

Estrutura de Dados

Hashing

Profa. Dra. Jaqueline Brigladori Pugliesi

Hashing

- ▶ Os registros armazenados em uma tabela são diretamente endereçados a partir de uma transformação aritmética sobre a chave de pesquisa.
- ▶ Hash significa (Webster's New World Dictionary):
 1. Fazer picadinho de carne e vegetais para cozinhar.
 2. Fazer uma bagunça.

Hashing (cont.)

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

Hashing (cont.)

- ▶ Qualquer que seja a função de transformação, algumas colisões irão ocorrer fatalmente, e tais colisões têm de ser resolvidas de alguma forma.
- ▶ Mesmo que se obtenha uma função de transformação que distribua os registros de forma uniforme entre as entradas da tabela, existe uma alta probabilidade de haver colisões.

Paradoxo do aniversário

- ▶ O paradoxo do aniversário (Feller, 1968), diz que em um grupo de 23 ou mais pessoas, juntas ao acaso, existe uma chance maior do que 50% de que 2 pessoas comemorem aniversário no mesmo dia.
- ▶ Assim, se for utilizada uma função de transformação uniforme que enderece 23 chaves randômicas em uma tabela de tamanho 365, a probabilidade de que haja colisões é maior do que 50%.

Funções de Transformação

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

Funções de Transformação (cont.)

- ▶ As chaves não numéricas devem ser transformadas em números.
- ▶ Uma função bastante utilizada é resto da divisão por M
$$h(K) = K \bmod M$$
onde K é um inteiro correspondente à chave.

Funções de Transformação (cont.)

- ▶ Cuidado na escolha do valor de M. M deve ser um número primo, mas não qualquer primo: devem ser evitados os números primos obtidos a partir de $b^i \pm j$ onde b é a base do conjunto de caracteres (geralmente $b = 64$ para BCD, 128 para ASCII, 256 para EBCDIC, ou 100 para alguns códigos decimais), e i e j são pequenos inteiros.

Listas Encadeadas

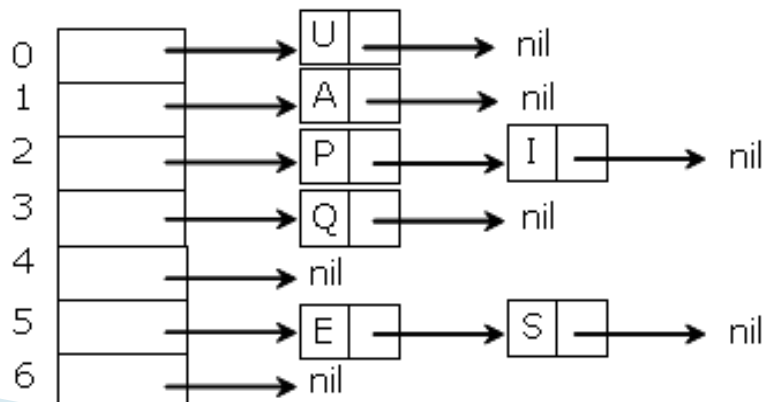
- ▶ Uma das formas de resolver as colisões é simplesmente construir uma lista linear encadeada para cada endereço da tabela. Assim, todas as chaves com mesmo endereço são encadeadas em uma lista linear.
- ▶ Exemplo: Se a i -ésima letra do alfabeto é representada pelo número i e a função de transformação $h(\text{Chave}) = \text{Chave} \bmod M$ é utilizada para $M = 7$, o resultado da inserção das chaves P E S Q U I A na tabela é o seguinte:

Profa. Dra. Jaqueline Brigliadori Pugliesi

9

Listas Encadeadas (cont.)

- ▶ Por exemplo, $h(A) = h(1) = 1$, $h(E) = h(5) = 5$, $h(S) = h(19) = 5$, e assim por diante.



Profa. Dra. Jaqueline Brigliadori Pugliesi

10

Listas Encadeadas (cont.)

- Assumindo que qualquer item do conjunto tem igual probabilidade de ser endereçado para qualquer entrada de T , então o comprimento esperado de cada lista encadeada é N/M , onde N representa o número de registros na tabela e M o tamanho da tabela.

Endereçamento Aberto (*Open Addressing*)

- Quando o número de registros a serem armazenados na tabela puder ser previamente estimado, então não haverá necessidade de usar apontadores para armazenar os registros.
- Existem vários métodos para armazenar N registros em uma tabela de tamanho $M > N$, os quais utilizam os lugares vazios na própria tabela para resolver as colisões.
- No endereçamento aberto todas as chaves são armazenadas na própria tabela, sem o uso de apontadores explícitos.

Hashing Linear

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

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

- 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 L U N E S na tabela, usando hashing linear para resolver colisões é:

Profa. Dra. Jaqueline Brigliadori Pugliesi

13

Hashing Linear (cont.)

- Por exemplo,

$$h(L) = h(12) = 5$$

$$h(U) = h(21) = 0$$

$$h(N) = h(14) = 0$$

$$h(E) = h(5) = 5$$

$$h(S) = h(19) = 5$$

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

Profa. Dra. Jaqueline Brigliadori Pugliesi

14

Hashing Linear (cont.)

- ▶ 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.

Hashing Linear (cont.)

- ▶ Vantagens:
 - Alta eficiência no custo de pesquisa, que é $O(1)$ para o caso médio.
 - Simplicidade de implementação.
- ▶ Desvantagens:
 - Custo para recuperar os registros na ordem lexicográfica das chaves é alto, sendo necessário ordenar o arquivo.
 - Pior caso é $O(N)$.

Hashing Perfeito

- ▶ Se $h(x_i) = h(x_j)$ se e somente se $i = j$, então não há colisões, e a função de transformação é chamada de função de transformação perfeita ou função hashing perfeita (hp).
- ▶ Se o número de chaves N e o tamanho da tabela M são iguais ($\alpha = N/M = 1$), então temos uma função de transformação perfeita mínima.

Hashing Perfeito (cont.)

- ▶ Vantagens e Desvantagens
 - Não há necessidade de armazenar a chave, pois o registro é localizado sempre a partir do resultado da função de transformação.
 - Uma função de transformação perfeita é específica para um conjunto de chaves conhecido.
 - A desvantagem no caso é o espaço ocupado para descrever a função de transformação hp. Entretanto, é possível obter um método com $M \approx 1,25N$, para valores grandes de N .

Exercício

- ▶ Desenhe o conteúdo da tabela hash resultante da inserção de registros com as chaves N I V O Z U A E F R B L , nesta ordem, em uma tabela inicialmente vazia de:
 - a) tamanho 7, usando listas encadeadas;
 - b) tamanho 13, usando hashing linear.

Aplicações

- ▶ Criptografia
- ▶ Banco de Dados
- ▶ Compiladores
- ▶ Verificação de integridade de dados ou mensagens
- ▶ Verificação de senha
- ▶ Identificador de arquivos ou dados

Trabalho – Parte 3

- ▶ Fazer um programa que utilize Hashing Linear para manipular chaves (representadas como um caractere) e possua as funções para inserir e buscar chaves nesta tabela hash.

FIM