

HASHING

Notas de aula da disciplina IME 04-10820
Estruturas de Dados I

Paulo Eustáquio Duarte Pinto
(pauloedp arroba ime.uerj.br)

setembro/2018

HASHING

Árvores de Busca:

Atualização/Busca - caso médio: $O(\log n)$

Atualização/Busca - pior caso: $O(\log n)$

Pergunta: Pode-se melhorar os dados anteriores?

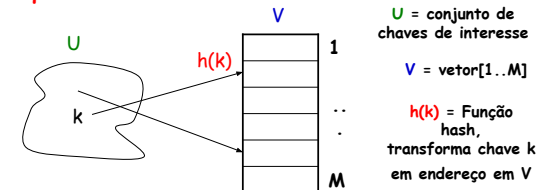
Resposta: Pode-se melhorar o primeiro, ao custo de piorar o segundo.

HASHING

Pergunta: Como tentar buscar mais rapidamente?

Resposta: usando-se um vetor e fazendo a transformação chave/endereço, para localizar a chave no vetor.

Esquema:

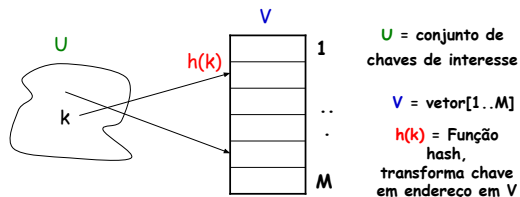


HASHING

Função hash - $h(k): U \rightarrow V$.

Pergunta: $h(k)$ pode ser uma função injetora?

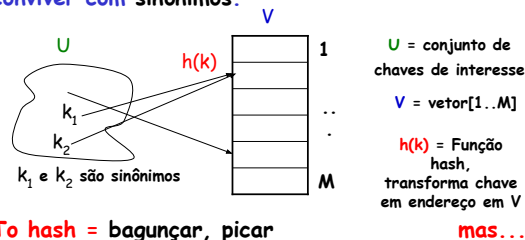
Resposta: quase nunca... (pensar no paradoxo do aniversário...)



HASHING

Função hash - $h(k): U \rightarrow V$ - Sinônimos

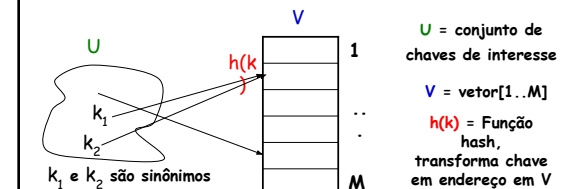
Já que é tão difícil funções injetoras, tem-se que conviver com sinônimos.



To hash = bagunçar, picar

HASHING - Função hash - Propriedades

- 1) Cálculo rápido.
- 2) Gerar endereços na faixa 1..M (0..M-1)
- 3) Gerar distribuição uniforme de sinônimos
- 4) Não preservar propriedades das chaves



HASHING - Função hash

a) Resto da Divisão:

$$h(k) = k \bmod M + 1.$$

a.1) cálculo rápido

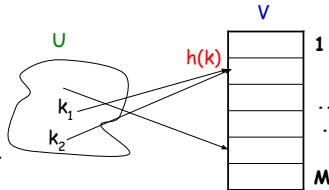
a.2) gera endereços na faixa 1..M

a.3) gera distribuição uniforme de sinônimos

(é periódica com período M)

0..1
1..2
M-1..M
M..1 ...

a.4) preserva sequência(!?)



HASHING - Função hash

b) Ou Exclusivo:

$$h(k) = k_a \oplus k_b + 1.$$

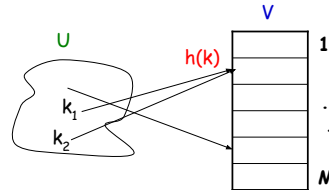
b.1) cálculo rápido

b.2) gera endereços na faixa 1..M=2^p

b.3) gera distribuição uniforme de sinônimos

Exemplo

b.4) não preserva sequência(!?)



HASHING - Função hash

c) Função de Multiplicação:

$$h(k) = (k \cdot b \bmod R) \bmod M + 1.$$

c.1) cálculo rápido

c.2) gera endereços na faixa 1..M

c.3) pode gerar distribuição uniforme de sinônimos.

Depende de R, b, M.

$$R = 2^{16}, M = 613, b = 31415821$$

1	171
2	341
3	566
4	123
5	293
6	518
7	75
8	245
9	440
10	27

c.4) não preserva sequência.

HASHING - Função hash

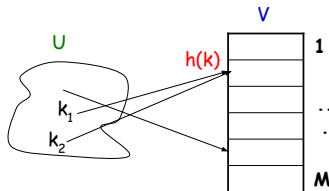
d) Outras Funções hash

d.1) Dobradura

d.2) Seleção de partes da chave

d.3) Metade do quadrado

d.4) Segredos industriais...



HASHING - Tratamento de sinônimos

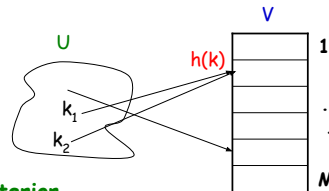
MÉTODOS

1) Encadeamento Exterior

2) Encadeamento Interior (Listas coligadas)

3) Endereçamento aberto

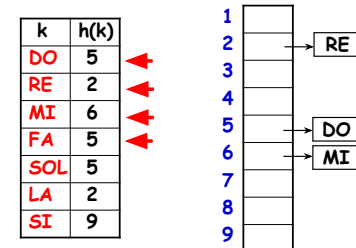
4) Endereçamento aberto com duplo hashing



HASHING - Tratamento de sinônimos

1) Encadeamento Exterior

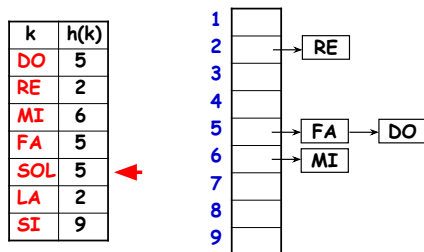
Idéia: criar uma lista encadeada de sinônimos para cada endereço do vetor.



HASHING - Tratamento de sinônimos

1) Encadeamento Exterior

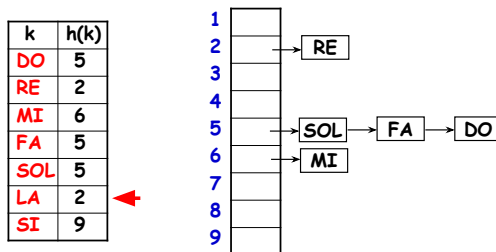
Idéia: criar uma lista encadeada de sinônimos para cada endereço do vetor.



HASHING - Tratamento de sinônimos

1) Encadeamento Exterior

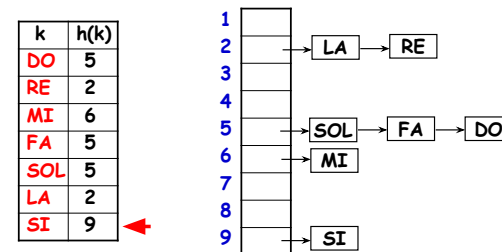
Idéia: criar uma lista encadeada de sinônimos para cada endereço do vetor.



HASHING - Tratamento de sinônimos

1) Encadeamento Exterior

Idéia: criar uma lista encadeada de sinônimos para cada endereço do vetor.



HASHING - Tratamento de sinônimos

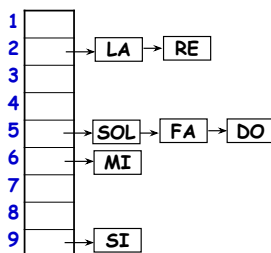
1) Encadeamento Exterior

Busca(k, i, p):

$p \leftarrow V[i];$
 enquanto $((p \neq \text{nulo}) \text{ e } (p.c \neq k)):$
 $p \leftarrow p.\text{prox};$

Inserção(k, p):

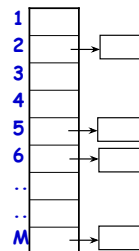
$i \leftarrow h(k);$ Busca(k, i, p);
 se $(p = \text{nulo}):$
 $p \leftarrow \text{Nó}();$
 $p.c \leftarrow k; p.\text{prox} \leftarrow V[i];$
 $V[i] \leftarrow p$
 senão:
 $p \leftarrow \text{nulo}$



HASHING - Tratamento de sinônimos

1) Encadeamento Exterior - Análise

Melhor caso médio: $NC = 1$

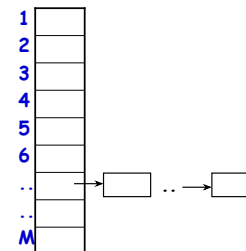


HASHING - Tratamento de sinônimos

1) Encadeamento Exterior - Análise

Melhor caso médio: $NC = 1$

Pior caso médio: $NC = N$



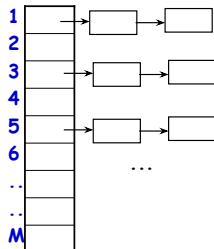
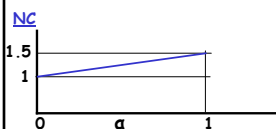
HASHING - Tratamento de sinônimos

1) Encadeamento Exterior - Análise da Busca

Melhor caso médio: $NC = 1$

Pior caso médio: $NC = N$

Caso médio (média): $NC = N(1+\alpha/2)$
 $\alpha = N/M = \text{fator de carga}$

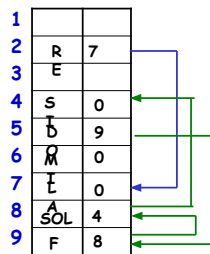


HASHING - Tratamento de sinônimos

2) Encadeamento Interior (listas coligadas)

Idéia: criar listas encadeadas de sinônimos para cada endereço do vetor, no próprio vetor.

k	h(k)
DO	5
RE	2
MI	6
FA	5
SOL	5
LA	2
SI	9



HASHING - Tratamento de sinônimos

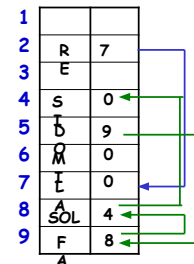
2) Encadeamento Interior (listas coligadas)

Busca(k, ant, p):
 $p \leftarrow h(k); \text{ ant} \leftarrow p;$
 Enquanto $(V[p].prox \neq \text{Nulo})$ e $(V[p].c \neq k):$
 $\text{ant} \leftarrow p; \quad p \leftarrow V[p].prox;$
 Fe;
 Se $(V[p].c \neq k)$ Então $p \leftarrow \text{Nulo};$

Fim.

Inserção(k, p):
 Busca(k, ant, p);
 Se $(p = \text{Nulo})$ e $(n < M)$ Então:
 Se $(V[ant].c = ' ')$ Então
 $p \leftarrow ant; \quad V[p].c \leftarrow k;$
 Senão
 $j \leftarrow \text{Next}(ant); \quad V[j].c \leftarrow k;$
 $V[ant].prox \leftarrow j; \quad p \leftarrow j;$
 $n \leftarrow n+1;$
 Senão $p \leftarrow \text{Nulo};$

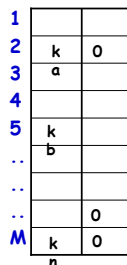
Fim.



HASHING - Tratamento de sinônimos

2) Encadeamento Interior - Análise

Melhor caso médio: $NC = 1$

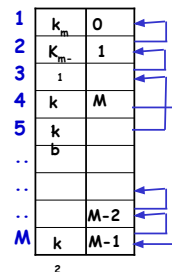


HASHING - Tratamento de sinônimos

2) Encadeamento Interior - Análise

Melhor caso médio: $NC = 1$

Pior caso médio: $NC = N$



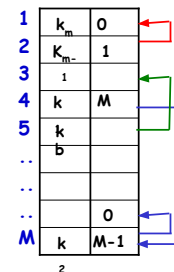
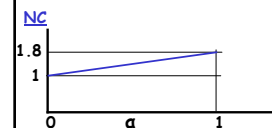
HASHING - Tratamento de sinônimos

2) Encadeamento Exterior - Análise da Busca

Melhor caso médio: $NC = 1$

Pior caso médio: $NC = N$

Caso médio (média): $NC = N(1+.8\alpha)$
 $\alpha = N/M = \text{fator de carga}$



HASHING - Tratamento de sinônimos

Exercício: Criar as tabelas hashing para Encadeamento Interior/Exterior, usando como entrada o

MIXSTRING,

$M = 9$ e

$h(\text{letra}) = \text{Pos}(\text{letra}) \bmod 9 + 1$.

Calcular NC para cada tabela.

HASHING - Tratamento de sinônimos

3) Endereçamento Aberto

Idéia: usar apenas um vetor, sem links e colocar sinônimos em outra posição. Uma das formas é procurar, circularmente, a próxima posição disponível.

k	h(k)
DO	5
RE	2
MI	6
FA	5
SOL	5
LA	8
SI	9

1	SI
2	RE
3	
4	
5	DO
6	MI
7	FA
8	SOL
9	LA

$$NC = \frac{(1+1+1+3+4+2+2)/7}{14/7}$$

HASHING - Tratamento de sinônimos

3) Endereçamento aberto

Busca(k, ant, p):

$p \leftarrow h(k); \text{ ant} \leftarrow p;$
 enquanto $((V[p] \neq \text{nulo}) \text{ e } (V[p] \neq k)):$
 $\text{ ant} \leftarrow p; \quad p \leftarrow p \bmod M + 1;$

se $(V[p] = \text{nulo}):$
 $p \leftarrow \text{nulo}$

Inserção(k, p):

Busca(k, ant, p)
 se $((p = \text{nulo}) \text{ e } (n < M-1)):$
 $V[\text{ant}] \leftarrow k; \quad p \leftarrow \text{ant};$
 senão:
 $p \leftarrow \text{nulo}$

1	SI
2	RE
3	
4	
5	DO
6	MI
7	FA
8	SOL
9	LA

HASHING - Tratamento de sinônimos

3) Endereçamento aberto - Análise

Melhor caso médio: $NC = 1$

1	
2	
3	ka
4	
...	
...	
...	kb
...	
M	kn

HASHING - Tratamento de sinônimos

3) Endereçamento aberto - Análise

Melhor caso médio: $NC = 1$

Pior caso médio: $NC = N$

1	
2	ka
3	kb
4	
...	
...	
...	
...	
M	kn

HASHING - Tratamento de sinônimos

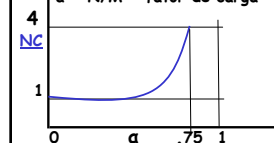
3) Endereçamento aberto - Análise

Melhor caso médio: $NC = 1$

Pior caso médio: $NC = N$

Caso médio (média): $NC = 4 p / \alpha = .75$

$\alpha = N/M = \text{fator de carga}$



1	
2	ka
3	kb
4	
...	
...	
...	
...	
M	kn

HASHING - Tratamento de sinônimos

4) Endereçamento Aberto com duplo hash

Idéia: usar uma segunda função para determinar os saltos na busca de espaço.

k	$h_1(k)$	$h_2(k)$	
DO	5	1	
RE	2	2	
MI	6	8	
FA	5	2	
SOL	5	4	
LA	8	8	
SI	9	4	

1	
2	RE
3	
4	SI
5	DO
6	MI
7	FA
8	LA
9	SOL

$NC = (1+1+1+2+2+1+2)/7 = 10/7$

HASHING - Tratamento de sinônimos

4) Endereçamento aberto com duplo hashing

Busca(k, ant, p):

$p \leftarrow h_1(k); \quad ant \leftarrow p; \quad s \leftarrow h_2(k);$
 Enquanto $(V[p] \neq \text{Nulo})$ e $(V[p] \neq k)$:
 $ant \leftarrow p; \quad p \leftarrow (p+s-1) \bmod M + 1;$
 Fe;
 Se $(V[p] = \text{Nulo})$ Então $p \leftarrow \text{Nulo};$
 Fim.

Inserção(k, p):

Busca(k, ant, p);
 Se $(p = \text{Nulo})$ e $(n < M-1)$ Então:
 $V[ant] \leftarrow k; \quad p \leftarrow ant;$
 Senão $p \leftarrow \text{Nulo};$
 Fim.

1	
2	SI
3	RE
4	
5	
6	DO
7	MI
8	FA
9	SOL
	LA

HASHING - Tratamento de sinônimos

4) Endereçamento Aberto com duplo hash

Requisito importante para $h_2(k)$:

k	$h_1(k)$	$h_2(k)$	
DO	5	1	
RE	2	2	
MI	6	8	
FA	5	2	
SOL	5	4	
LA	8	8	
SI	8	3	

1	
2	
3	RE
4	
5	
6	DO
7	MI
8	FA
9	LA

HASHING - Tratamento de sinônimos

4) Endereçamento Aberto com duplo hash

Requisito importante para $h_2(k)$: gerar números relativamente primos a M!

k	$h_1(k)$	$h_2(k)$	
DO	5	1	
RE	2	2	
MI	6	8	
FA	5	2	
SOL	5	4	
LA	8	8	
SI	8	6	

1	
2	
3	RE
4	
5	
6	DO
7	MI
8	FA
9	LA

Exemplos:
 a) M ímpar e $h_2(k) = 2^{k \bmod M}$
 b) M primo e $h_2(k) = k \bmod M-1 + 1$

HASHING - Tratamento de sinônimos

4) Endereçamento aberto com duplo hash - Análise

Melhor caso médio: $NC = 1$

1	
2	
3	ka
4	
...	
...	
...	kb
...	
...	kn

HASHING - Tratamento de sinônimos

4) Endereçamento aberto com duplo hash - Análise

Melhor caso médio: $NC = 1$

Pior caso médio: $NC = N$

1	
2	
3	kb
4	
...	
...	
...	
...	kn

HASHING - Tratamento de sinônimos

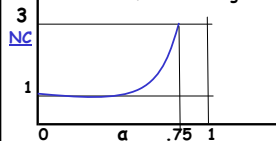
4) Endereçamento aberto com duplo hash - Análise

Melhor caso médio: $NC = 1$

Pior caso médio: $NC = N$

Caso médio (média): $NC = 3$ p/ $\alpha = .75$

$\alpha = N/M =$ fator de carga



1	
2	
3	kb
4	
..	
..	
..	
..	
M	kn

HASHING - Tratamento de sinônimos

Exercício: Criar as tabelas hashing para Endereçamento aberto simples e com duplo hash, usando como entrada o

MIXSTRING,

$M = 9$ e

$h(\text{letra}) = \text{Pos}(\text{letra}) \bmod 9 + 1$.

Calcular NC para cada tabela.

HASHING

Fim