

Algoritmos e Estruturas de Dados III

Aula 6.2 - Hashing Dinâmico

Prof. Hayala Curto
2022



PUC Minas

Hash Dinâmico

Definições



PUC Minas

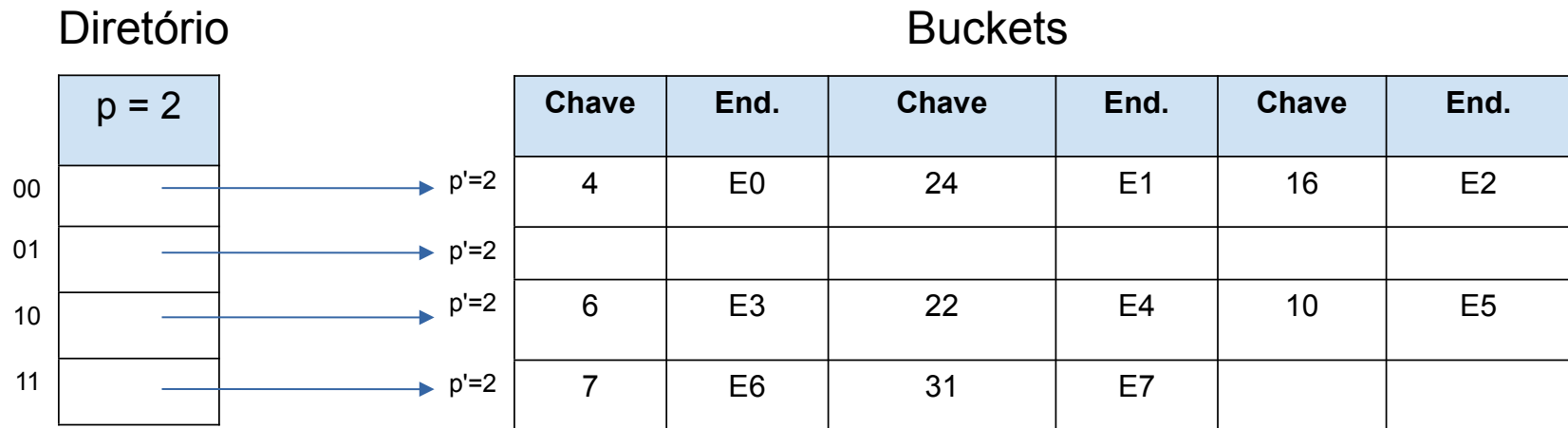
Hashing – Hash Dinâmico

- Quando o arquivo de dados cresce ou diminui com frequência (muitas inclusões e exclusões), o índice também precisará ser ajustado.
- Uma tabela hash estática, para crescer, **precisa reposicionar** todos os registros.
- Basicamente, refazer todo o índice!

Hashing – Hash Dinâmico

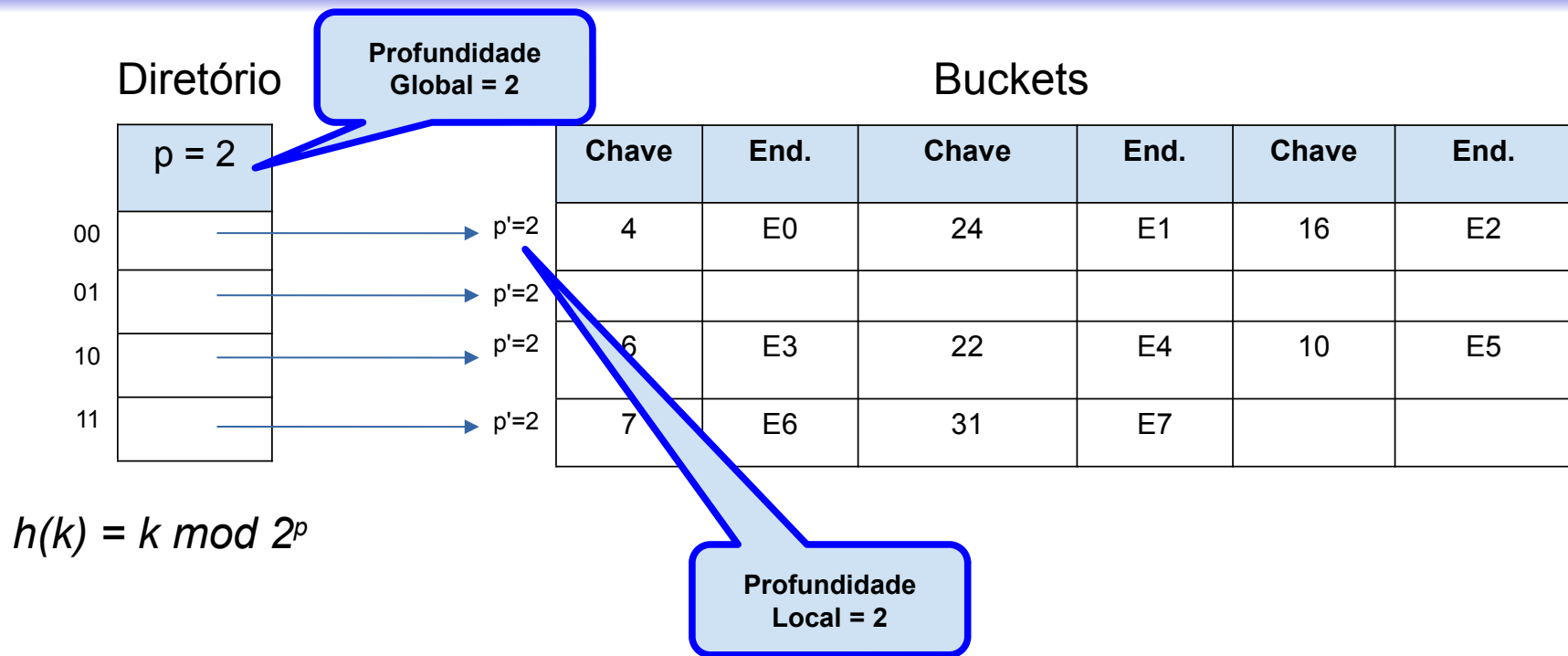
- Uma tabela hash extensível é uma tabela hash em que apenas alguns registros afetados (aqueles do bucket) precisam ser reposicionados.
- O diretório **sempre** tem tamanho equivalente a uma potência de dois

Hashing – Hash Dinâmico

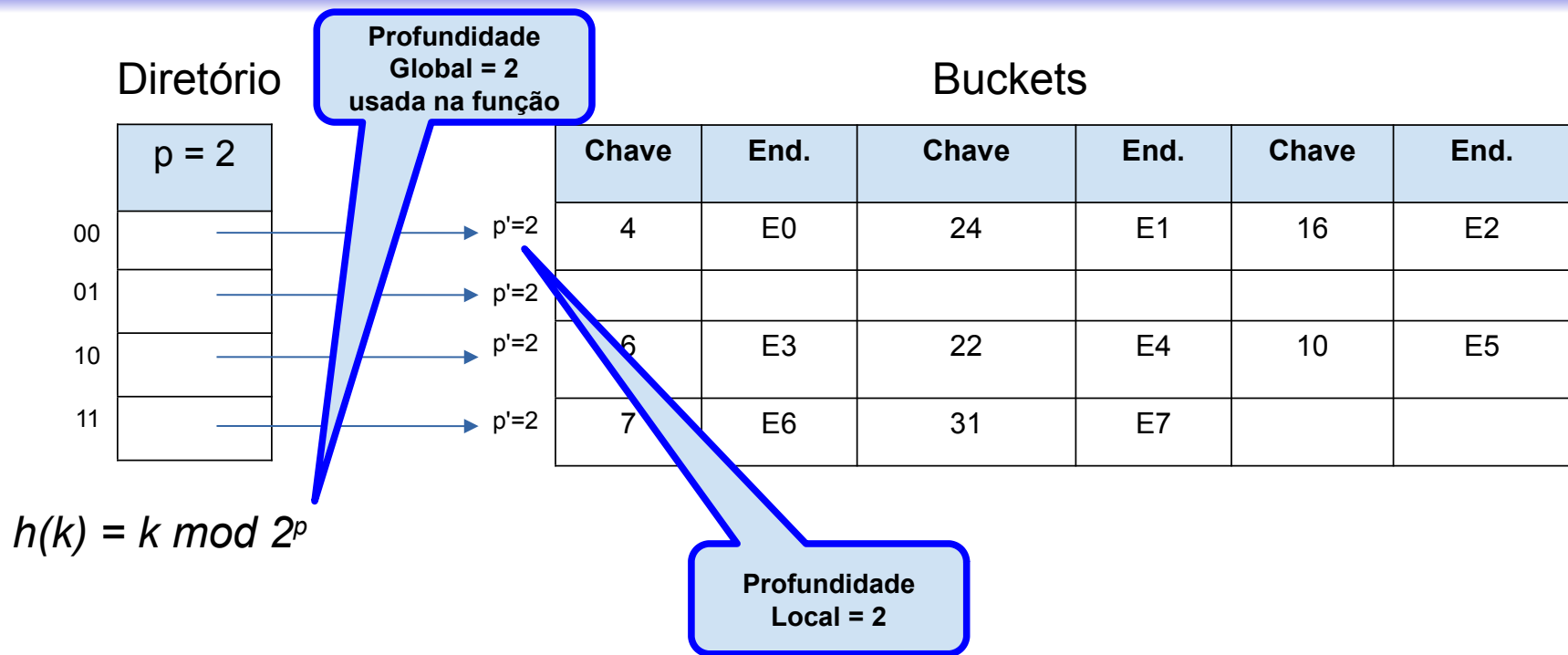


$$h(k) = k \bmod 2^p$$

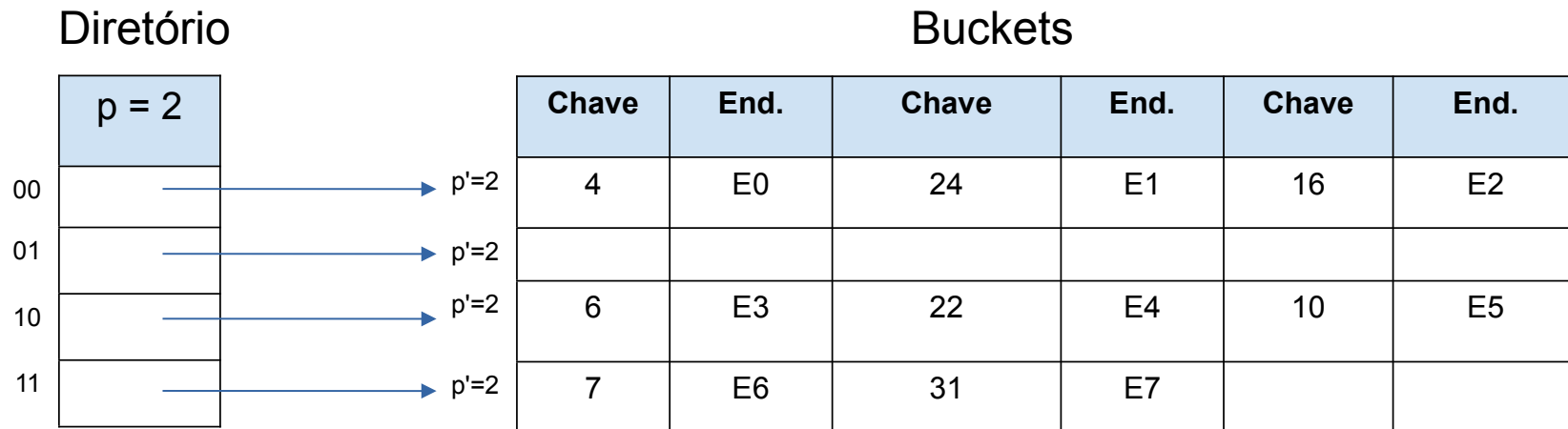
Hashing – Hash Dinâmico



Hashing – Hash Dinâmico



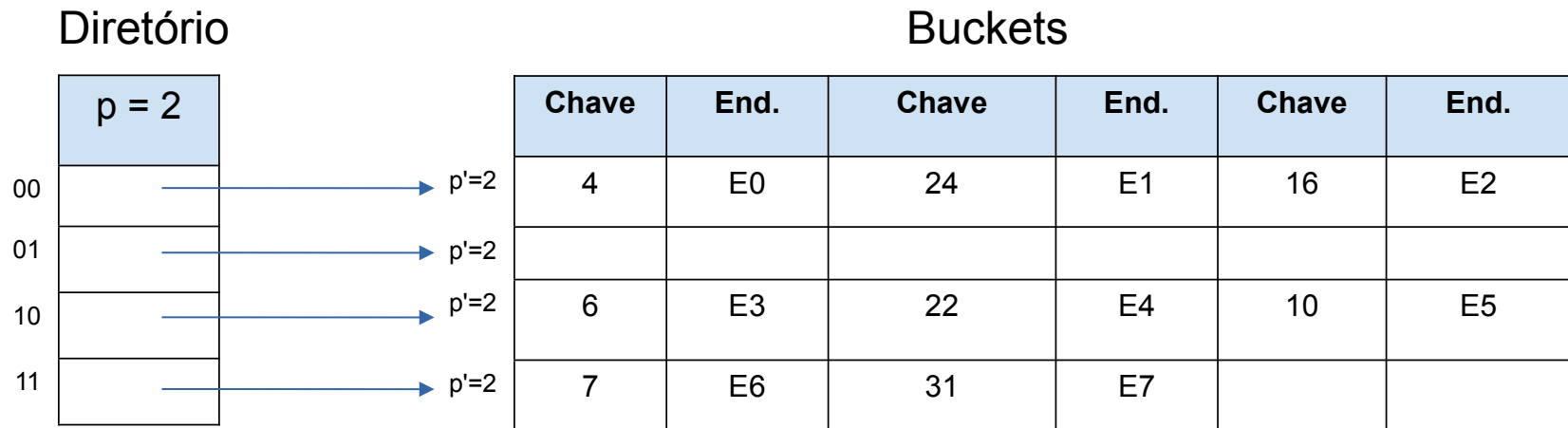
Hashing – Hash Dinâmico



$$h(k) = k \bmod 2^p$$

Função Hash
depende de p

Hashing – Hash Dinâmico



$$h(k) = k \bmod 2^p$$

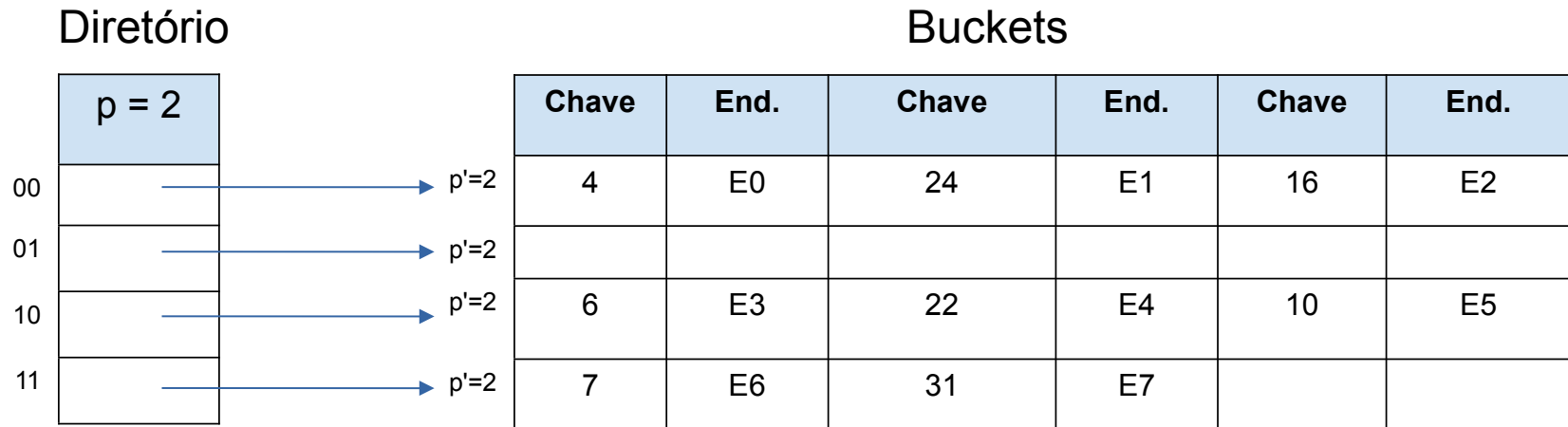
Buckets com
total de
elementos fixo

Inserção



PUC Minas

Inserindo chaves



$$h(k) = k \bmod 2^p$$

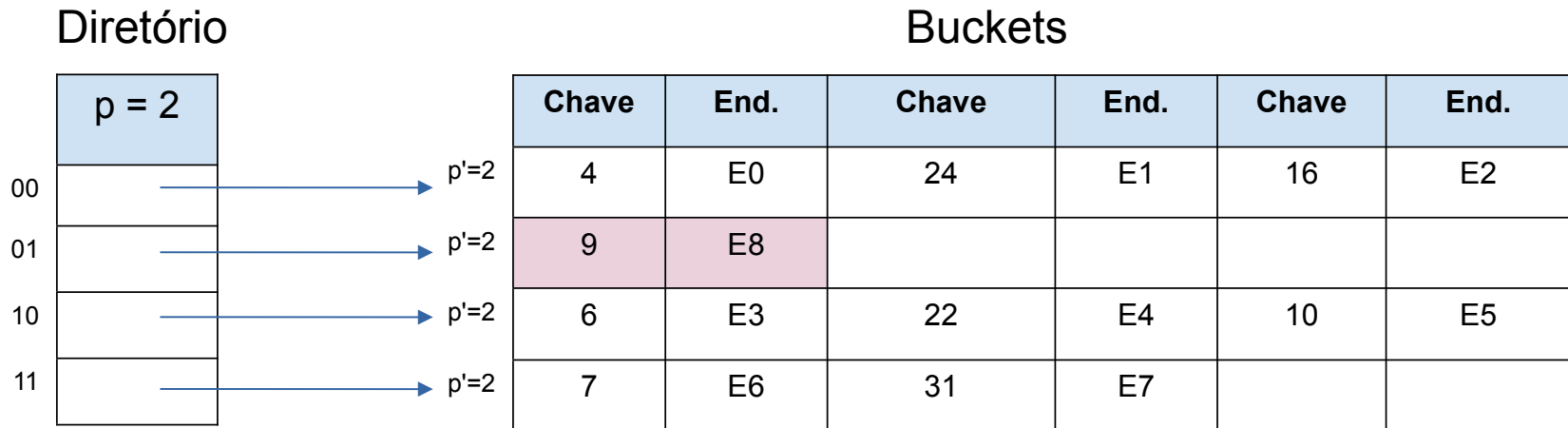
$$h(9) = 9 \bmod 2^2$$

$$h(9) = 9 \bmod 4$$

$$h(9) = 1$$

Adicionar chave 9

Inserindo chaves



$$h(k) = k \bmod 2^p$$

$$h(9) = 9 \bmod 2^2$$

$$h(9) = 9 \bmod 4$$

$$h(9) = 1$$

Adicionar chave 9

Inserindo chaves

Diretório

	p = 2	
00		p'=2
01		p'=2
10		p'=2
11		p'=2

Buckets

Chave	End.	Chave	End.	Chave	End.
4	E0	24	E1	16	E2
9	E8				
6	E3	22	E4	10	E5
7	E6	31	E7		

$$h(k) = k \bmod 2^p$$

$$h(20) = 20 \bmod 2^2$$

$$h(20) = 20 \bmod 4$$

$$h(20) = 0$$

Adicionar chave 20

Inserindo chaves

Diretório

$p = 2$	
00	$p'=2$
01	$p'=2$
10	$p'=2$
11	$p'=2$

Buckets

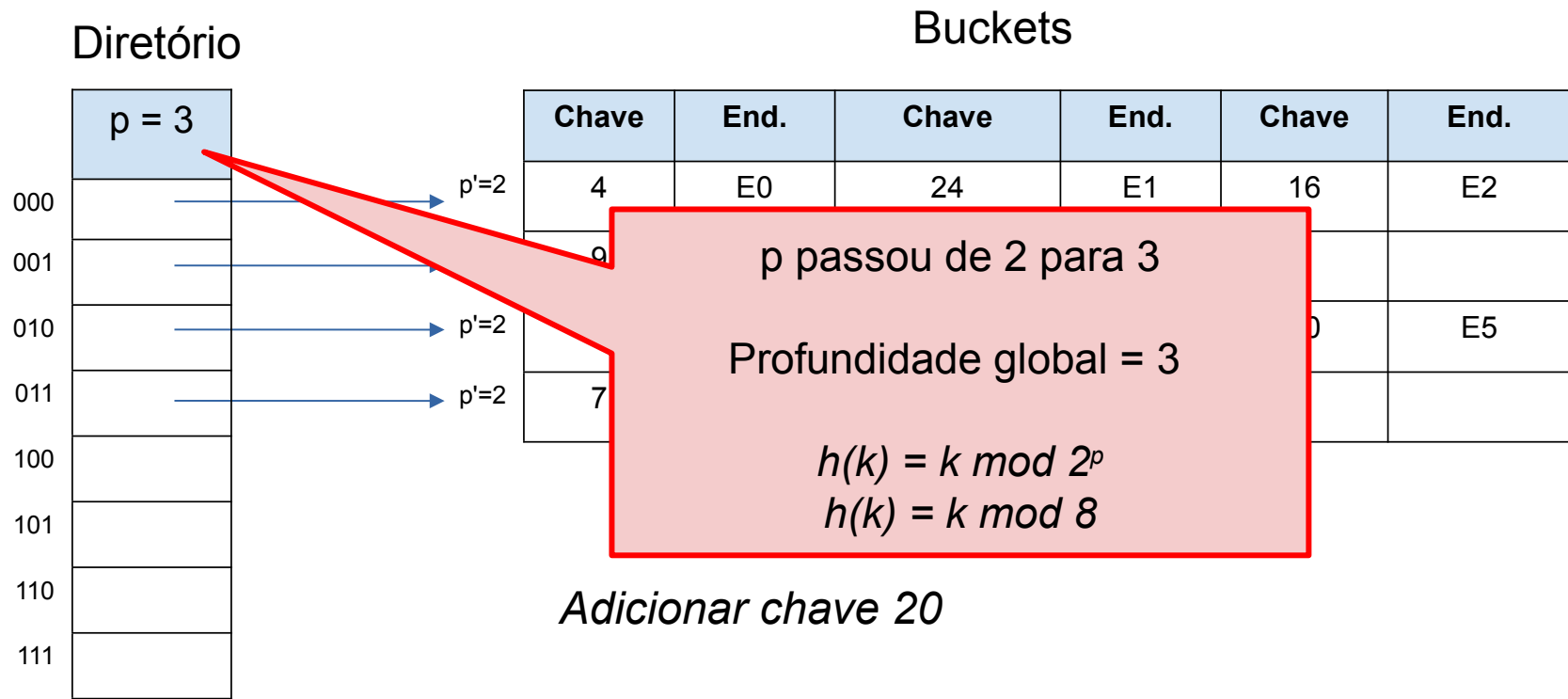
Chave	End.	Chave	End.	Chave	End.
4	E0	24	E1	16	E2
9	E8				
6	E3	22		10	E5
7	E6				

Não cabe no bucket 0!

Adicionar chave 20





$$\begin{aligned}h(k) &= k \bmod 2^p \\h(20) &= 20 \bmod 2^2 \\h(20) &= 20 \bmod 4 \\h(20) &= 0\end{aligned}$$

Inserindo chaves



Inserindo chaves

Diretório

	p = 3
000	
001	
010	
011	
100	
101	
110	
111	

p'=2

p'=2

p'=2

p'=2

Buckets

Chave	End.	Chave	End.	Chave	End.
4	E0	24	E1	16	E2
9					
6				0	E5
7					

Vamos ajustar os ponteiros

Adicionar chave 20

Inserindo chaves

Diretório

$p = 3$	
000	
001	
010	
011	
100	
101	
110	
111	

Buckets

Chave	End.	Chave	End.	Chave	End.
4	E0	24	E1	16	E2
9	E8				
6					E5
7					

Profundidade local = 2

Precisamos aumentar essa profundidade

Adiciona

Inserindo chaves

Diretório

	$p = 3$
000	
001	
010	
011	
100	
101	
110	
111	

$p'=2$

$p'=2$

$p'=2$

$p'=2$









Buckets

Chave	End.	Chave	End.	Chave	End.
4	E0	24	E1	16	E2
9	E8				
6	E3	22	E4	10	E5
7	E6	31	E7		

Adicionar chave 20

Inserindo chaves

Diretório

	$p = 3$
000	
001	
010	
011	
100	
101	
110	
111	

$p'=3$

$p'=2$

$p'=2$

$p'=2$

$p'=3$

Buckets

Chave	End.	Chave	End.	Chave	End.
4	E0	24	E1	16	E2
9	E8				
6	E3	22	E4	10	E5
7	E6	31	E7		

Adicionar chave 20

Inserindo chaves

Diretório

	$p = 3$
000	→ $p'=3$
001	→ $p'=2$
010	→ $p'=2$
011	→ $p'=2$
100	→ $p'=3$
101	
110	
111	

Buckets

Chave	End.	Chave	End.	Chave	End.
4	E0	24	E1	16	E2
9	E8				
6	E3	22	E4	10	E5
7	E6				

Precisamos organizar as chaves
4, 24, 16 e a 20

Adicionar

Inserindo chaves

Diretório

	$p = 3$
000	
001	
010	
011	
100	
101	
110	
111	

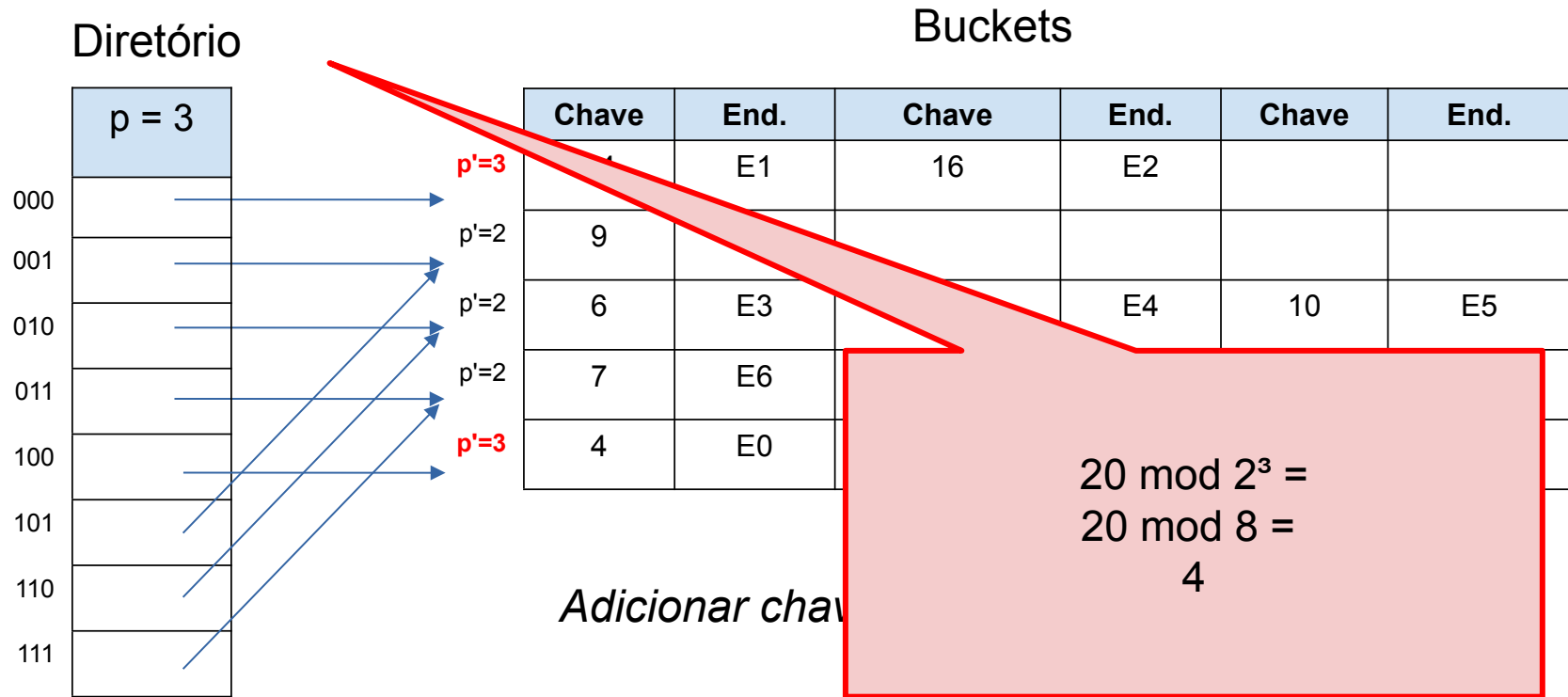
$p'=3$
 $p'=2$
 $p'=2$
 $p'=2$
 $p'=3$
 $p'=2$
 $p'=2$
 $p'=2$

Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
6	E3	22	E4	10	E5
7	E6	31	E7		
4	E0				

Adicionar chave 20

Inserindo chaves



Inserindo chaves

Diretório

$p = 3$	
000	
001	
010	
011	
100	
101	
110	
111	

Diagram illustrating the mapping of directory entries to buckets. Arrows show the mapping from directory entries to buckets:

- 000 → $p'=3$
- 001 → $p'=2$
- 010 → $p'=2$
- 011 → $p'=2$
- 100 → $p'=3$
- 101 → $p'=2$
- 110 → $p'=2$
- 111 → $p'=2$

Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
6	E3	22	E4	10	E5
7	E6	31	E7		
4	E0	20	E9		

Inserindo chaves

Diretório

	p = 3
000	
001	
010	
011	
100	
101	
110	
111	

Diagram illustrating the Directory structure with 8 slots (000 to 111). The top slot (000) is labeled p = 3. Arrows indicate pointers to buckets:

- 000 → p'=3
- 001 → p'=2
- 010 → p'=2
- 011 → p'=2
- 100 → p'=3
- 101 → p'=2
- 110 → p'=2
- 111 → p'=3

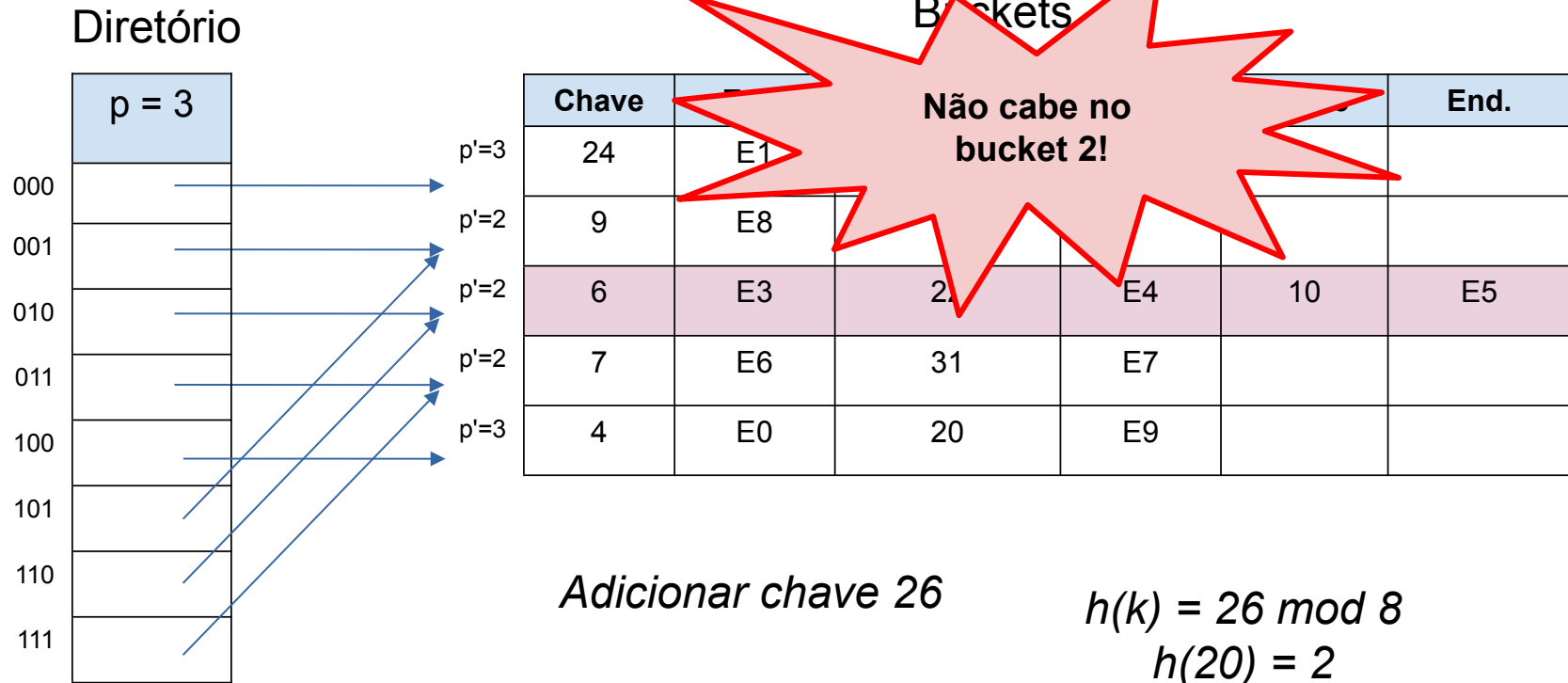
Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
6	E3	22	E4	10	E5
7	E6	31	E7		
4	E0	20	E9		

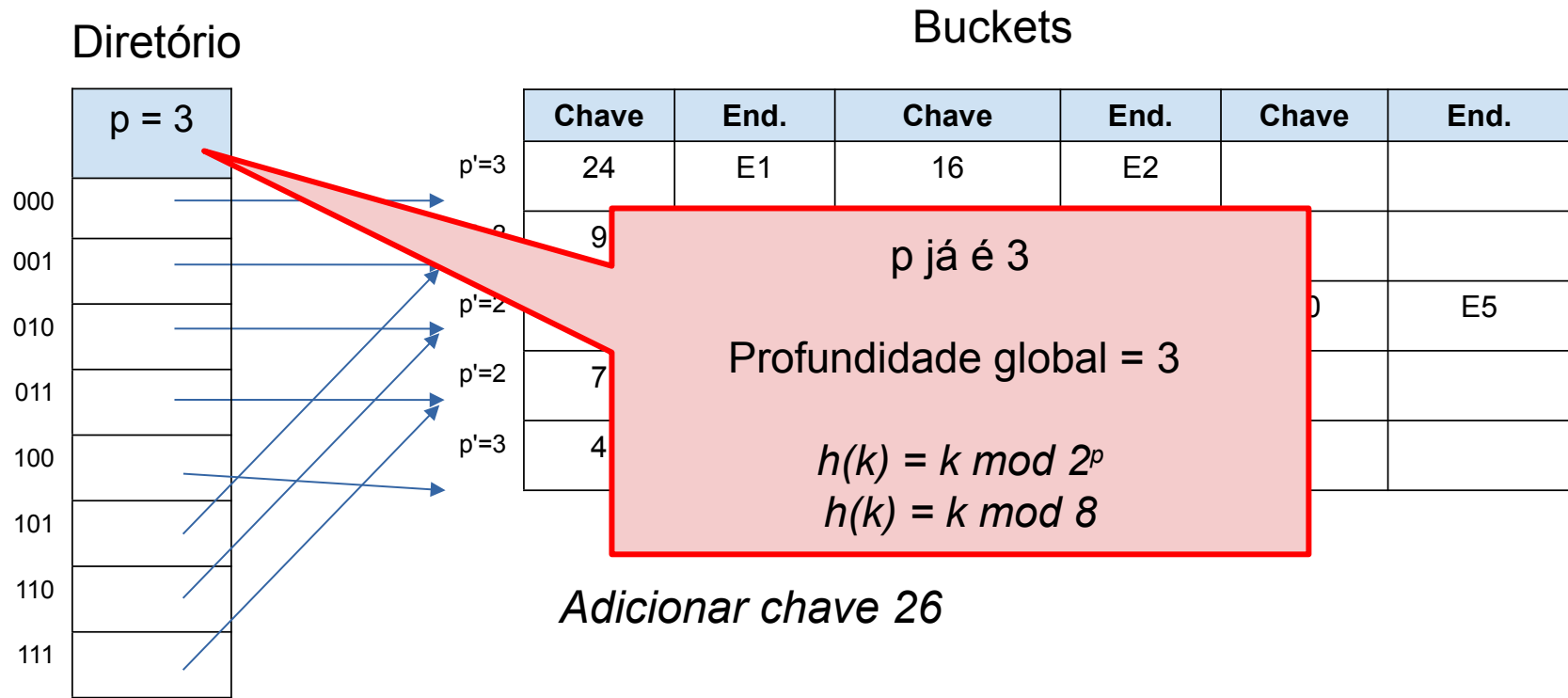
Adicionar chave 26

$$h(k) = 26 \bmod 8$$
$$h(26) = 2$$

Inserindo chaves



Inserindo chaves



Inserindo chaves

Diretório

	$p = 3$
000	
001	
010	
011	
100	
101	
110	
111	

$p'=3$

$p'=2$

$p'=2$

$p'=2$

$p'=3$

Chave	End.				
24	E1				
9					
6	E3				
7	E6	31	E7		
4	E0	20	E9		









Profundidade local = 2

Precisamos aumentar essa profundidade

Adicionar chave 26

Inserindo chaves

Diretório

	$p = 3$
000	
001	
010	
011	
100	
101	
110	
111	

Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
6	E3	22	E4	10	E5
7	E6	31	E7		
4	E0	20	E9		

Adicionar chave 26

Inserindo chaves

Diretório

	$p = 3$
000	
001	
010	
011	
100	
101	
110	
111	

Diagram illustrating the directory structure. The directory is a table with 8 rows (000 to 111) and 1 column ($p = 3$). Arrows point from the directory rows to the corresponding rows in the main table, indicating the mapping of directory entries to data entries.

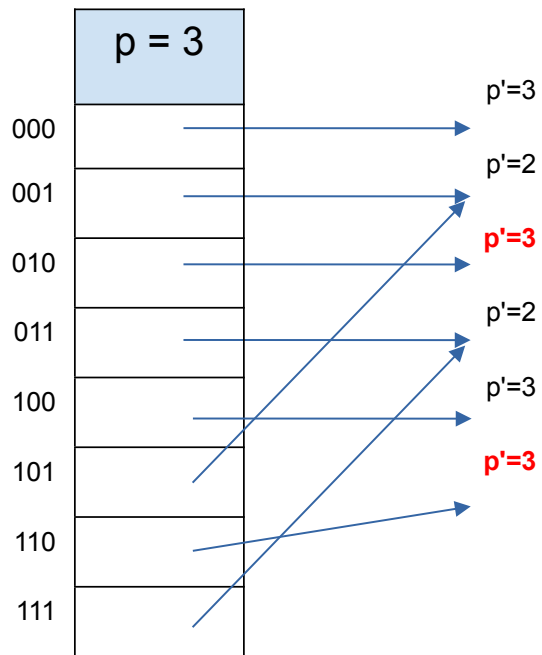
Chave	End.				
24	E1				
9					
6	E3	22	E4	10	E5
7	E6	31	E7		
4	E0	20	E9		

Precisamos organizar as chaves 6, 22, 10 e 26

Adicionar chave 26

Inserindo chaves

Diretório

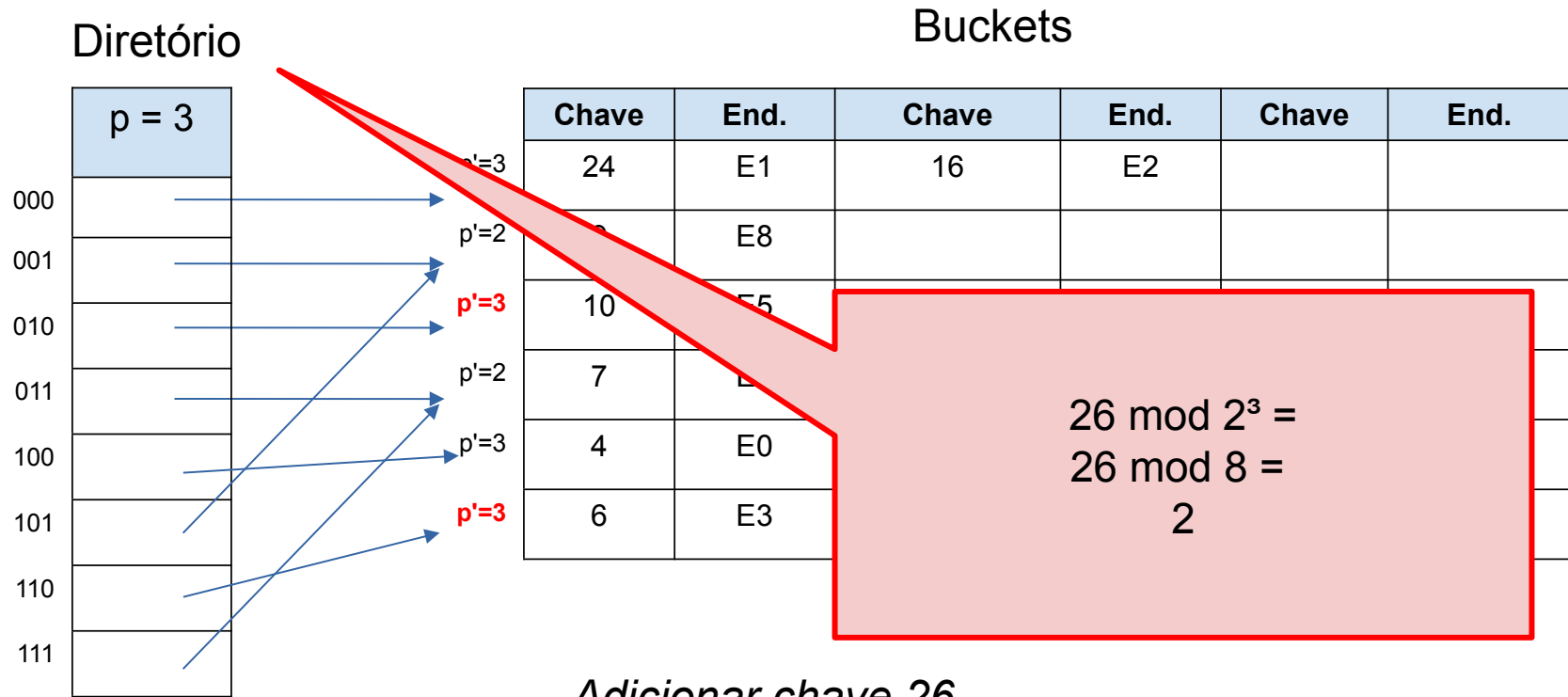


Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
10	E5				
7	E6	31	E7		
4	E0	20	E9		
6	E3	22	E4		

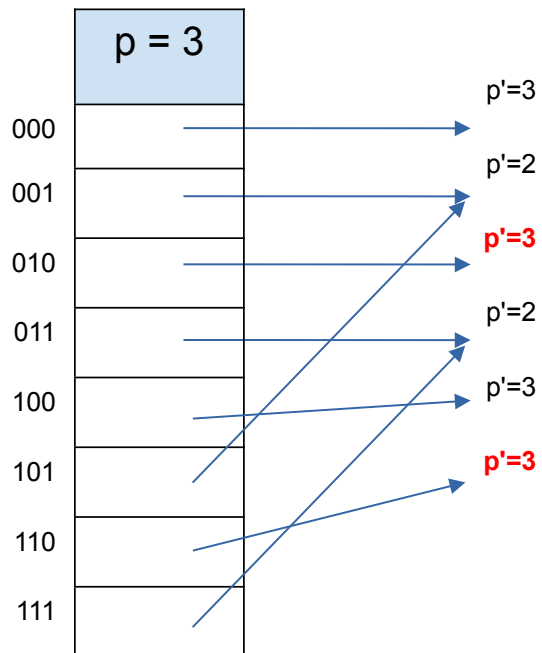
Adicionar chave 26

Inserindo chaves



Inserindo chaves

Diretório



Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
10	E5	26	E10		
7	E6	31	E7		
4	E0	20	E9		
6	E3	22	E4		

Adicionar chave 26

Inserindo chaves

Diretório

	$p = 3$	
000		$p'=3$
001		$p'=2$
010		$p'=3$
011		$p'=2$
100		$p'=3$
101		$p'=3$
110		
111		

Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
10	E5	26	E10		
7	E6	31	E7		
4	E0	20	E9		
6	E3	22	E4		

Deletando



PUC Minas

Deletando chaves

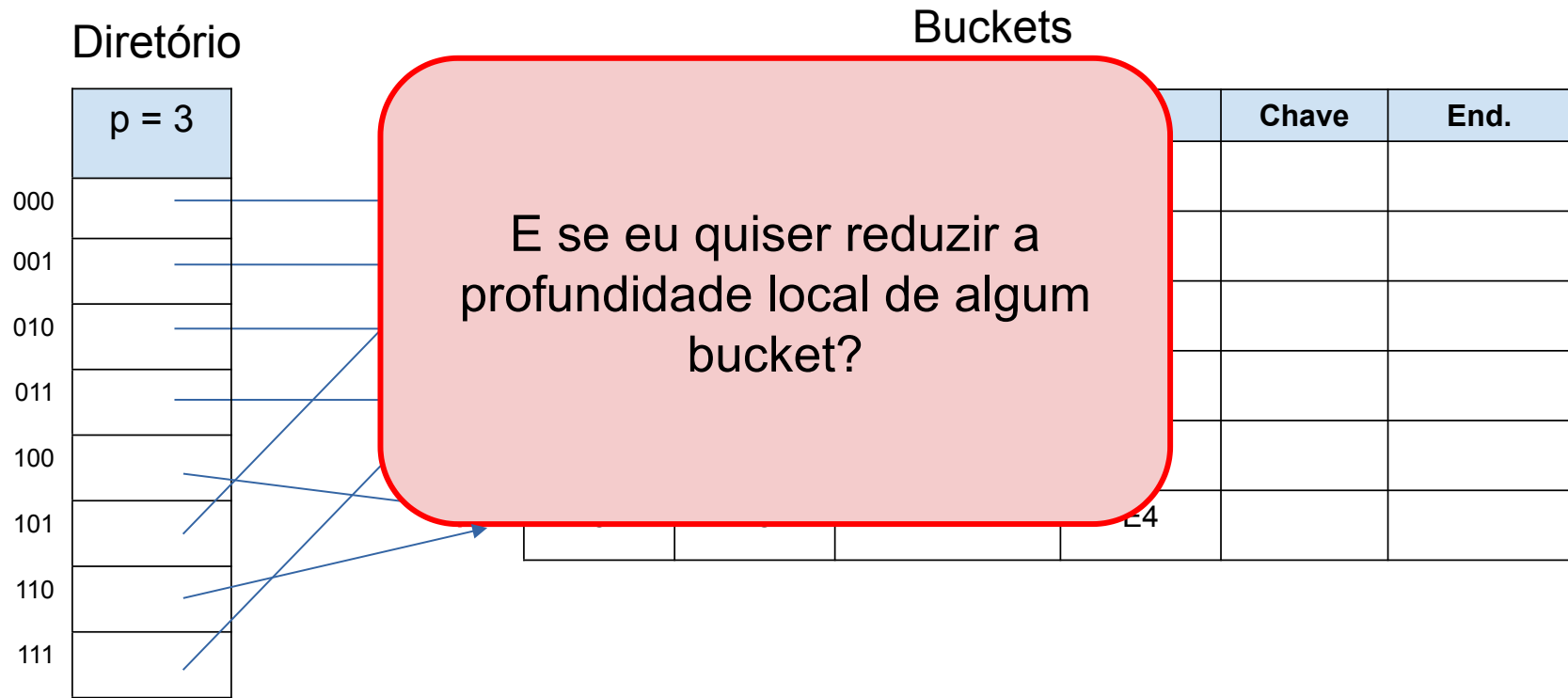
Diretório

	$p = 3$	
000		$p'=3$
001		$p'=2$
010		$p'=3$
011		$p'=2$
100		$p'=3$
101		$p'=3$
110		
111		

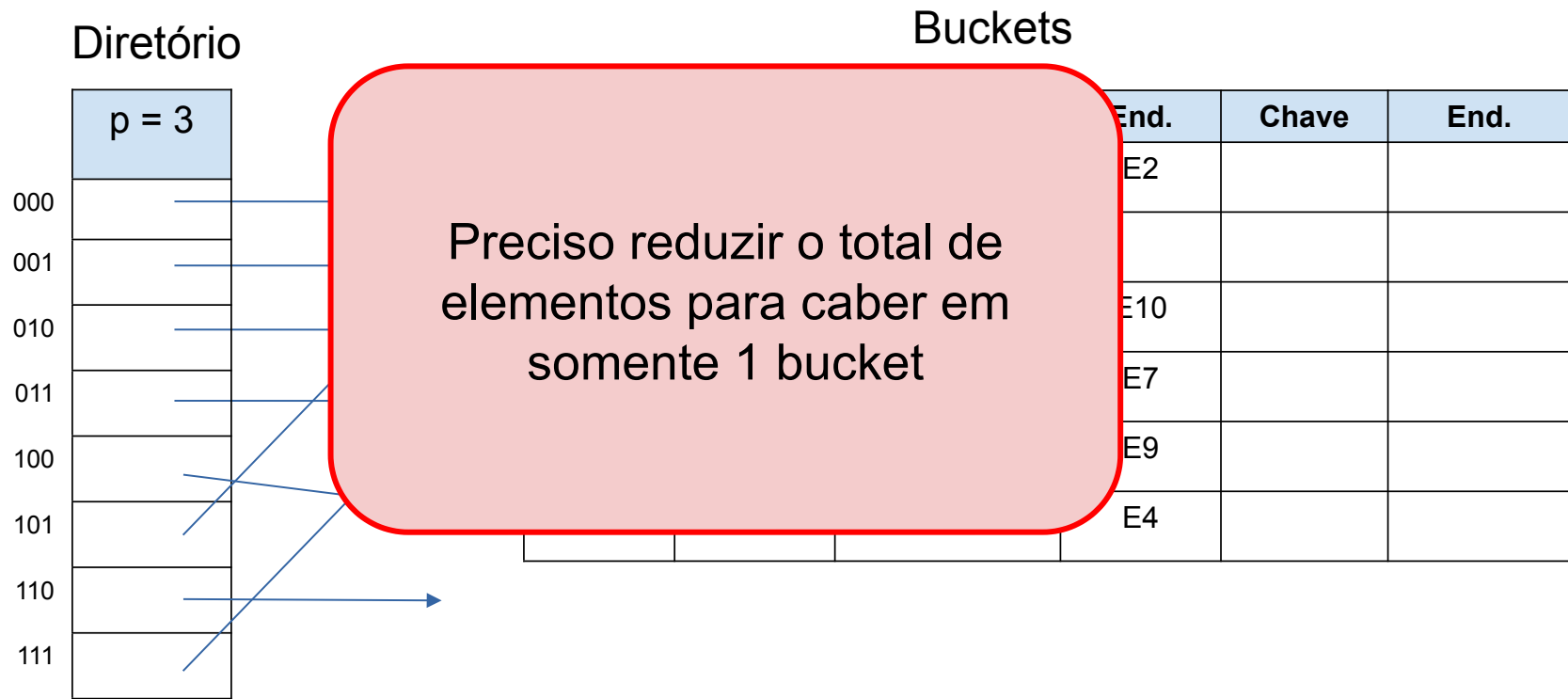
Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
10	E5	26	E10		
7	E6	31	E7		
4	E0	20	E9		
6	E3	22	E4		

Deletando chaves



Deletando chaves



Deletando chaves

Diretório

	$p = 3$
000	
001	
010	
011	
100	
101	
110	
111	

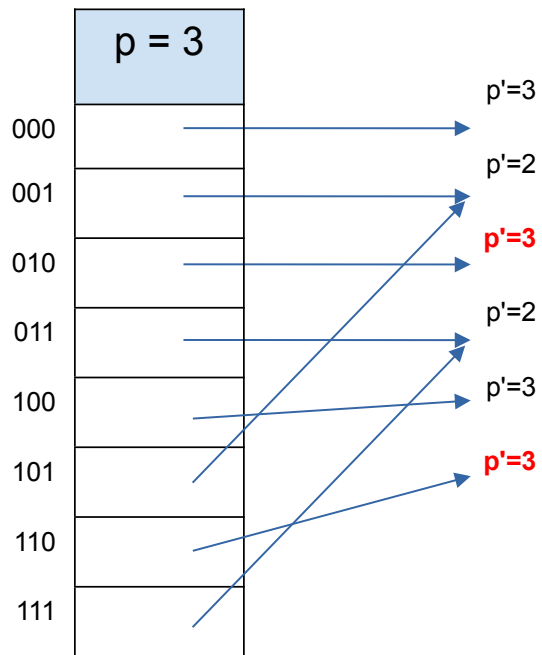
Deletando o 22

	$p'=3$					
4	E0	20	E9			
6	E3	22	E4			

End.	Chave	End.
E2		
E10		
E7		

Deletando chaves

Diretório



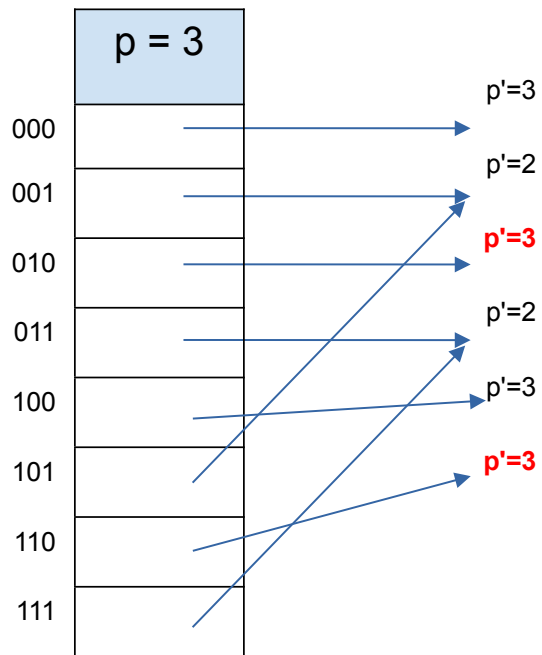
Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
10	E5	26	E10		
7	E6	31	E7		
4	E0	20	E9		
6	E3	22	E4		

Deletando a chave 22

Deletando chaves

Diretório



Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
10	E5	26	E10		
7	E6	31	E7		
4	E0	20	E9		
6	E3				

Deletando chaves

Juntando os buckets 2 e 5

Tenho 3 elementos e o bucket
cabe 3 elementos

Diretório

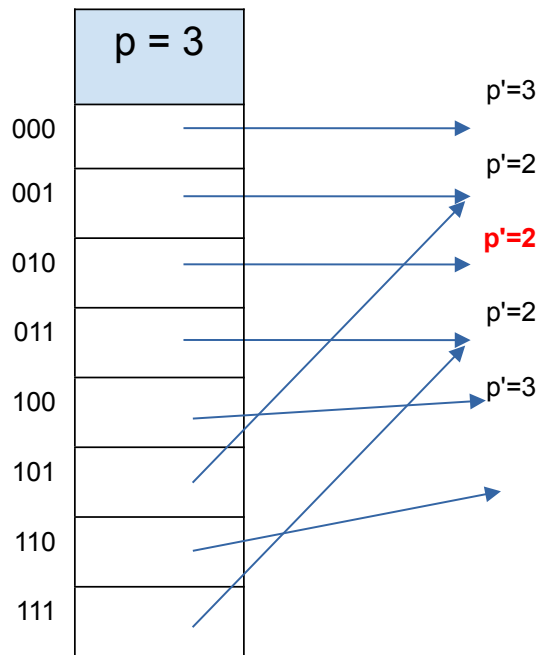
	$p = 3$
000	
001	
010	
011	
100	
101	
110	
111	

Diagram illustrating the deletion of keys from a hash table. The directory (Diretório) shows buckets 000 to 111. Arrows indicate the mapping of keys to buckets. The original bucket 2 (010) is being merged with bucket 5 (101). The new bucket 5 (101) now contains 3 elements (E5, E10, E3).

Chave					
24					
9					
10	E5	26	E10		
7	E6	31	E7		
4	E0	20	E9		
6	E3				

Deletando chaves

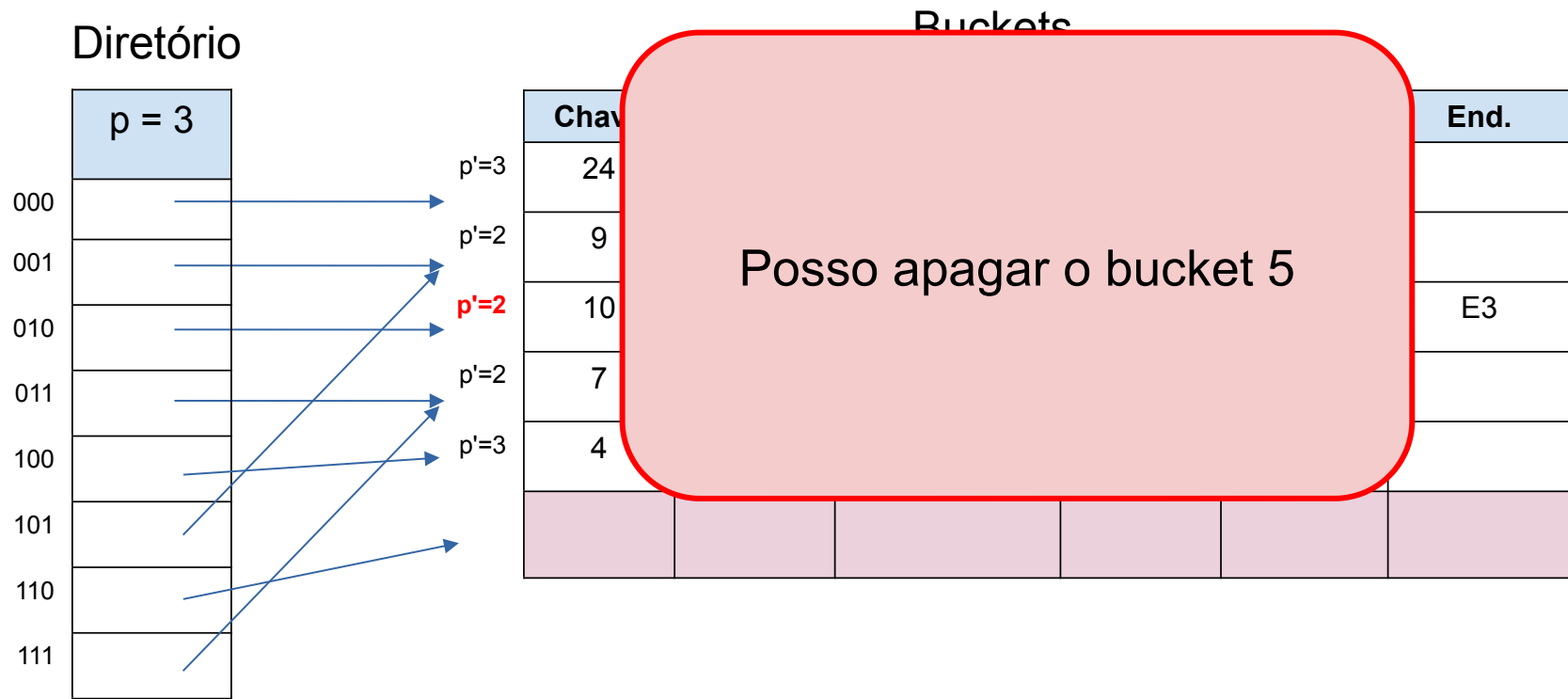
Diretório



Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
10	E5	26	E10	6	E3
7	E6	31	E7		
4	E0	20	E9		

Deletando chaves



Deletando chaves

Diretório

	$p = 3$
000	
001	
010	
011	
100	
101	
110	
111	

Diagram illustrating the mapping from the Directory (Diretório) to the Buckets. The Directory has 8 slots (000 to 111) and a parameter $p = 3$. Arrows indicate the mapping to the Buckets:

- 000 → $p'=3$
- 001 → $p'=2$
- 010 → $p'=2$
- 011 → $p'=2$
- 100 → $p'=3$
- 101 → $p'=2$
- 110 → $p'=2$
- 111 → $p'=3$

Note: The value $p'=2$ is highlighted in red in the original image.

Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
10	E5	26	E10	6	E3
7	E6	31	E7		
4	E0	20	E9		

Deletando chaves

Diretório

	$p = 3$	
000		$p'=3$
001		$p'=2$
010		$p'=2$
011		$p'=2$
100		$p'=3$
101		
110		
111		

Buckets

Chave	End.	Chave	End.	Chave	End.
24	E1	16	E2		
9	E8				
10	E5	26	E10	6	E3
7	E6	31	E7		
4	E0	20	E9		

Conclusões



Hashing – Hash Dinâmico

- O diretório cresce, sem precisarmos reposicionar todos os registros (do índice)
- O índice (lista de buckets) cresce de acordo com a necessidade
- Como não há encadeamento dos buckets, não há perda de eficiência

Hashing – Hash Dinâmico

- Qual a melhor estrutura auxiliar para índices:

Árvore B, B+, B* ou Hash Dinâmico?

Hashing – Hash Dinâmico

- Qual a melhor estrutura auxiliar para índices?

Você só pode acessar elementos por sua chave primária em uma tabela de hash. Isso é mais rápido do que com um algoritmo de árvore (**$O(1)$ em vez de $\log(n)$**), mas você não pode selecionar **intervalos** (tudo entre x e y).

Os algoritmos de árvore suportam isso em $\log(n)$, enquanto os índices de hash podem resultar em uma varredura completa da tabela $O(n)$.

A sobrecarga dos índices de hash geralmente é maior.

Os algoritmos de árvore geralmente são mais fáceis de manter, crescer com dados, escalar, etc.