

Árvores-B: Definição

Profa. Dra. Cristina Dutra de Aguiar Ciferri

Árvore-B

- Método genérico para o armazenamento e a recuperação de dados
 - voltado para arquivos volumosos
 - proporciona rápido acesso aos dados
 - possui custo mínimo de *overhead*
 - Referência
 - Bayer, R.; McCreight, E. *Organization and Maintenance of Large Ordered Indexes*.
 - Boing Corporation, 1972.
-

Sistemas de Banco de Dados

- 1979
 - árvore-B é, de fato, a organização padrão para indexação
 - Atualmente
 - amplamente utilizada
 - existem variantes
 - árvore-B⁺
 - árvore-B^{*}
 - árvore-B paralela
-

Propósito Geral

- Índice
 - extremamente volumoso
 - *Buffer-pool* pequeno
 - apenas uma parcela do índice pode ser carregada em memória principal
 - operações baseadas em disco
-

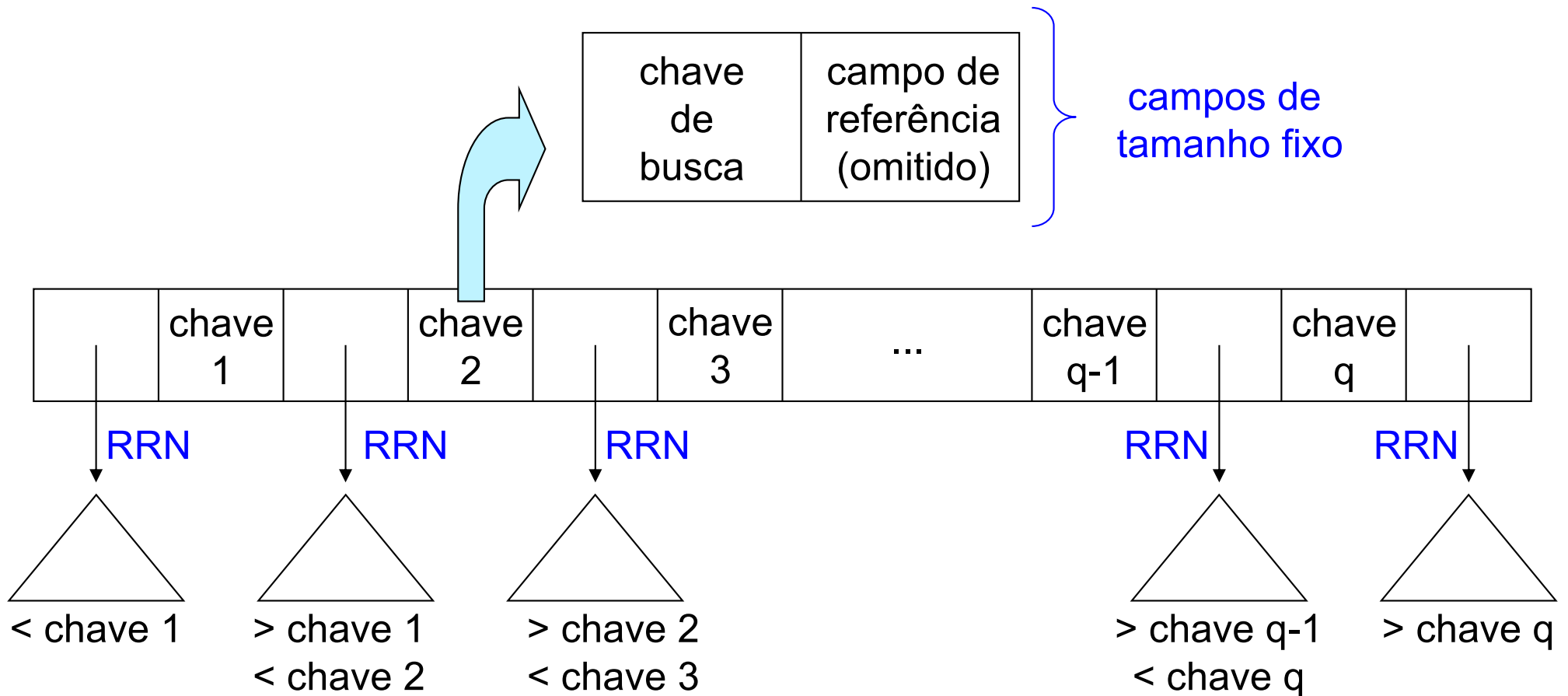
Características

- Balanceada
 - *Bottom-up* para a criação (em disco)
 - nós folhas → nó raiz
 - as chaves na raiz da árvore emergem naturalmente
 - Nó (= página de disco)
 - sequência ordenada de chaves
 - conjunto de ponteiros
 - número de ponteiros = número de chaves + 1
-

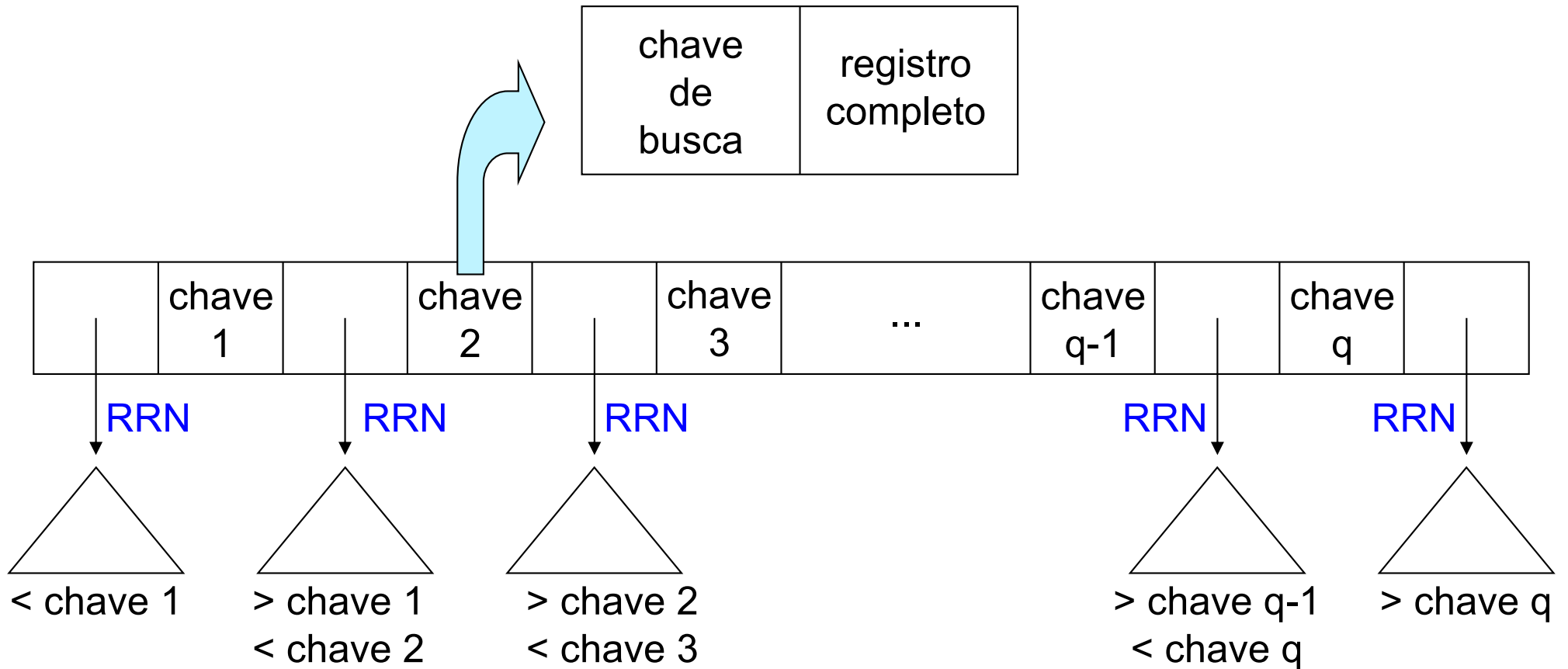
Características

- Ordem (Knuth 1973)
 - número máximo de ponteiros que pode ser armazenado em um nó
 - exemplo: árvore-B de ordem 8
 - máximo de 7 chaves e 8 ponteiros
 - Observações
 - número máximo de ponteiros é igual ao número máximo de descendentes de um nó
 - nós folhas (nível mais baixo das chaves) não possuem filhos, e seus ponteiros são nulos
-

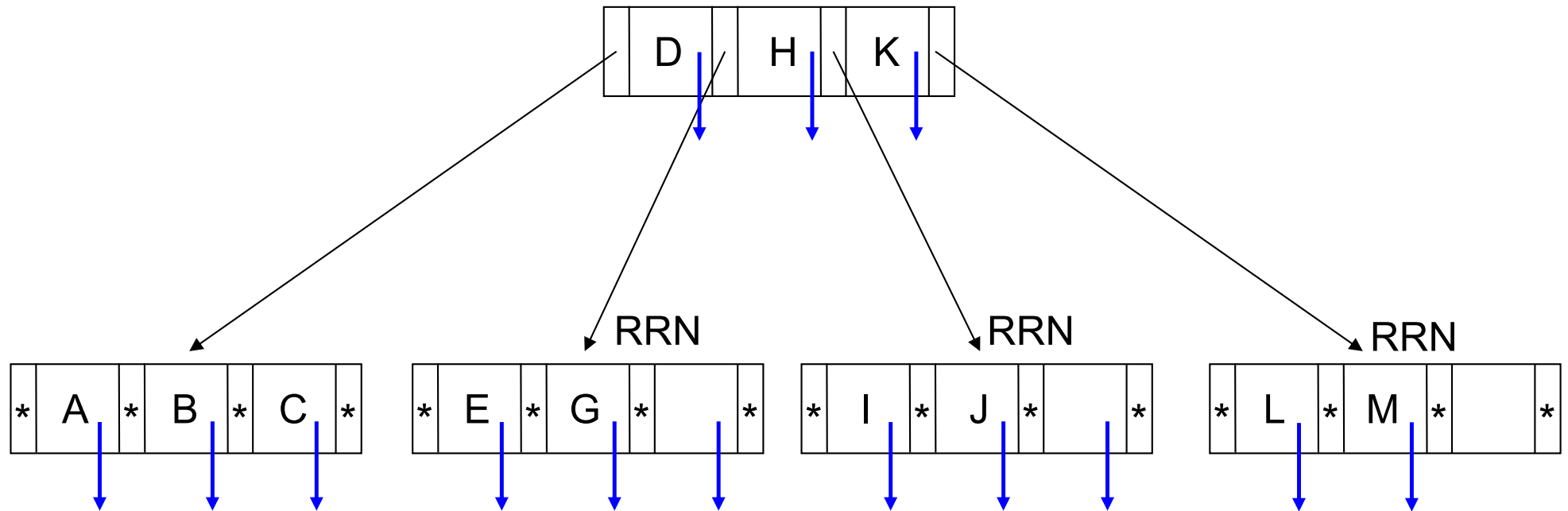
Estrutura Lógica de um Nó



Estrutura Lógica de um Nó



Exemplo



—————> RRN dos nós da árvore-B

—————> RRN ou *byte offset* dos registros do arquivo de dados

Árvore-B⁺

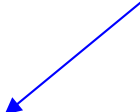
Profa. Dra. Cristina Dutra de Aguiar Ciferri

Acesso Sequencial Indexado

- Alternativas
 - acesso indexado
 - o arquivo pode ser visto como um conjunto de registros que são indexados por uma chave
 - acesso sequencial
 - o arquivo pode ser acessado sequencialmente (i.e., registros fisicamente contínuos)
 - Ideia
 - arquivos devem permitir acesso indexado eficiente, e também acesso sequencial
-

Organização dos Registros

- Problema
 - manter os registros ordenados fisicamente pela chave (*sequence set*)
- Solução
 - organizar registros em blocos



um bloco consiste na unidade básica de entrada e saída e deve ter seu tamanho determinado pelo tamanho do *buffer-pool*

Uso de Blocos

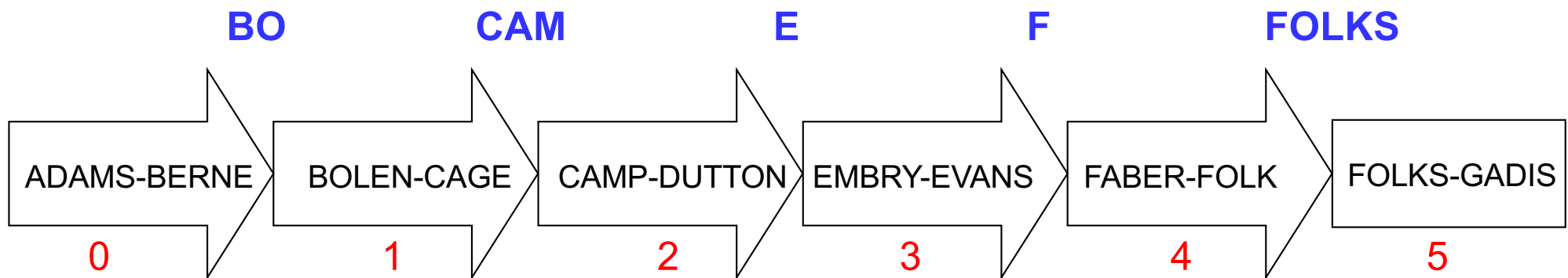
- Características
 - o conteúdo de cada bloco está ordenado, e pode ser recuperado em um acesso
 - cada bloco mantém um ‘ponteiro’ para o bloco antecessor e um ‘ponteiro’ para o bloco sucessor
 - blocos logicamente adjacentes não estão (necessariamente) fisicamente adjacentes
 - Garante acesso sequencial ao arquivo
-

Árvore-B+ Pré-Fixada

- Estrutura híbrida
 - chaves
 - organizadas como *árvore-B*
 - nós folhas
 - consistem em blocos de *sequence set*
 - Pré-fixada simples
 - armazena na árvore as cadeias separadoras mínimas entre cada par de blocos
-

Separadores

- Características
 - são mantidos no índice, ao invés das chaves de busca
 - possuem tamanho variável
- Exemplo

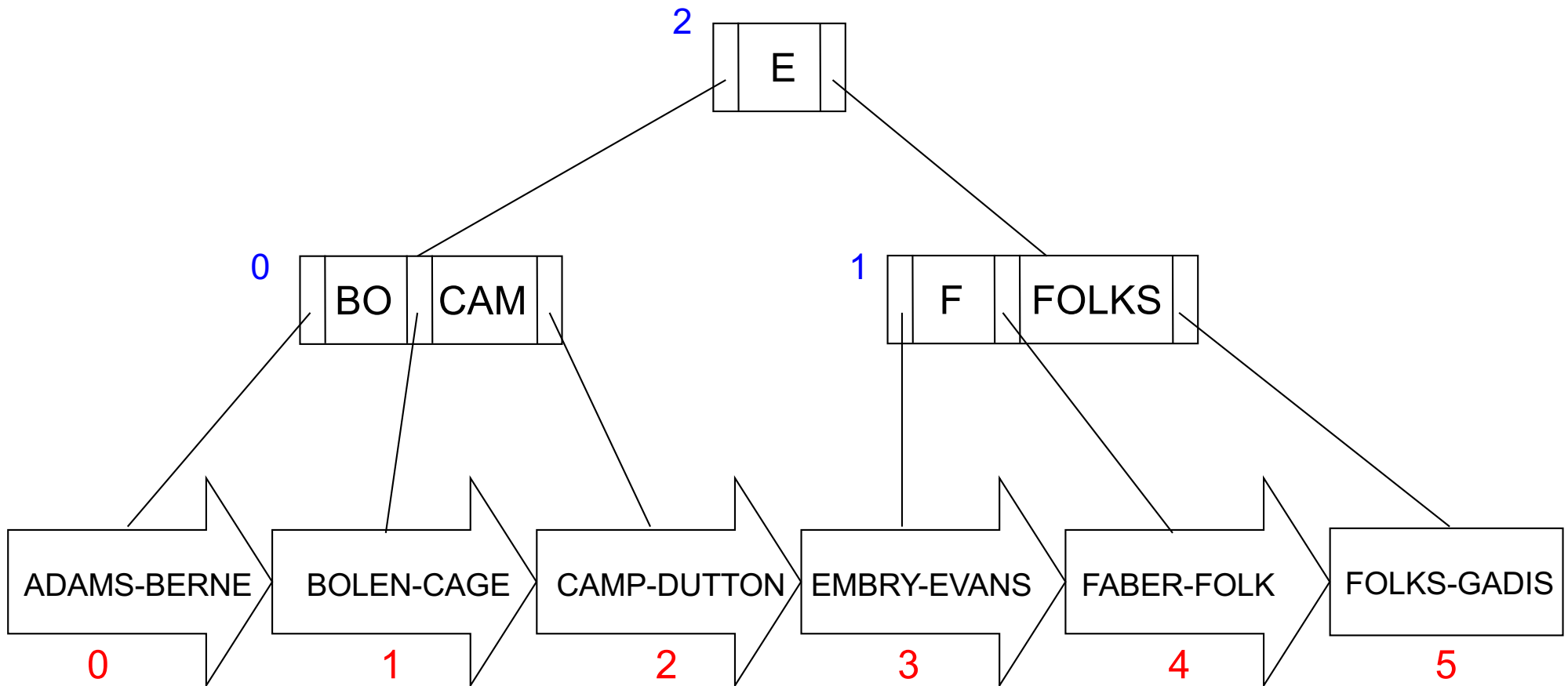


Separadores

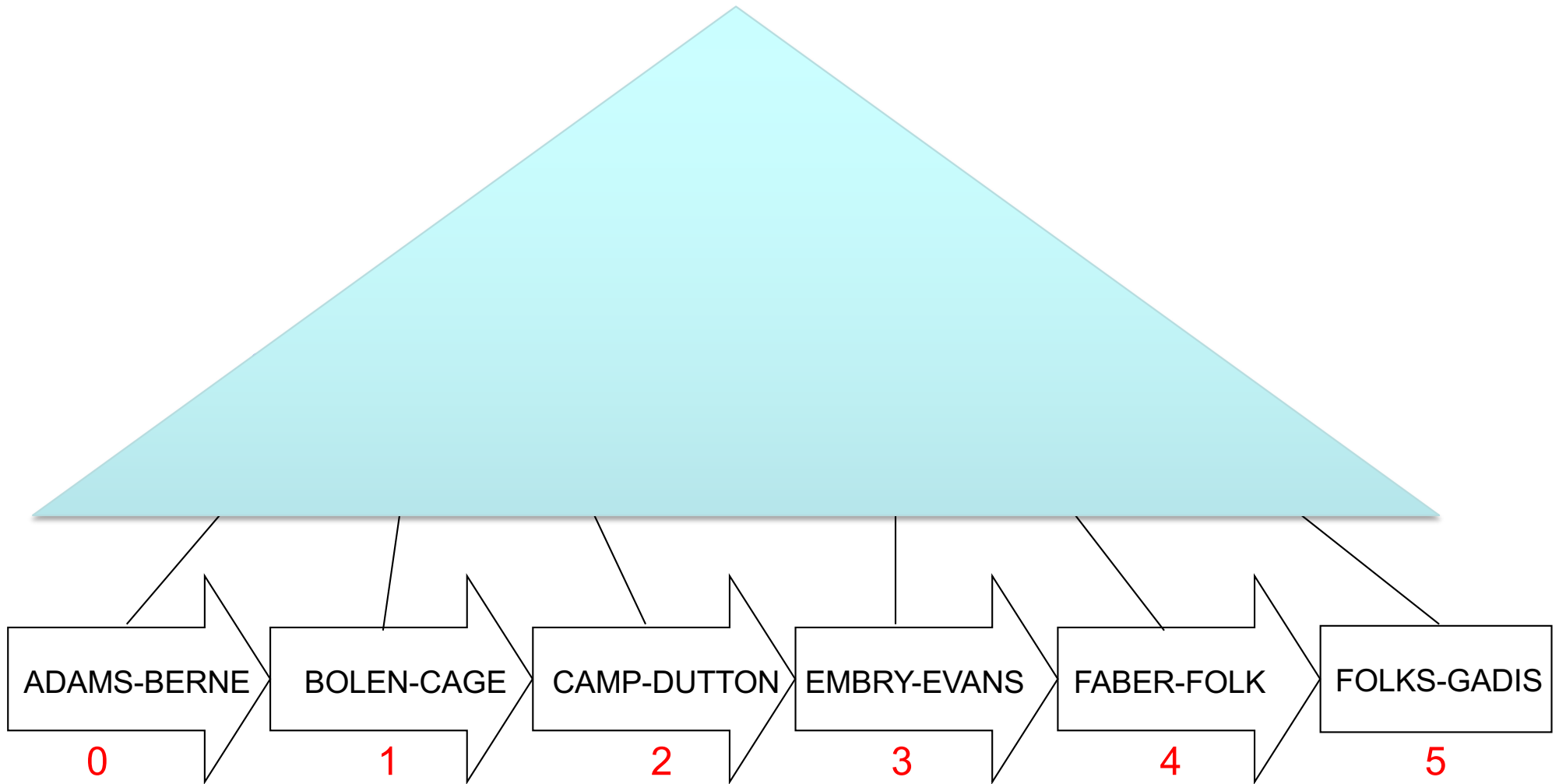
- Desafio
 - escolher o menor separador para utilizar no índice
- Tabela de decisão

chave de busca x separador	decisão
chave < separador	procure à esquerda
chave = separador	procure à direita
chave > separador	procure à direita

Árvore-B+ Pré-Fixada



Árvore-B+ Pré-Fixada



Pesquisa

- Passos
 - primeiro: *Árvore-B⁺*
 - segundo: *Sequence Set*
- Inserção e Remoção
 - iniciam-se pela pesquisa

buscas são
sempre realizadas
a partir do arquivo
arquivo de índice !

Inserção e Remoção

- Primeiro passo: *Sequence Set*
 - inserir ou remover o dado
 - tratar *split*, concatenação e redistribuição (se necessário)
- Segundo passo: *Árvore-B⁺*
 - se *split* no *sequence set*,
inserir um novo separador no índice
 - se *concatenação* no *sequence set*
remover um separador do índice
 - se *distribuição* no *sequence set*
alterar o valor do separador no índice

inserções e
remoções são
sempre
realizadas a
partir do arquivo
de dados !

Observações Adicionais

- Tamanho físico de um nó no índice (i.e., árvore-B⁺) = Tamanho físico de um bloco no *sequence set*
 - Escolha direcionada pelos mesmos quesitos
 - tamanho do bloco
 - características do disco
 - quantidade de memória disponível
-

Observações Adicionais

- Tamanho físico de um nó no índice (i.e., árvore-B⁺) = Tamanho físico de um bloco no *sequence set*
 - Facilidade para a implementação da árvore-B⁺ virtual
 - Uso de um mesmo arquivo para armazenar os blocos do índice e os blocos do *sequence set*
 - evita *seeks* entre dois arquivos separados
-