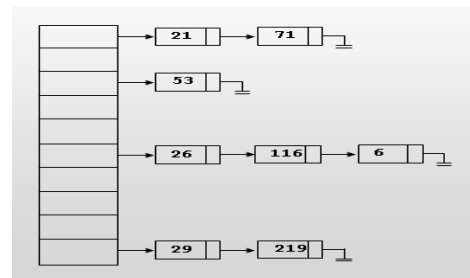
14

Resolução de Colisão Listas Encadeadas

- ♦ Uma das formas de resolver as colisões é simplesmente construir uma lista linear encadeada para cada endereço da tabela. Assim, todas as chaves com mesmo endereço são encadeadas em uma lista linear.

15

Resolução de Colisão Listas Encadeadas



16

Resolução de Colisão Open Addressing

1. Existem várias propostas para a escolha de localizações alternativas.
2. A alternativa mais simples é chamada de hashing linear, onde a posição h_j na tabela é dada por:

$$h_j = (h(x) + j) \bmod M, \text{ para } 1 \leq j \leq M-1$$

17

Exemplo com Hashing Linear


- ♦ Se a i -ésima letra do alfabeto é representada pelo número i e a função de transformação $h(\text{chave}) = \text{chave} \bmod M$ é utilizada para $M = 7$
- ♦ Então o resultado da inserção das chaves LUNES na tabela, usando hashing linear para resolver colisões é mostrado a seguir.

18

Exemplo com Hashing Linear

$h(L) = h(12) = 5$
 $h(U) = h(21) = 0$
 $h(N) = h(14) = 0$
 $h(E) = h(5) = 5$
 $h(S) = h(19) = 5$

0	U
1	N
2	S
3	
4	
5	L
6	E



19

Análise do Hashing Linear

- O hashing linear sofre de um mal chamado agrupamento (clustering)
- Este fenômeno ocorre na medida em que a tabela começa a ficar cheia, pois a inserção de uma nova chave tende a ocupar uma posição na tabela que esteja contígua a outras posições já ocupadas, o que deteriora o tempo necessário para novas pesquisas
- Entretanto, apesar do hashing linear ser um método relativamente pobre para resolver colisões os resultados apresentados são bons.

20