MATA54 - Estruturas de Dados e Algoritmos II

Hashing Extensível

Flávio Assis
Versão gerada a partir de slides do Prof. George Lima

IC - Instituto de Computação

Salvador, agosto de 2021

Motivação

E se em tabelas hashing de tamanho m fixo $\alpha \to 1$?

- Cadeias de sondagem tendem a ficar maiores $\to$ desempenho cai
- Necessidade de construir nova tabela hashing com maior valor de m

E se o valor de m for muito maior que n?

- Uso desnecessário de espaço
- Conveniente diminuir o tamanho de m

Abordagem:

Alterar dinamicamente a tabela hashing para se adaptar ao número de chaves: o espaço de espalhamento cresce ou diminui em função do número de chaves

Dois Enfoques de Hashing Dinâmico

- **Hashing extensível** [Fagin, R; Nievergelt, J.; Pippenger, N; Strong, H. R. “Extendible Hashing-A Fast Access Method for Dynamic Files”. ACM TODS, 4(3):315-344, 1979.]

- **Hashing linear** [Litwin, W. ”Linear hashing: A new tool for file and table addressing”. 6th Conference on Very Large Databases. pp 212–223, 1980.]

Hashing Extensível

Ideia básica

Armazenamento de registros/chaves em **páginas** endereçadas a partir de um **índice** (função de hashing aplicada às chaves)

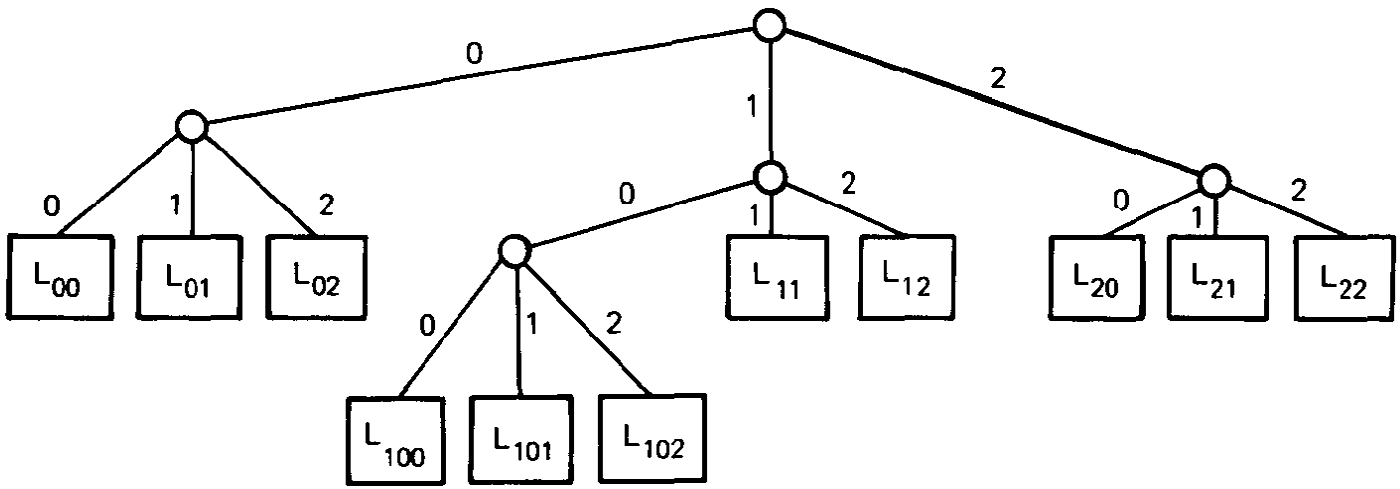


Fig. 1. A radix search tree

Retirado de [Fagin *et al.*, 79]

Ideia básica

Armazenamento de registros/chaves em **páginas** endereçadas a partir de um **índice** (função de hashing aplicada às chaves)

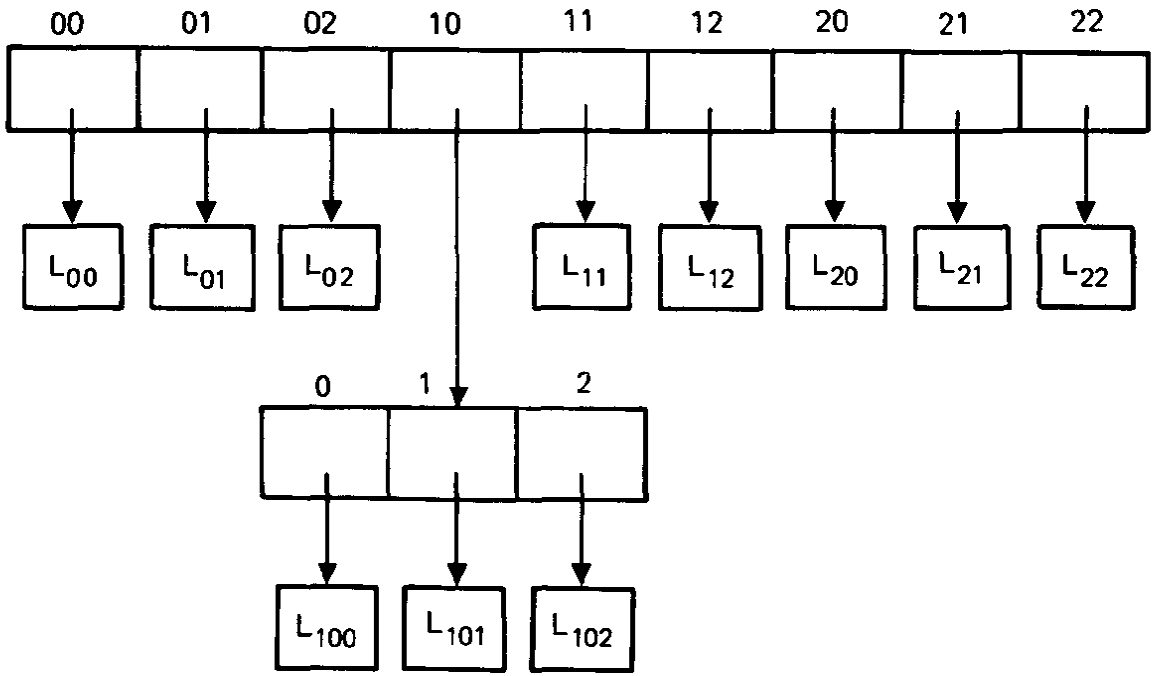


Fig. 2. Radix search tree with two levels compressed into one

Retirado de [Fagin *et al.*, 79]

Ideia básica

Armazenamento de registros/chaves em **páginas** endereçadas a partir de um **índice** (função de hashing aplicada às chaves)

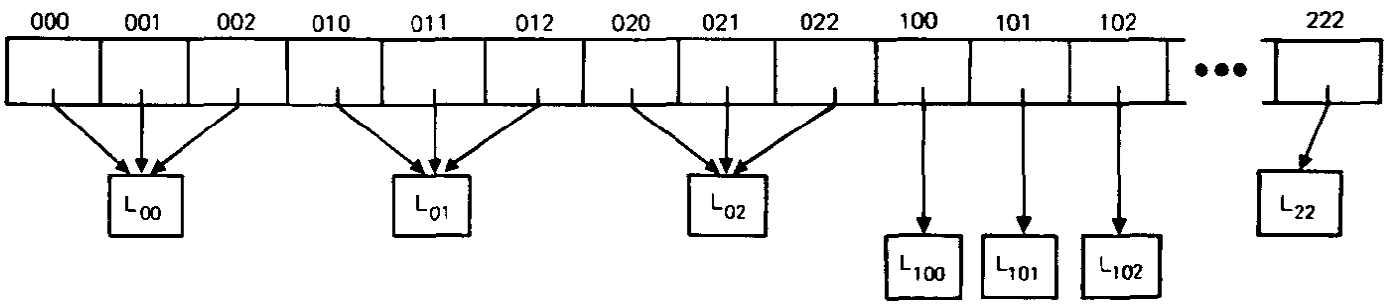
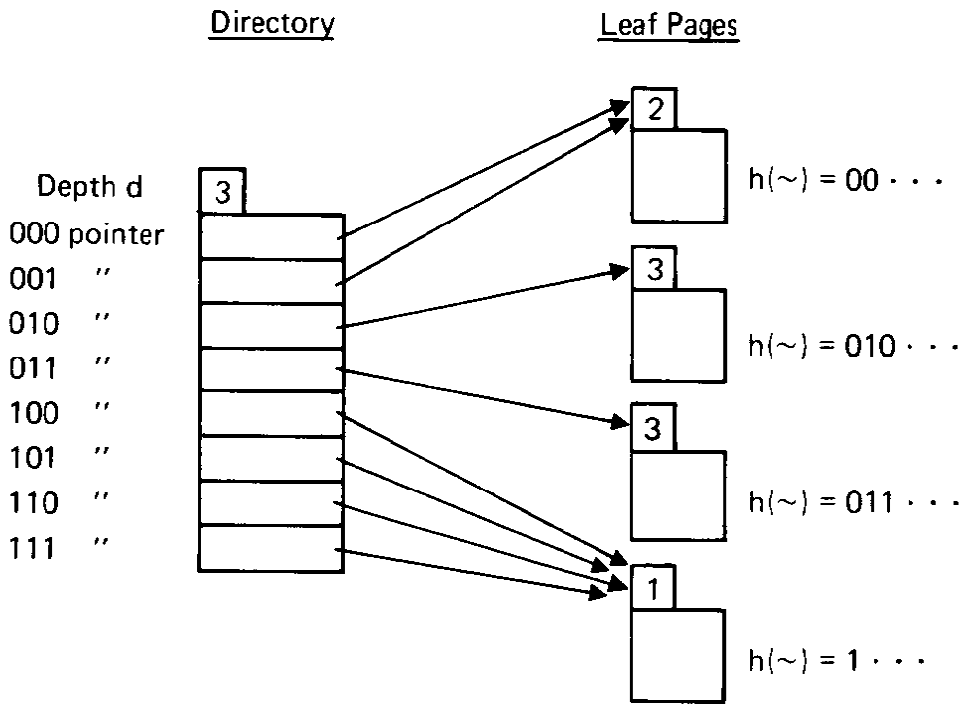


Fig. 3. Degenerate radix search tree

Retirado de [Fagin *et al.*, 79]

Ideia básica

Armazenamento de registros/chaves em **páginas** endereçadas a partir de um **índice** (função de hashing aplicada às chaves)



Retirado de [Fagin *et al.*, 79]

Estruturação

Página de dados (Bucket)

- Registros contidos na página
- Indicador d_i de profundidade da página P_i
  - Indica que os d_i bits mais significativos para todos os registros em P_i são iguais

Diretório

- Ponteiros para páginas de dados
- Indicador d de profundidade do diretório
  - Máximo dos indicadores de profundidade das páginas
- 2^d valores possíveis para $h(k)$
  - Não precisam ser explicitamente armazenados

Ilustração: Inserção

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 11$.

O valor da função hashing é considerado em formato binário, da esquerda para a direita.

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

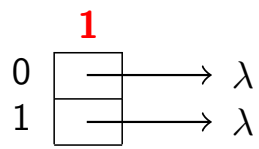


Ilustração: Inserção

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 11$.

O valor da função hashing é considerado em formato binário, da esquerda para a direita.

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

Inserção do **27**:

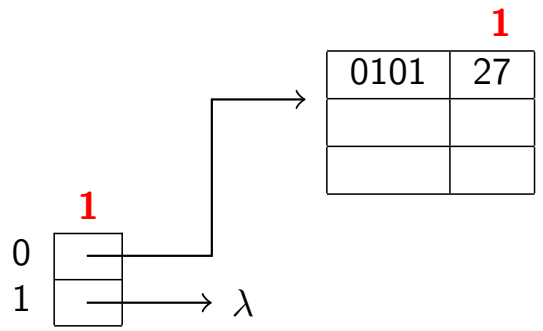


Ilustração: Inserção

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 11$.

O valor da função hashing é considerado em formato binário, da esquerda para a direita.

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

Inserção do **19**:

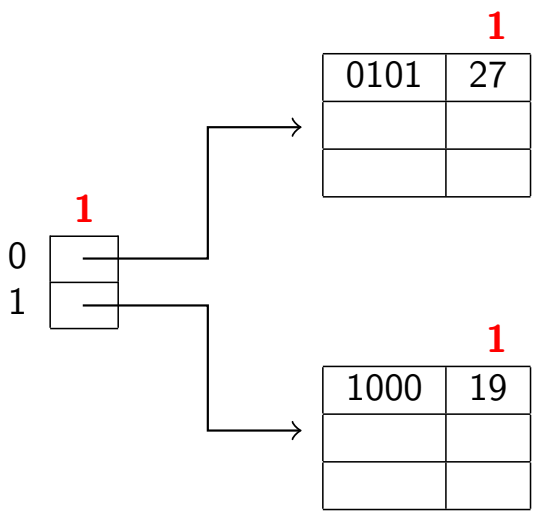


Ilustração: Inserção

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 11$.

O valor da função hashing é considerado em formato binário, da esquerda para a direita.

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

Inserção do **18**:

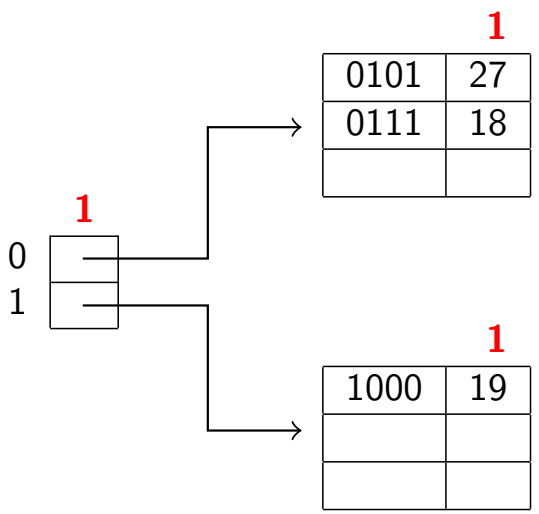


Ilustração: Inserção

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 11$.

O valor da função hashing é considerado em formato binário, da esquerda para a direita.

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

Inserção do **25**:

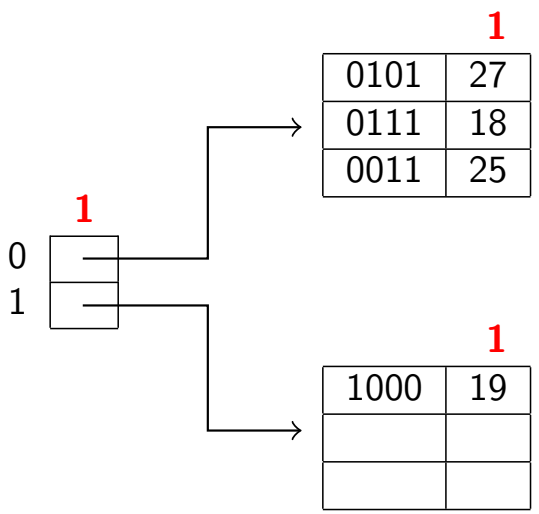


Ilustração: Inserção

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 11$.

O valor da função hashing é considerado em formato binário, da esquerda para a direita.

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

Inserção do **28**:

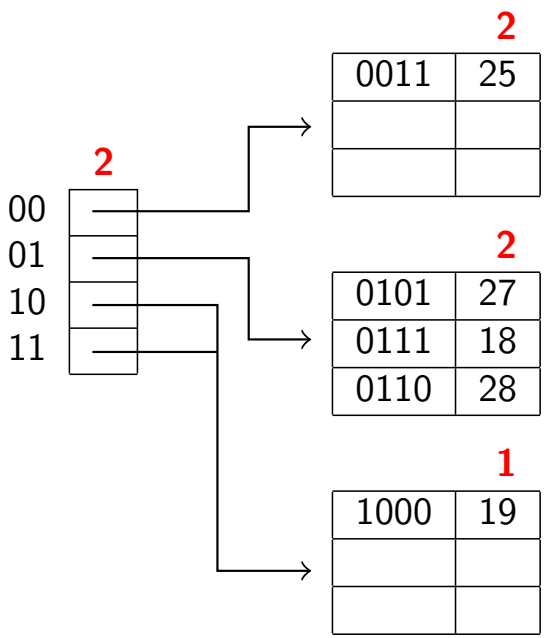


Ilustração: Inserção

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 11$.

O valor da função hashing é considerado em formato binário, da esquerda para a direita.

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

Inserção do **42**:

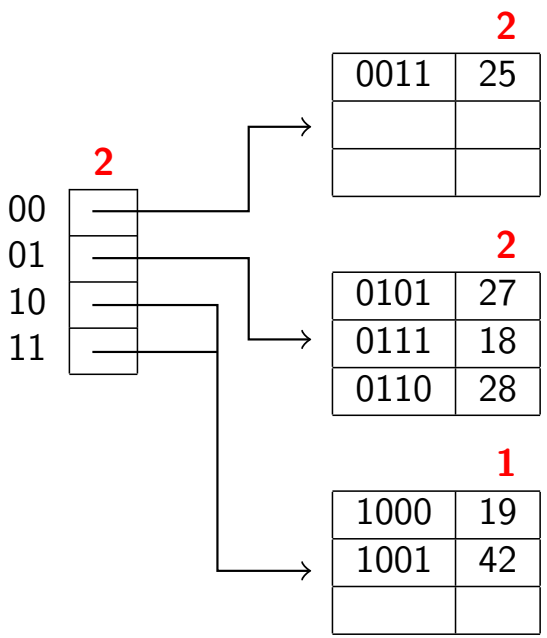


Ilustração: Inserção

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 11$.

O valor da função hashing é considerado em formato binário, da esquerda para a direita.

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

Inserção do **43**:

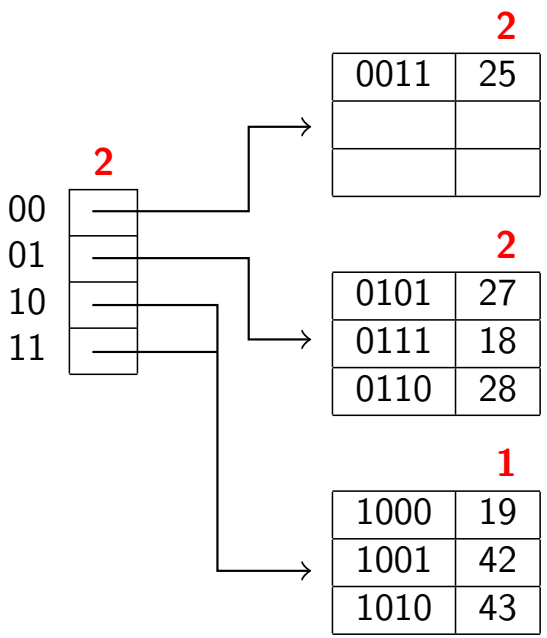
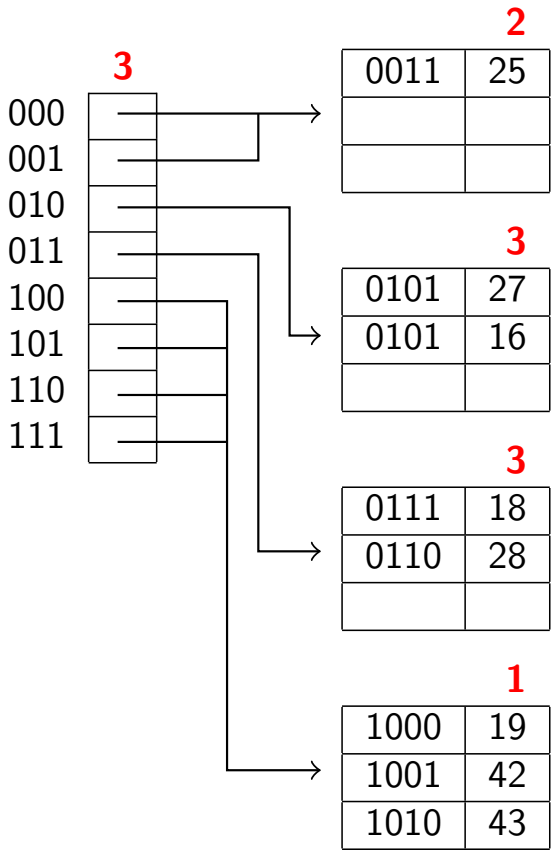


Ilustração: Inserção

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

Inserção do **16**:



Inserção de um registro

Inserção de um registro

1. Executar início do procedimento de busca, localizando o ponteiro p_i correspondente ao registro R com chave k a ser inserido

Inserção de um registro

1. Executar início do procedimento de busca, localizando o ponteiro p_i correspondente ao registro R com chave k a ser inserido

2. Ir para a página P_i indicada por p_i

Inserção de um registro

1. Executar início do procedimento de busca, localizando o ponteiro p_i correspondente ao registro R com chave k a ser inserido
2. Ir para a página P_i indicada por p_i
3. Se há espaço para R ser inserido em P_i, executar a inserção de R

Inserção de um registro

1. Executar início do procedimento de busca, localizando o ponteiro p_i correspondente ao registro R com chave k a ser inserido

2. Ir para a página P_i indicada por p_i

3. Se há espaço para R ser inserido em P_i, executar a inserção de R

4. Caso contrário, considerar os indicadores de profundidade d_i de P_i e d do diretório:
   - 4.1 Alocar nova página P_j (com endereço p_j)
   - 4.2 Transferir os registros de P_i para nova página temporária Q
   - 4.3 Atualizar os indicadores de profundidades de P_i e P_j para $d_i + 1$
   - 4.4 Se $d_i > d$, fazer $d = d_i$ e duplicar o tamanho do diretório, atualizando os ponteiros.
   - 4.5 Inserir todos os registros contidos em Q e o novo registro R

Busca de um registro

Busca de um registro

1. Calcular o valor de $k' = h(k)$ e ler o indicador de profundidade d do diretório

Busca de um registro

1. Calcular o valor de $k' = h(k)$ e ler o indicador de profundidade d do diretório

2. Seja r os primeiros d bits de k'

Busca de um registro

1. Calcular o valor de $k' = h(k)$ e ler o indicador de profundidade d do diretório

2. Seja r os primeiros d bits de k'

3. Ir para o ponteiro p indicado por r
   - A posição do ponteiro é $b + r \times t$, com: t tamanho do ponteiro; b endereço de início do diretório.

Busca de um registro

1. Calcular o valor de $k' = h(k)$ e ler o indicador de profundidade d do diretório

2. Seja r os primeiros d bits de k'

3. Ir para o ponteiro p indicado por r
   - A posição do ponteiro é $b + r \times t$, com: t tamanho do ponteiro; b endereço de início do diretório.

4. Ir para a página indicada por p

Ilustração: Remoção

Ao se remover um registro de uma página, procura-se a sua **página irmã**. Se os registros nessas duas páginas puderem ser armazenados em apenas uma página, faz-se a **junção** dessas duas páginas.

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

Ex.: Remoção do **18** - as páginas apontadas por **010** e **011** são irmãs:

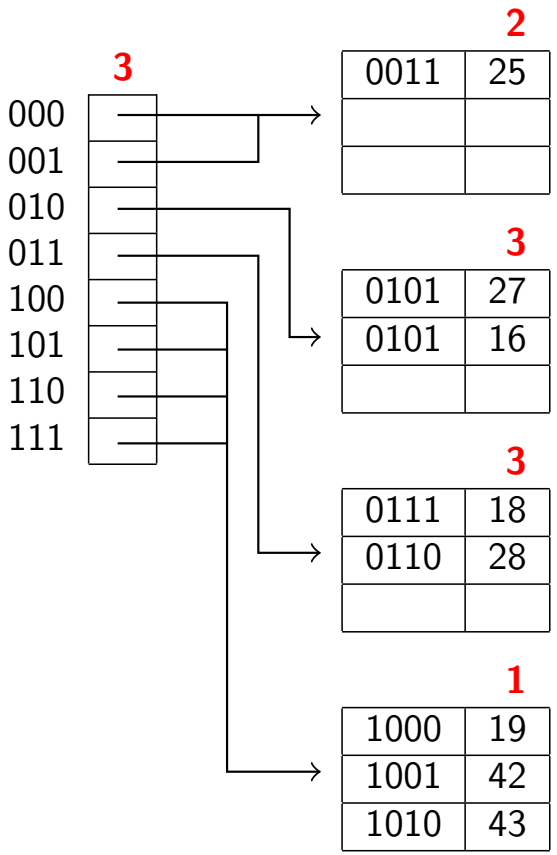


Ilustração: Remoção

Ao se remover um registro de uma página, procura-se a sua **página irmã**. Se os registros nessas duas páginas puderem ser armazenados em apenas uma página, faz-se a **junção** dessas duas páginas.

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$

Ex.: Remoção do **18** - as páginas do meio são irmãs:

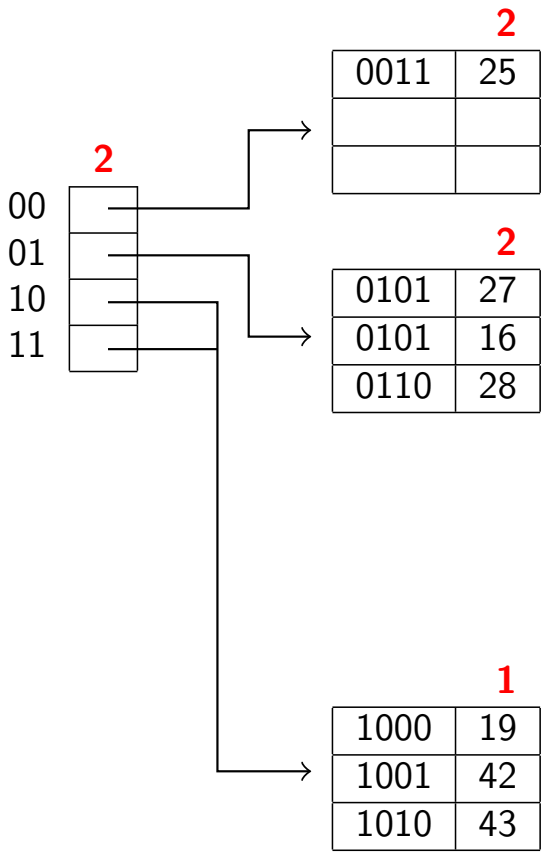


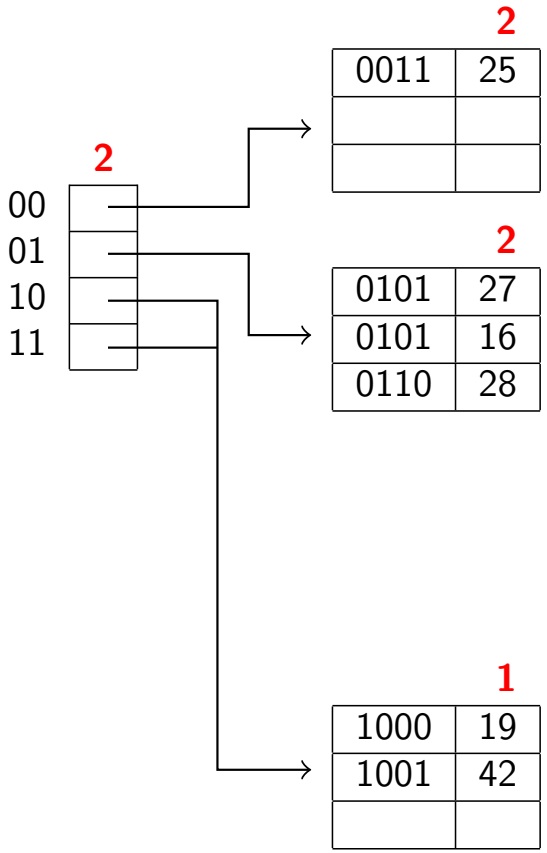
Ilustração: Remoção

Se os registros nas páginas irmãs não puderem ser unidos em apenas uma página, simplesmente remove-se o registro.

Ex.: Remoção do **43**:

Valores de $h(k)$:

$h(27) = 5 = 0101$
$h(19) = 8 = 1000$
$h(18) = 7 = 0111$
$h(25) = 3 = 0011$
$h(28) = 6 = 0110$
$h(42) = 9 = 1001$
$h(43) = 10 = 1010$
$h(16) = 5 = 0101$



Remoção de um registro

Remoção de um registro

1. Buscar registro R com chave k obtendo a página P_i. Se R não for encontrado, retornar.

Remoção de um registro

1. Buscar registro R com chave k obtendo a página P_i. Se R não for encontrado, retornar.

2. Remover R de P_i

Remoção de um registro

1. Buscar registro R com chave k obtendo a página P_i. Se R não for encontrado, retornar.

2. Remover R de P_i

3. Seja P_j página "irmã" de P_i e d o indicador de profundidade do diretório. Se ambas puderem ser unidas:
   - 3.1 Transferir os registros de P_i e P_j para uma área temporária Q
   - 3.2 Remover P_j (ou P_i) a atualizar p_j ($p_j \leftarrow p_i$) no diretório
   - 3.3 Decrementar d_i de uma unidade
   - 3.4 Inserir todos os registros em Q, liberando Q em seguida

Remoção de um registro

1. Buscar registro R com chave k obtendo a página P_i. Se R não for encontrado, retornar.

2. Remover R de P_i

3. Seja P_j página "irmã" de P_i e d o indicador de profundidade do diretório. Se ambas puderem ser unidas:
   - 3.1 Transferir os registros de P_i e P_j para uma área temporária Q
   - 3.2 Remover P_j (ou P_i) a atualizar p_j ($p_j \leftarrow p_i$) no diretório
   - 3.3 Decrementar d_i de uma unidade
   - 3.4 Inserir todos os registros em Q, liberando Q em seguida

4. Enquanto todos os ponteiros "irmãos" no diretório forem iguais e $d > 1$
   - 4.1 Reduzir o indicador de profundidade d de uma unidade
   - 4.2 Reduzir pela metade o tamanho do diretório, atualizando os ponteiros.

Observações

- Diretório construído de forma eficiente (geralmente armazenado em memória principal)
- Pode-se determinar um limite para a maior profundidade da tabela de índice. Neste caso pode-se encadear páginas, quando não for mais possível dividir páginas.
- Enquanto não houver encadeamento de páginas e o diretório couber na memória principal, é necessário apenas um acesso a disco para encontrar qualquer registro - **O(1)**
- Em cada página, registros podem estar armazenados usando métodos hashing tradicionais (de tamanho estático)

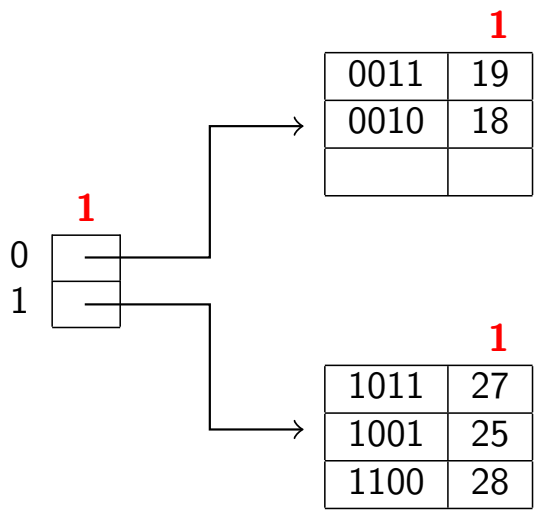
Exercício

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 16$.

Valores de $h(k)$:

$h(27) = 11 = 1011$
$h(19) = 3 = 0011$
$h(18) = 2 = 0010$
$h(25) = 9 = 1001$
$h(28) = 12 = 1100$
$h(42) = 10 = 1010$
$h(43) = 11 = 1011$
$h(16) = 0 = 0000$

Inserção de **27**, **19**, **18**, **25** e **28**:



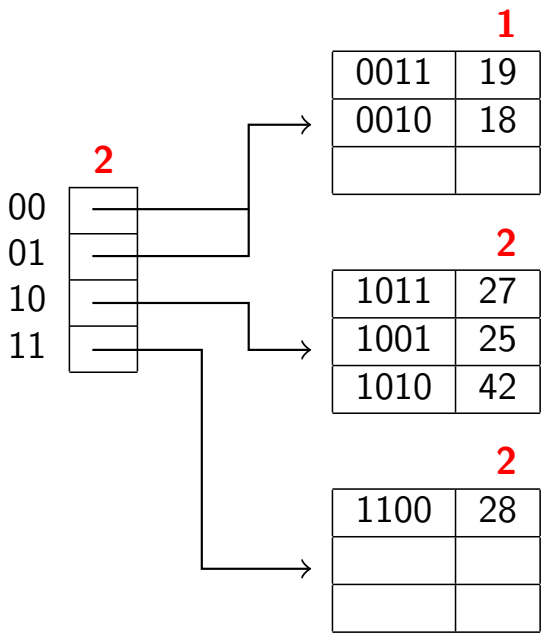
Exercício

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 16$.

Valores de $h(k)$:

$h(27) = 11 = 1011$
$h(19) = 3 = 0011$
$h(18) = 2 = 0010$
$h(25) = 9 = 1001$
$h(28) = 12 = 1100$
$h(42) = 10 = 1010$
$h(43) = 11 = 1011$
$h(16) = 0 = 0000$

Inserção do **42**:



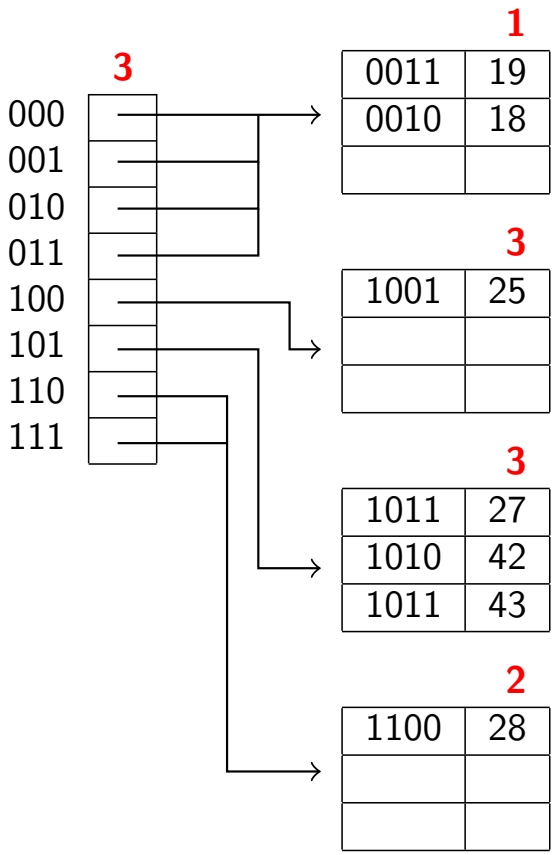
Exercício

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 16$.

Valores de $h(k)$:

$h(27) = 11 = 1011$
$h(19) = 3 = 0011$
$h(18) = 2 = 0010$
$h(25) = 9 = 1001$
$h(28) = 12 = 1100$
$h(42) = 10 = 1010$
$h(43) = 11 = 1011$
$h(16) = 0 = 0000$

Inserção de **43**:



Exercício

Incluir registros com chaves $27, 19, 18, 25, 28, 42, 43$ e 16 usando hashing extensível, com páginas de capacidade 3 e $h(k) = k \bmod 16$.

Valores de $h(k)$:

$h(27) = 11 = 1011$
$h(19) = 3 = 0011$
$h(18) = 2 = 0010$
$h(25) = 9 = 1001$
$h(28) = 12 = 1100$
$h(42) = 10 = 1010$
$h(43) = 11 = 1011$
$h(16) = 0 = 0000$

Inserção de **16**: