ESTRUTURAS DE DADOS II

MSC. DANIELE CARVALHO OLIVEIRA

DOUTORANDA EM CIÊNCIA DA COMPUTAÇÃO - USP

MESTRE EM CIÊNCIA DA COMPUTAÇÃO – UFU

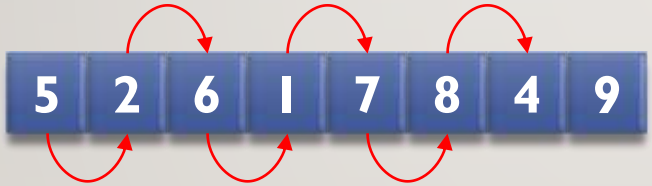
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO - UFJF

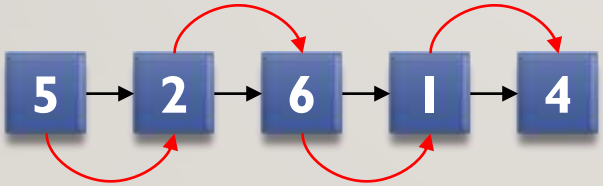
2

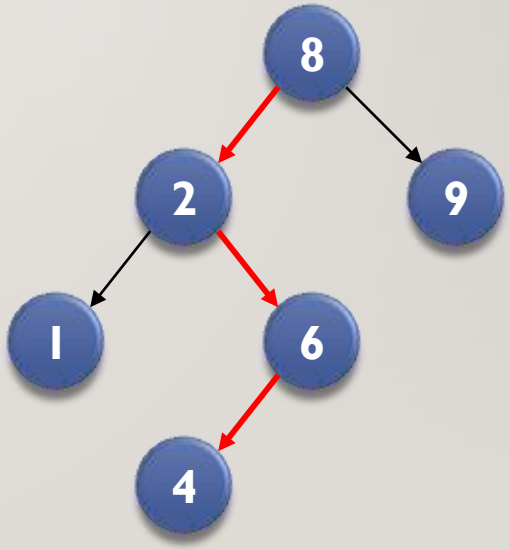
HASHING

3 INTRODUÇÃO

- Todos os métodos de pesquisa apresentados anteriormente são baseados em comparação de chaves
  - Demandando esforço computacional O(n), em média.







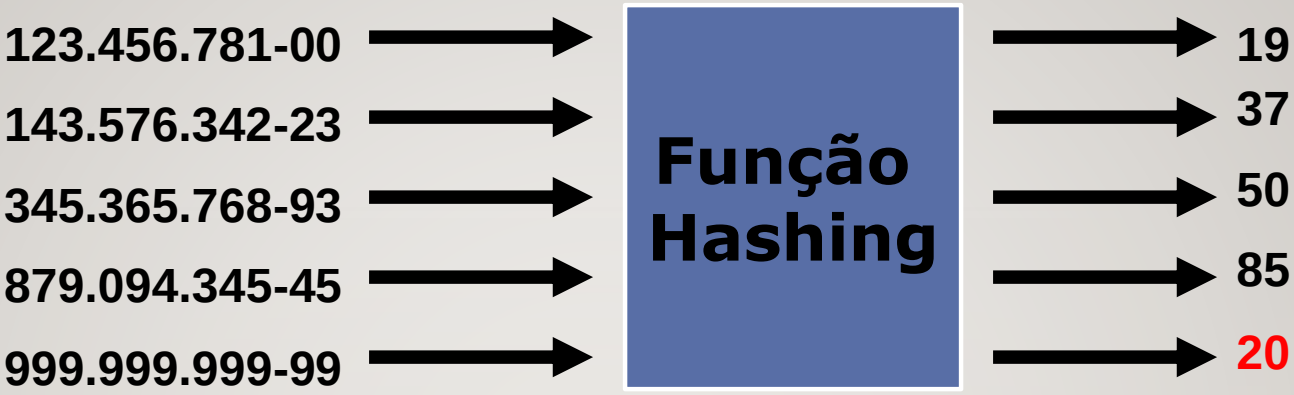
4 INTRODUÇÃO

- O ideal é realizar todas as operações em tempo constante O(1)

- As Tabelas Hash oferecem tempo constante por operação (em média)
  - No pior caso o tempo de processamento é proporcional ao tamanho de uma lista de tamanho k.

5 TABELAS HASH

- Também chamado de Tabela de dispersão.
- A Tabela Hash pode ser representada por um vetor, em que cada posição representa um endereço.
- Elementos armazenados: valor-chave
- A ideia central do hash é utilizar uma função, aplicada sobre a informação (chave), para retornar o índice (endereço) onde a informação deve ser armazenada.



19	**123.456.781-00; Fulano; Av. Canal. Nº 45.**
20	
…	
37	**143.576.342-23; Ciclano; Rua Celso Oliva. Nº 27.**
…	
50	**345.365.768-93; Beltrano; Av. Atlântica. S/N.**
…	
85	**879.094.345-45 ; José Maria; Rua B. Nº 100.**
…	

7 FUNÇÃO DE HASHING

- Também chamada de Função de espalhamento ou de dispersão.
- É responsável por gerar um endereço a partir de uma determinada chave
- Executam a transformação do valor de uma chave em um endereço, pela aplicação de operações aritméticas e/ou lógicas

$$f: C \rightarrow E$$

  - f: função de cálculo de endereço
  - C: espaço de valores da chave
  - E: espaço de endereçamento

8 FUNÇÕES DE HASHING

- Seja
  - Um conjunto C de chaves a serem armazenadas
  - Uma tabela de tamanho m
  - E dado $x \in C$

- A função hash calcula o endereço onde armazenar x

- A função de hashing tem como objetivo reduzir o espaço de endereço para armazenar C

$$m \leq |C|$$

9 FUNÇÕES DE HASHING

- A implementação da função Hash tem influência direta na eficiência das operações sobre a tabela Hash.
  - Pois ela é responsável por distribuir as informações pela tabela.

- Divisão
- Compressão de chaves alfanuméricas
- Multiplicação
- Enlaçamento
  - Deslocado
  - Limite
- Função Meio-Quadrado
- Extração
- Transformação da Raiz

10 MÉTODO DA DIVISÃO

- Na utilização do método da divisão devem ser evitados potência de 2 como valores de m

$$m \neq 2^n$$

- A melhor escolha para m é um número primo não muito próximo de uma potência de 2.

MÉTODO DA DIVISÃO

- A chave x é mapeada em um dos m endereços da tabela da seguinte forma:
  - É calculado o resto da divisão de x por m
  - $h(x) = x \; mod \; m$

O paradoxo do aniversário (Feller, 1968)

- ...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%.

13 COLISÃO

- Quando 2 ou mais chaves são mapeadas, pela função hash, para o mesmo endereço, diz-se que ocorreu uma colisão.

- Uma vez que colisões são teoricamente inevitáveis, é necessário ter técnicas para reduzi-las ou tratá-las

14 UMA BOA FUNÇÃO HASH

- As funções hash, em geral, deveriam satisfazer às seguintes condições:
  - Pequeno número de Colisões
  - Ser facilmente computável
  - Ser uniforme

15 TRATAMENTO DE COLISÕES

- Um bom método de resolução de colisões é essencial, não importando a qualidade da função de hashing.
- Há diversas técnicas de resolução de colisão.
- As técnicas mais comuns podem ser enquadradas em duas categorias:
  - Endereçamento Aberto (Rehash);
  - Encadeamento.

TRATAMENTO DE COLISÕES POR ENCADEAMENTO

- As chaves são armazenadas em estruturas encadeadas fora da tabela Hash
- Cada endereço da tabela passa a apontar para o inicio de uma lista encadeada
- Os elementos inseridos na tabela podem ser ordenados de diversas formas:
  - Pelo valor (Ex: Insertion Sort)
  - Pela frequência de acesso
  - Por ordem de inserção

FUNÇÕES DE HASHING

- A maioria das funções de hashing assumem que as chaves são números naturais
- Caso a chave não seja um número, uma operação deve ser realizada mapeando-a em um número
  - Por exemplo, soma do valor ASCII de cada letra
- Ex: "Jose Maria"

J	o	s	e		M	a	r	i	a
74	111	115	101	255	76	97	114	105	97

- 74+111+115+101+255+76+97+114+105+97 = 1221

18 COMPRESSÃO DE CHAVES ALFANUMÉRICAS

- Utilizado quando as chaves são alfanuméricas
- Chaves são representadas utilizando sua representação interna

B	R	A	S	I	L
C2	D2	C1	D3	C9	CC
11000010	11010010	11000001	11010011	11001001	11001100

Representação hexadecimal e binária do código ASCII

- Muitas vezes surge a necessidade de comprimir uma chave, dado sua representação numérica excessivamente grande
- Uma idéia útil é utilizar o operador XOR (ou exclusivo)

19 COMPRESSÃO DE CHAVES ALFANUMÉRICAS

- Utilização do XOR

A	B	A xor B
0	0	0
0	1	1
1	0	1
1	1	0

- Comprimindo a chave BRASIL para um total de 16 bits:

```
(BR)  1100001011010010
(AS)  1100000111010011
(IL)  1100100111001100
xor   1100101011001101 = 51917(10)
```

- Problema do método:
- Chaves com permutações de grupos de letras/dígitos irão produzir colisões.
- BRASIL, BRILAS, ASBRIL, ILBRAS irão produzir o mesmo endereço
- Uma forma de contornar esse problema é executar uma operação de rotação de bits de cada grupo
  - Rotação de 1 bit no segundo grupo
  - Rotação de 2 bits no terceiro grupo
  - ...
- Outra forma é atribuir peso para cada letra
  - Ex: 1ª letra = peso 1; 2ª letra = peso 2...

21 ANÁLISE DE COMPLEXIDADE

- Considere o Método de Divisão com tratamento de colisões por encadeamento:
  - A Operação de inserção de um novo elemento envolve apenas uma operação aritmética e atualizações de ponteiros
    - Tempo de execução $O(1)$
  - As operações de busca e pesquisa envolvem apenas uma operação aritmética e atualizações de ponteiros
    - No pior caso: $O(k)$

APLICAÇÕES DE HASHING

- Exemplos:
  - Integridade de Dados
  - Criptografia
  - Compactação de Dados
  - Compiladores

Trabalho

- Crie um arquivo hash com registros cidades do estado de Minas Gerais. A chave de cada registro será o nome da cidade e não são necessários outros campos para este exercício. Utilize o arquivo Cidades.txt como entrada do sistema.
  - Implemente uma função hash() que utiliza alguma combinação dos códigos ASCII das letras do nome. Execute o hash() várias vezes, cada vez utilizando um número diferente de registros e produzindo as seguintes estatísticas para cada execução:
    - O número de colisões;
    - O número de endereços com 0,1,2,3,...10, ou mais de 10 cidades associadas.

Trabalho

- Discuta os resultados de seu experimento em termos de efeitos da escolha de diferentes quantidades de registros e como eles se relacionam com o resultado que você poderia esperar de uma distribuição aleatória.
- Implemente o método da divisão e o segundo método sorteado em aula.
- Apresentação do trabalho:
  - Código
  - Resultados
  - Explicação do funcionamento do método

EXTRA

- URI
  - 1256; 1257

FIM DA AULA 9

Próxima aula:

Grafos