Fundamentos de Hashing

O que é Hashing?

Uma função de espalhamento (função *hash*) h(k) transforma uma chave k em um endereço.

- **Este endereço é usado como a base para o armazenamento e recuperação de registros.**
- **É similar a uma indexação, pois associa a chave ao endereço relativo do registro.**

- Diferenças do hash e a indexação
  - **no espalhamento os endereços parecem ser aleatórios - não existe conexão óbvia entre a chave e o endereço, apesar da chave ser utilizada no cálculo do endereço**
  - **no espalhamento duas chaves podem levar ao mesmo endereço (colisão) – portanto as colisões devem ser tratadas.**

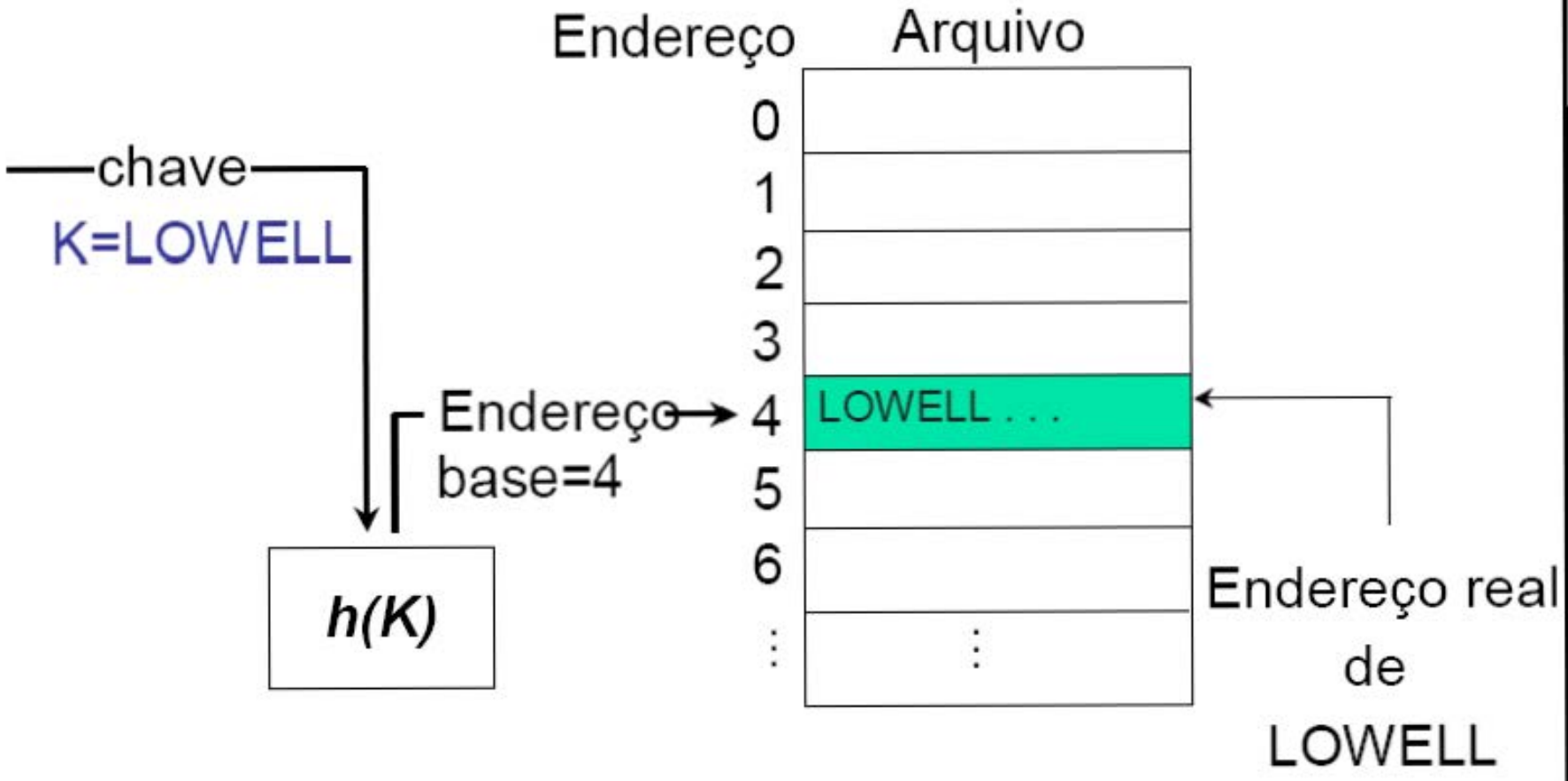
Hashing: Exemplo

Suponha que foi reservado espaço para manter 1.000 registros e considere a seguinte h(K):

- Obter as representações ASCII dos dois primeiros caracteres do sobrenome;
- multiplicar estes números e usar os três dígitos menos significativos do resultado para servir de endereço.

Name	Código ASCII para as 2 primeiras letras	Produto	Endereço
BALL	66 65	66 X 65 = 4.290	290
LOWELL	76 96	76 X 96 = 6.004	004
TREE	84 82	84 X 82 = 6.888	888

Hashing: Exemplo



Hashing: Colisões

Maneiras de reduzir o número de colisões:

- Utilizar um algoritmo que distribua os registros relativamente por igual entre os endereços disponíveis.
- Utilização de mais memória: é relativamente fácil achar um bom algoritmo quando se pode espalhar poucos registros em muitos endereços, em outras palavras, desperdiçar muito espaço.
- Utilização de mais de um registro por endereço: Nesse caso, o endereço (chamado de cesto - *bucket*) tem espaço para armazenar vários registros.

Algoritmo para espalhamento
Exemplo

A função apresentada a seguir é bastante eficiente na maioria dos casos e é facilmente modificável para acomodar ajustes que considerem a relevância das chaves

1. **Representar a chave numericamente (caso a chave não seja numérica). Por exemplo, usar os códigos ASCII dos caracteres (todos) para formar um número. Por exemplo:**

LOWELL$$$$$ = 76 79 87 69 76 76 32 32 32 32 32 32 (considera+ 6 brancos)

Algoritmo para espalhamento
Exemplo

2. **Subdividir o número em partes e somar as partes** (*fold and add*)

Exemplo de Subdivisão: 76 79 | 87 69 | 76 76 | 32 32 | 32 32 | 32 32
Soma das partes: 7679 + 8769 + 7676 + 3232 + 3232 + 3232 = 33820

Suponha que utilizemos um endereço de 15 bits → o limite será 32767
se aplicássemos a soma direta das partes
7679 + 8769 + 7676 + 3232 + 3232 + 3232 = 33820 > 32767 (overflow!)
O maior endereço seria o "ZZ" → 9090
O maior resultado permitido será 32767 - 9090 = 19937

→ 33820 mod 19937 = 13883

Distribuição de registros entre endereços

Predizendo a distribuição de registros

- **probabilidade de que um mesmo endereço tenha x registros**

Seja N = número de endereços disponíveis
r = número de registros armazenados

p(x) a probabilidade de que um mesmo endereço tenha x registros associados depois da função ter sido aplicada aos N registros.

- **Fórmula de Poisson aplicada a Hashing**

$$p(x) = \frac{(r/N)^x \, e^{-r/N}}{x!}$$ (distribuição de Poisson)

Distribuição de registros entre endereços

Predizendo a distribuição de registros

$$p(x) = \frac{(r/N)^x \, e^{-r/N}}{x!}$$ (distribuição de Poisson)

Exemplo: Considere 1.000 endereços e 1.000 registros

$p(0) \cong 1^0 \, e^{-1} / 0! \cong 1/ 2.71 \cong 0.368$ (prob. de um endereço ter nenhuma chave associada)

$p(1) \cong 1^1 \, e^{-1} / 1! \cong 1/ 2.71 \cong 0.368$ (prob. de um endereço ter uma chave associada)

$p(2) \cong 1^2 \, e^{-1} / 2! \cong 1/ (2 \times 2.71) \cong 0.184$ (prob. de um endereço ter 2 chaves associadas)

$p(3) \cong 1^3 \, e^{-1} / 3! \cong 1/ (3 \times 2.71) \cong 0.061$ (prob. de um endereço ter 3 chaves associadas)

Distribuição de registros entre endereços

Predizendo a distribuição de registros

- Número esperado de endereços com x registros

Seja **N = número de endereços disponíveis**

r = número de registros armazenados

p(x) a prob. de que um mesmo endereço tenha x registros associados

 # esperado de endereços com x registros = N . p(x)

Distribuição de registros entre endereços

Reduzindo colisões com o acréscimo de endereços

- **Densidade de ocupação: Razão entre o número de registros a serem armazenados (r) e o número de espaços de endereçamento disponíveis (N, assumindo um registro por endereço).**

$$\text{Densidade de ocupação} = \frac{\#\ \text{de registros}}{\#\ \text{de espaços}} = \frac{r}{N}$$

- **A densidade de ocupação dá uma medida da quantidade de espaço do arquivo que está sendo de fato utilizada, e é o único valor necessário para avaliar o desempenho de um espalhamento, assumindo que a função de espalhamento produz uma distribuição razoavelmente aleatória dos registros.**
- **Exemplo:**

Se temos 500 registros para espalhar por 1000 endereços

densidade de ocupação = 500/1000=0,5 → 50%

Armazenamento de vários registros por endereço

Cestos (*Buckets*)

- Cestos permitem armazenar mais de um registro em um mesmo endereço. Por exemplo, um cesto de tamanho dois permite alocar dois regitros, e só existe colisão quando um terceiro registro precisar ser alocado no mesmo endereço.

- Note que a fórmula da densidade de ocupação muda para r/bN, onde b é número de registros em um cesto.

- O uso de cestos melhora consideravelmente ao desempenho do espalhamento, porque aumenta a densidade de ocupação e diminui o número de colisões.

- Cuidados na escolha do tamanho dos cestos
  Evite cestos maiores que uma trilha, estude alocar vários cestos a uma mesma trilha; o menor número pode ser o de um cluster.

Armazenamento de vários registros por endereço: Cestos (*Buckets*)

- ***Cesto*** → bloco de registros que compartilham o mesmo endereço

Chave	Endereço da chave
Green	30
Hall	30
Jerk	32
King	33
Land	33
Marx	33
Nutt	33

Endereço do cesto	Conteúdo do cesto		
30	Green ...	Hall ...	
31			
32	Jenks ...		
33	King...	Land...	Marks...

O efeito dos cestos no desempenho

$$\text{Densidade de ocupação} = \frac{\#\ \text{de registros}}{\#\ \text{de espaços}} = \frac{r}{b.N}$$

Exemplo. Vamos comparar as seguintes alternativas:

1. armazenar 750 registros de dados em um arquivo hash com N=1000 endereços, cada um contendo 1 registro.
2. armazenar 750 registros de dados em um arquivo hash com N=500 endereços de cestos , cada cesto com dois registros.

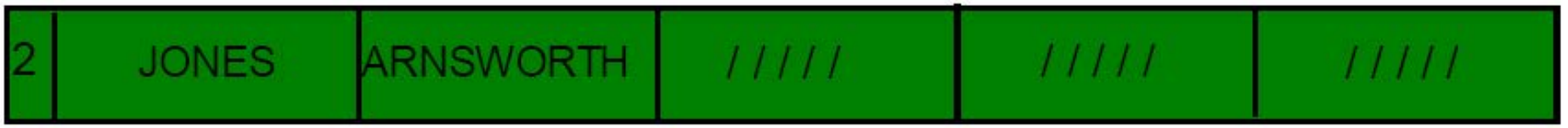
- Nos dois casos a densidade de ocupação é 75% (750 / 1000)
- Calculando r/N
  - No primeiro caso r/N=0,75
  - No segundo caso r/N=1,5
- Estimando as probabilidades

	p(0)	p(1)	p(2)	p(3)	p(4)
r/N=0,75 (b=1)	0.472	0.354	0.133	0.033	0.006
R/N=1,5 (b=2)	0.223	0.335	0.251	0.126	0.047

Implementação de cestos

Contador de colisões =< tamanho do cesto

cesto vazio



Cesto com duas entradas

Cesto cheio