Lição 10

Tabela *Hash* e Técnicas de *Hashing*

Objetivos

Ao final desta lição, o estudante será capaz de:

- Definir hashing e explicar como o *hashing* funciona
- Implementar técnicas simples de *hashing*
- Discutir como colisões são evitadas/minimizadas através da utilização de técnicas de resolução de colisão
- Explicar os conceitos por trás de arquivos dinâmicos e *hashing*

Técnicas Simples de *Hash*

- **Colisão**

- Uma boa função *hash* executa computação eficiente e produz pouca (ou nenhuma) colisão

Técnicas Simples de *Hash*

- Método de Divisão por Número Primo

Valor Chave k	Valor Hash h(k)
125	8
845	0
444	2
256	9
345	7
745	4
902	5
569	10
254	7
382	5

Valor Chave k	Valor Hash h(k)
234	0
431	2
947	11
981	6
792	12
459	4
725	10
652	2
421	5
458	3

Técnicas Simples de *Hash*

- O valor chave é dividido em duas ou mais partes que são somadas posteriormente, ou utilizar as operações AND ou XOR para obter o endereço *hash*
- Se o endereço resultante tem mais dígitos que o maior endereço no arquivo, os dígitos excedentes de maior ordem serão eliminados
- Desdobramento ao meio, desdobramento em terços, desdobramento por dígitos alternados
- Desdobramento pelos extremos, desdobramento por substituição
- Útil para converter chaves com um grande números de dígitos em um pequeno número de dígitos

Técnicas de Resolução de Colisão

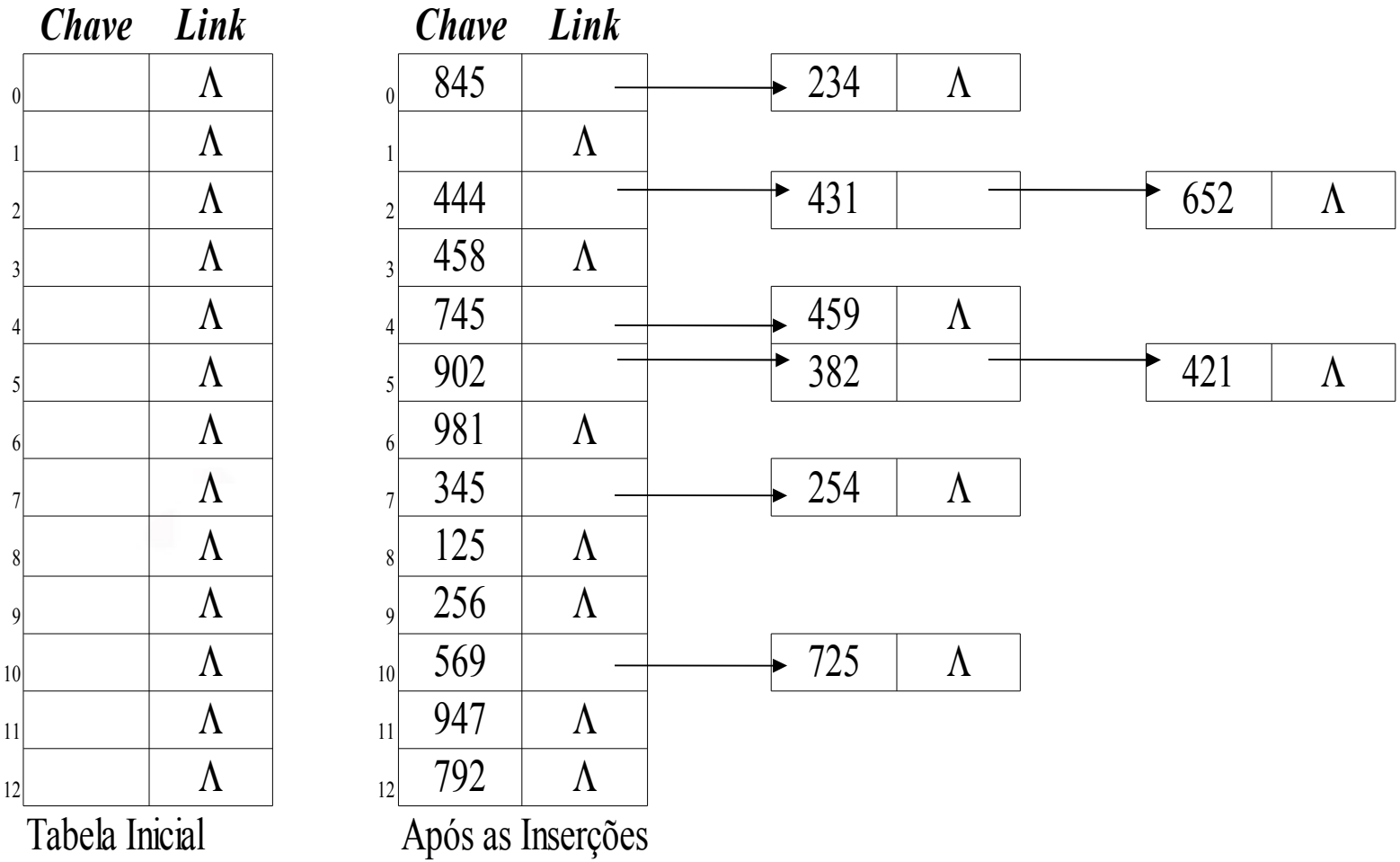
- Para evitar colisão:
  - Desdobramento de registros
  - Usar memória extra
  - Usar *buckets*
- Encadeamento

Técnicas de Resolução de Colisão: Encadeamento

Valor Chave k	Valor Hash h(k)
125	8
845	0
444	2
256	9
345	7
745	4
902	5
569	10
254	7

Valor Chave k	Valor Hash h(k)
234	0
431	2
947	11
981	6
792	12
459	4
725	10
652	2
421	5

Técnicas de Resolução de Colisão: Encadeamento



Técnicas de Resolução de Colisão: Uso de *Buckets*

	CHAVE1	CHAVE2	CHAVE3
0	845	234	
1			
2	444	431	652
3	458		
4	745	459	
5	902	382	421
6	981		
7	345	254	
8	125		
9	256		
10	569	725	
11	947		
12	792		

Técnicas de Resolução de Colisão: Uso de *Buckets*

- Gasta espaço
- Não está livre de estouros de carga adicionais
- Ocorrem problemas quando o número de chaves é maior que o número estático de *slots* em cada endereço

Técnicas de Resolução de Colisão: Espalhamento

- Endereçamento aberto
- Quando não existe mais espaço para inserção no endereço produzido pela função *hash* h(k), um *slot* vazio diferente de h(k) é alocado na tabela *hash*
- Duas técnicas:
  - Busca Linear
  - *Hashing* Duplo

Técnicas de Resolução de Colisão: Espalhamento

- Arquivo explorado sequencialmente como um arquivo circular
- Chave com colisão é armazenada no próximo espaço disponível no endereço
- *Slots* na tabela *hash* podem conter apenas uma chave ou podem ser um *bucket*

Valor Chave k	Valor Hash h(k)
125	8
845	0
444	2
256	9
345	7
745	4
902	5
569	10
254	7
382	5

Valor Chave k	Valor Hash h(k)
234	0
431	2
947	11
981	6
792	12
459	4
725	10
652	2
421	5
458	3

	Chave1	Chave2
0	845	234
1		
2	444	431
3	**652**	458
4	745	459
5	902	382
6	981	**421**
7	345	254
8	125	
9	256	
10	569	725
11	947	
12	792	

Técnicas de Resolução de Colisão: Endereçamento Aberto

- Faz uso de uma segunda função de *hash*, como $h_2(k)$, toda vez que ocorre uma colisão
- Use a função primária de *hash* $h_1(k)$ para determinar a posição i que irá posicionar o valor
- Se ocorrer uma colisão, use a função de *rehash* $r_h(i, k)$ sucessivamente até que um espaço vazio seja encontrado:

$$r_h(i, k) = (i + h_2(k)) \bmod m$$

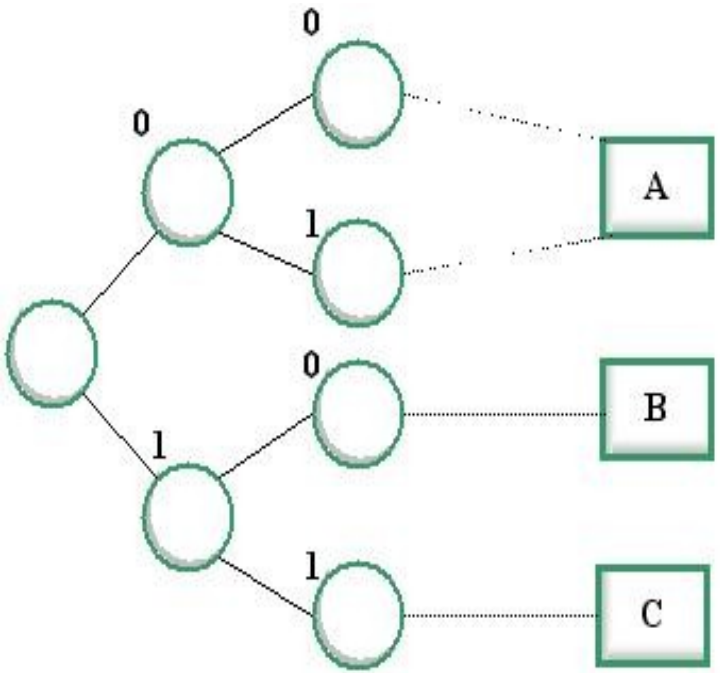
Técnicas de Resolução de Colisão: Endereçamento Aberto

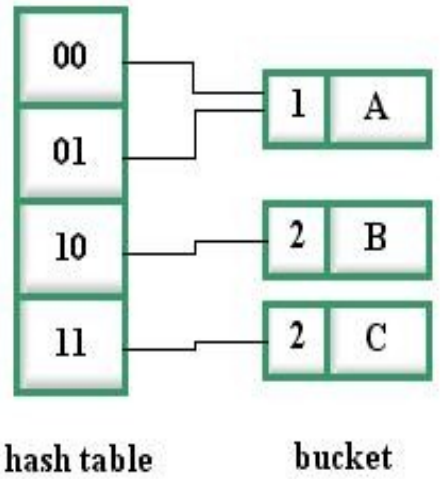
Valor Chave k	Valor Hash h(k)
125	8
845	0
444	2
256	9
345	7
745	4
902	5
569	10
254	7
382	5

Valor Chave k	Valor Hash h(k)
234	0
431	2
947	11
981	6
792	12
459	4
725	10
652	2
421	5
458	3

Arquivos Dinâmicos & *Hashing*: *Hashing* Extensível

- Ajuste próprio de estrutura com um tamanho do *bucket* ilimitado
- Árvore: constrói um índice baseado em uma representação numérica binária do valor do hash



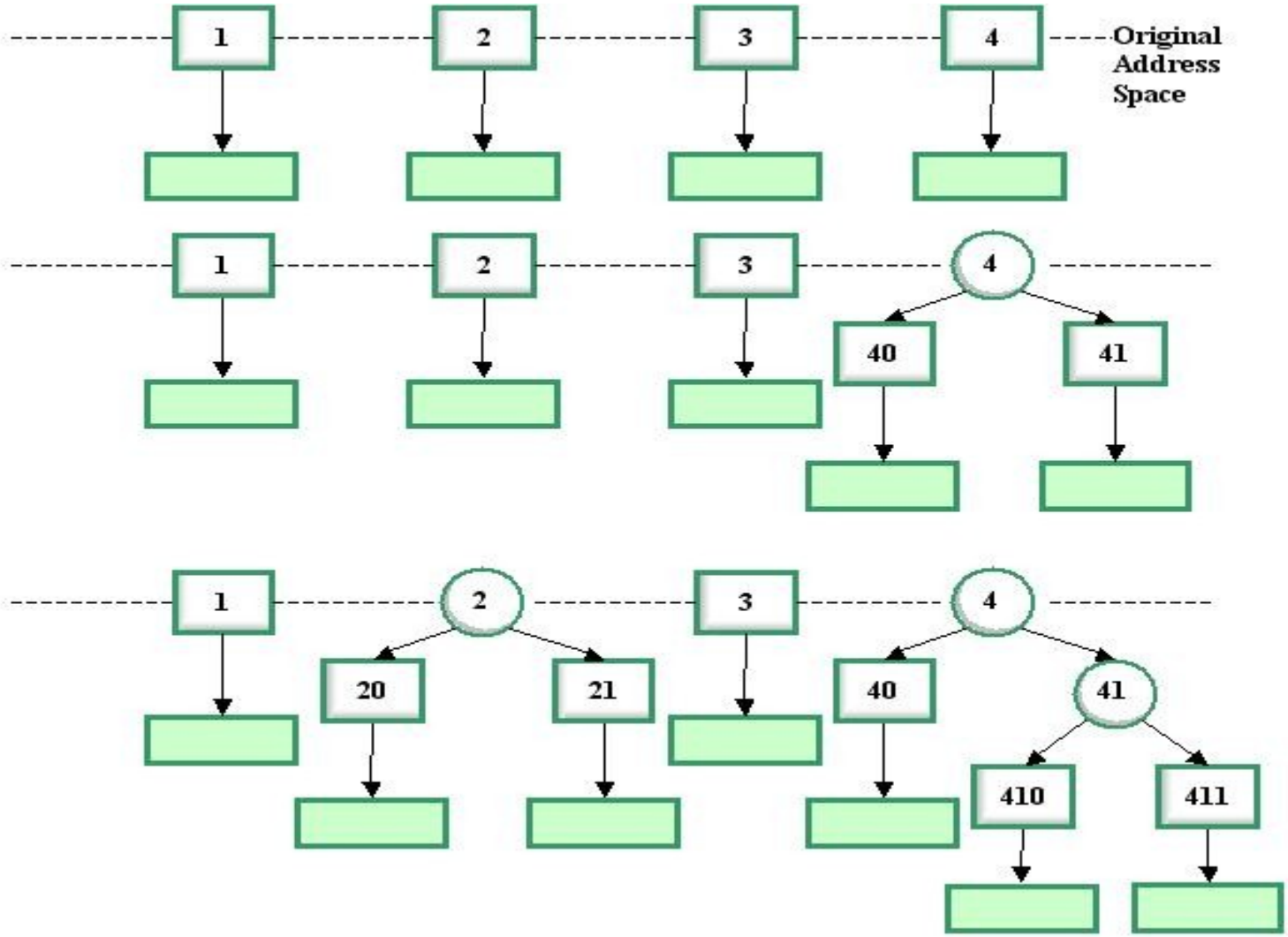


Bucket	Endereço
A	
B	10
C	11

Arquivos Dinâmicos & *Hashing*: *Hashing* Extensível

- Bucket tem uma estrutura
- Tabela Hash é duplicada de tamanho toda vez que é expandida
- *Buckets* vizinhos
- Inutilização dos *buddy buckets*

Arquivos Dinâmicos & *Hashing*: *Hashing* Extensível



Sumário

- Técnicas Simples de *Hash*
- Técnicas de Resolução de Colisão
  - Encadeamento
  - Uso de *Buckets*
  - *Hashing*
  - Endereçamento Aberto
- Arquivos Dinâmicos & *Hashing*
  - *Hashing* Extensível

Parceiros

- **Os seguintes parceiros tornaram JEDI™ possível em Língua Portuguesa:**