

EDCO4B

ESTRUTURAS DE DADOS 2

Aula 13A - B-Tree

Prof. Rafael G. Mantovani

Licença

Este trabalho está licenciado com uma Licença CC BY-NC-ND 4.0:



maiores informações:

https://creativecommons.org/licenses/by-nc-nd/4.0/deed.pt_BR

Roteiro



- 1** Introdução
- 2** Definição de B-Tree
- 3** Inserção de elementos
- 4** Exercícios
- 5** Referências

Roteiro

- 1 Introdução**
- 2 Definição de B-Tree**
- 3 Inserção de elementos**
- 4 Exercícios**
- 5 Referências**

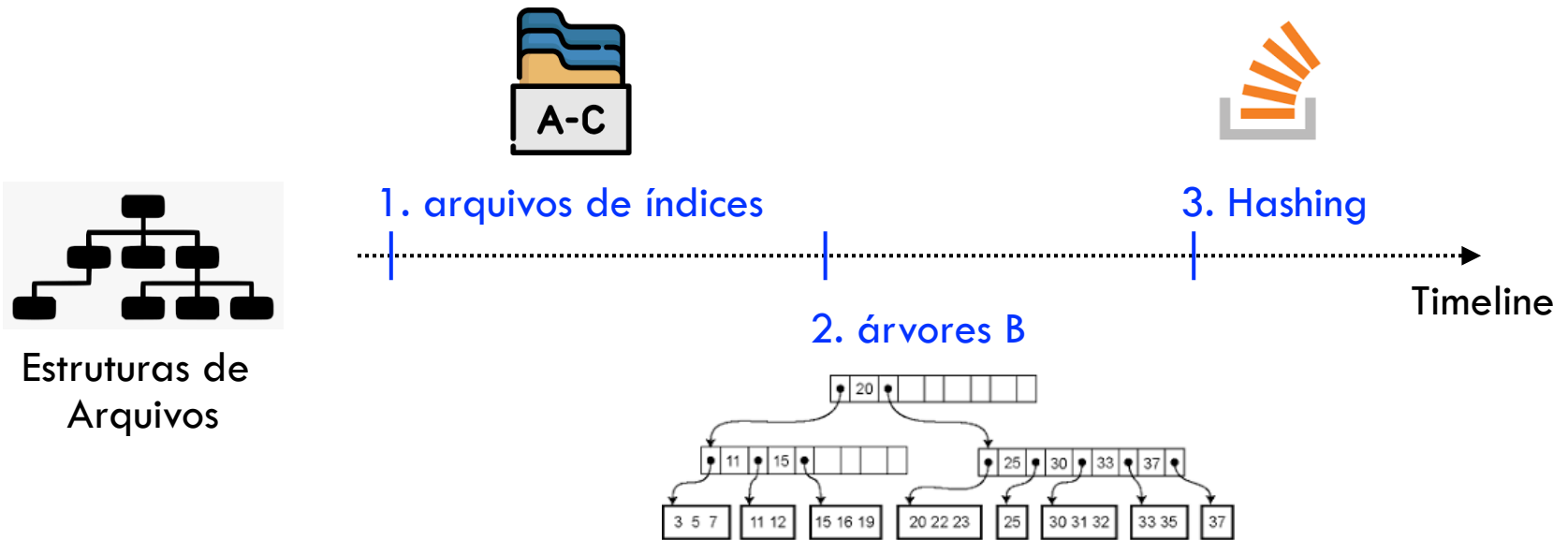
Introdução



- O que já fizemos:

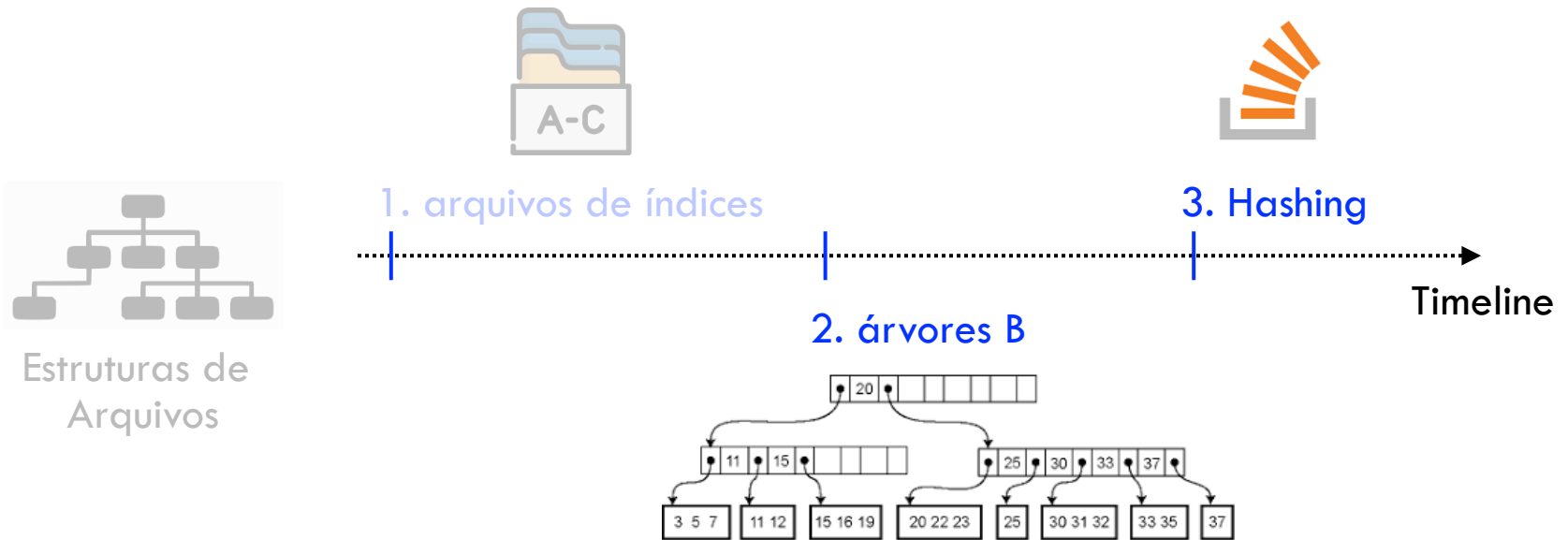
Introdução

- O que já fizemos:



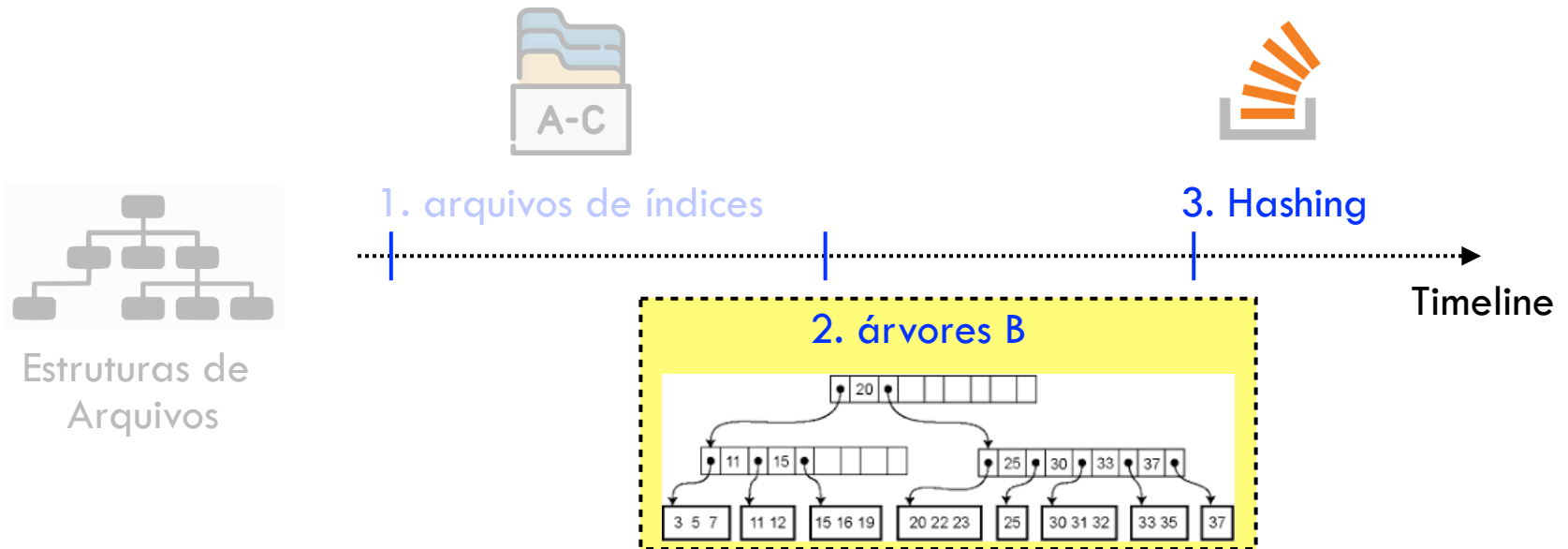
Introdução

- O que já fizemos:



Introdução

- O que já fizemos:



Introdução



- Índices possuem algumas limitações:

Introdução

- Índices possuem algumas limitações:
 - Não conseguem armazenar uma quantidade mto grande de dados
 - toda vez que inserimos, atualizamos, modificamos ou removemos algum registro, os índices precisam ser atualizados (no disco). Requer tempo adicional ...

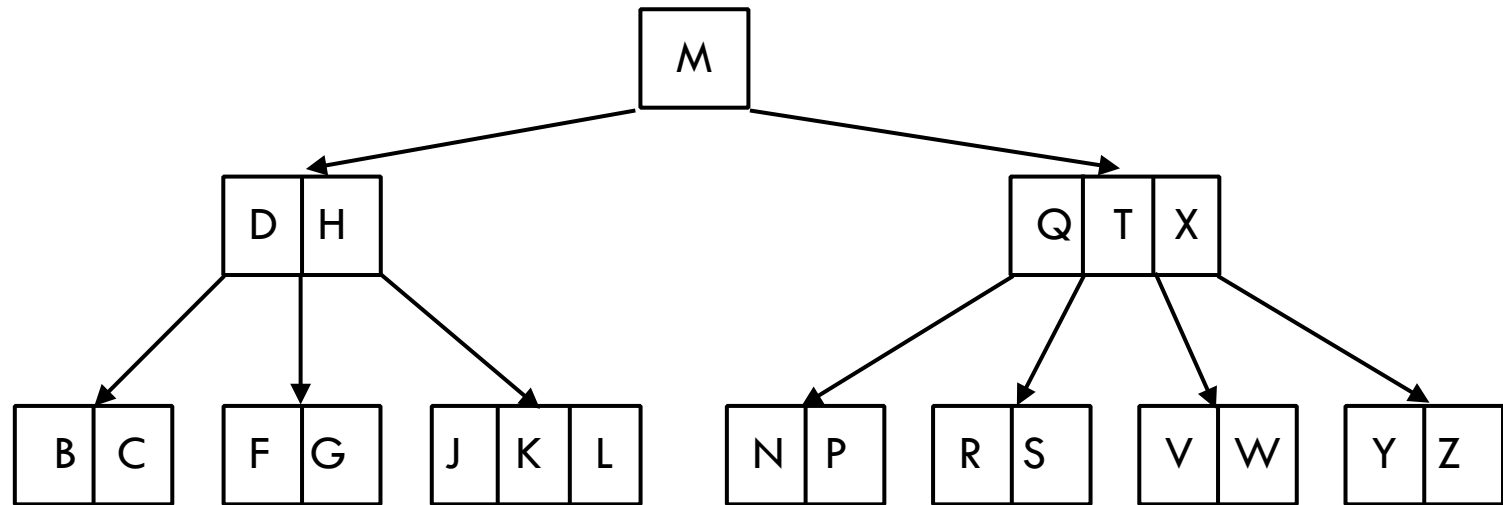
Introdução



- Solução: usar **árvores!**

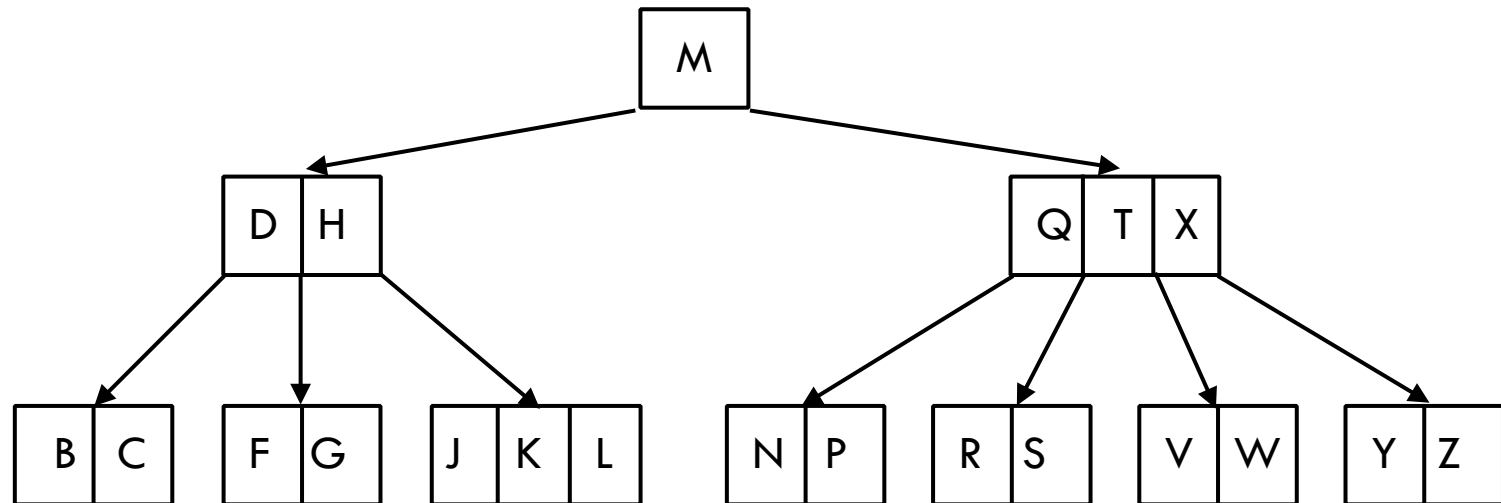
Introdução

- Solução: usar **árvores!**



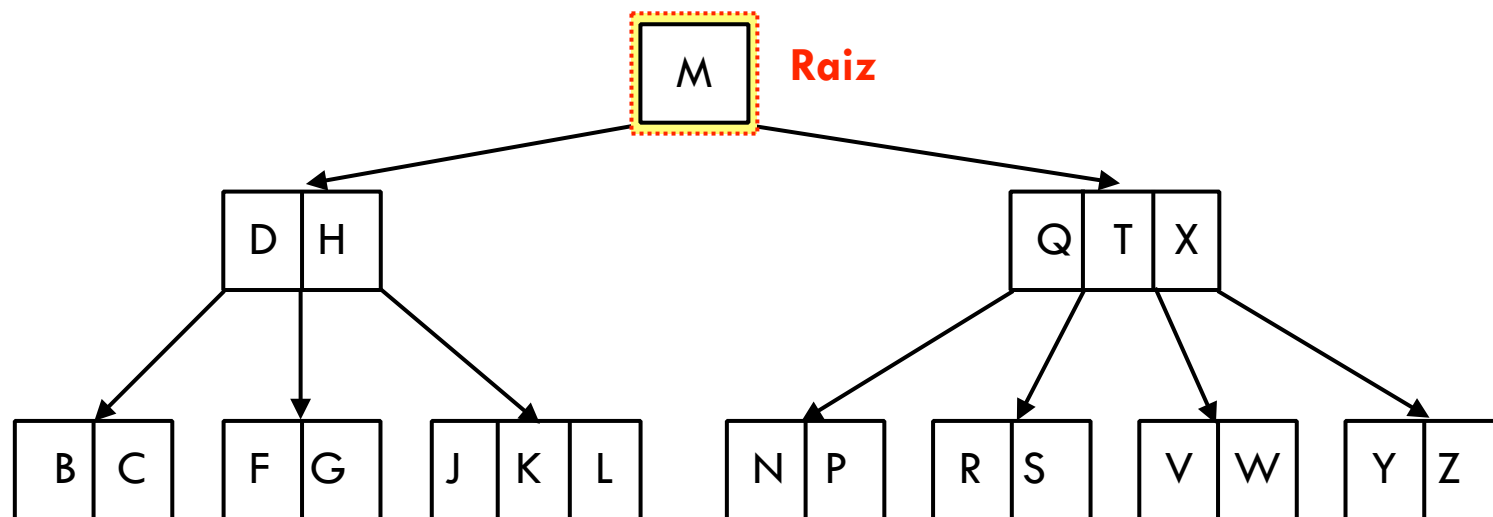
Introdução

B-Tree



Introdução

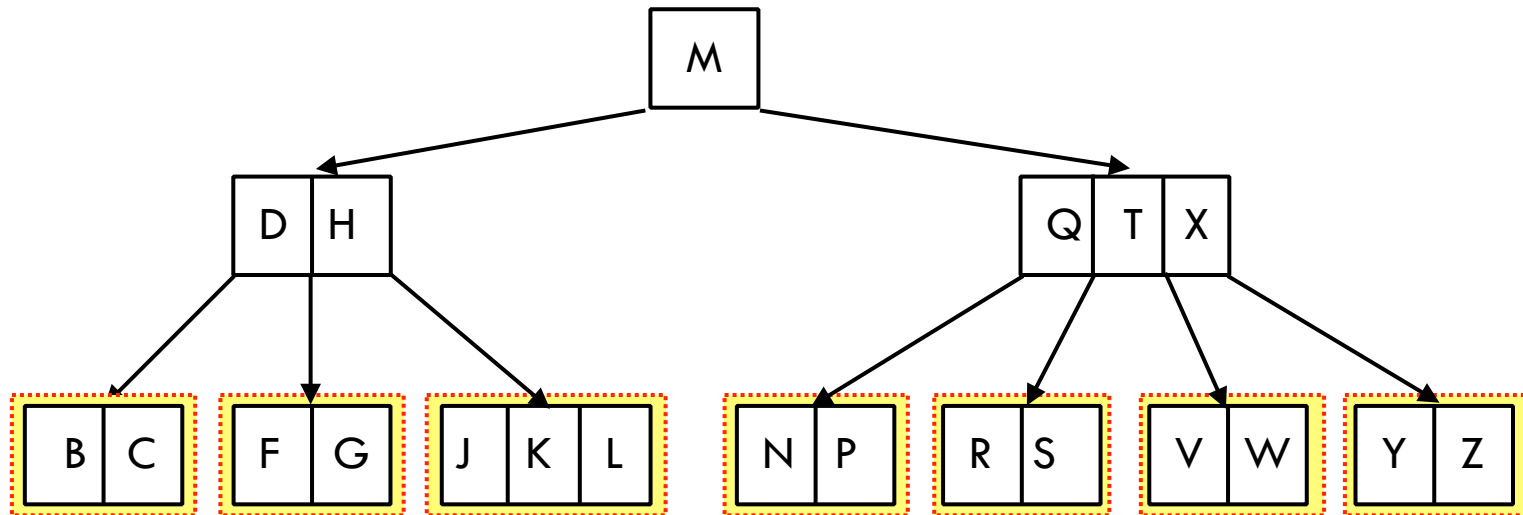
B-Tree



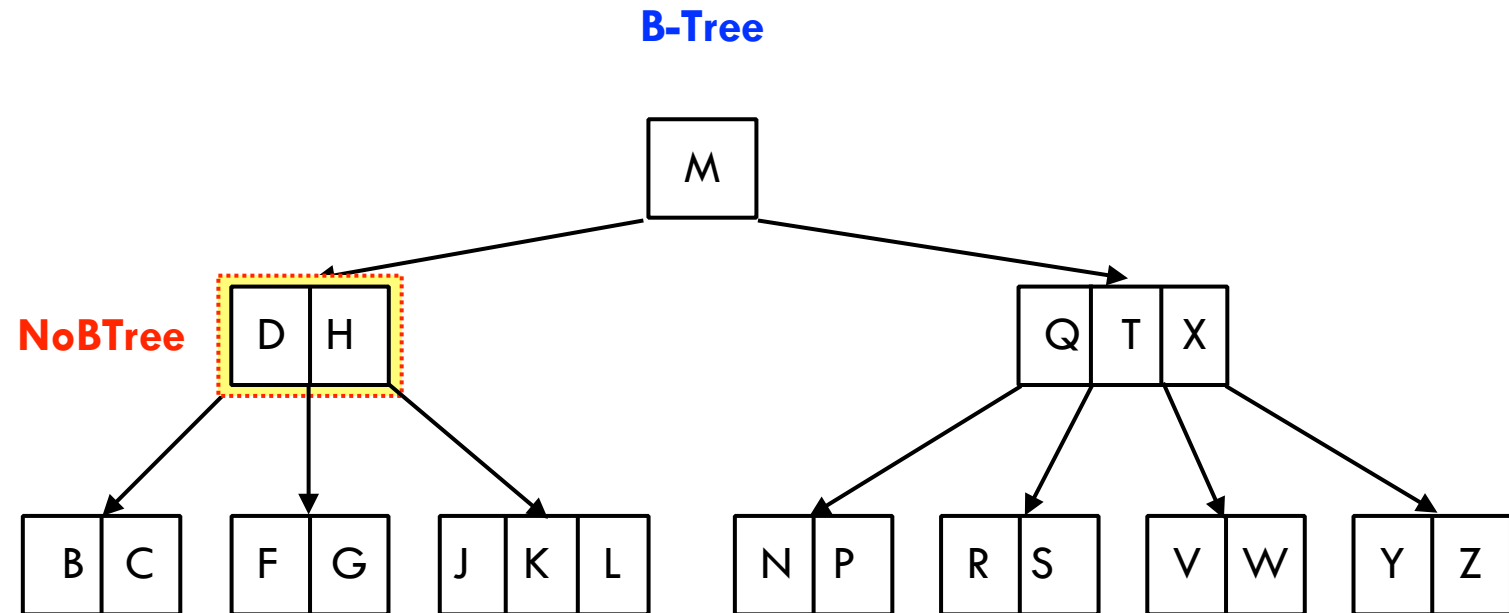
Introdução

B-Tree

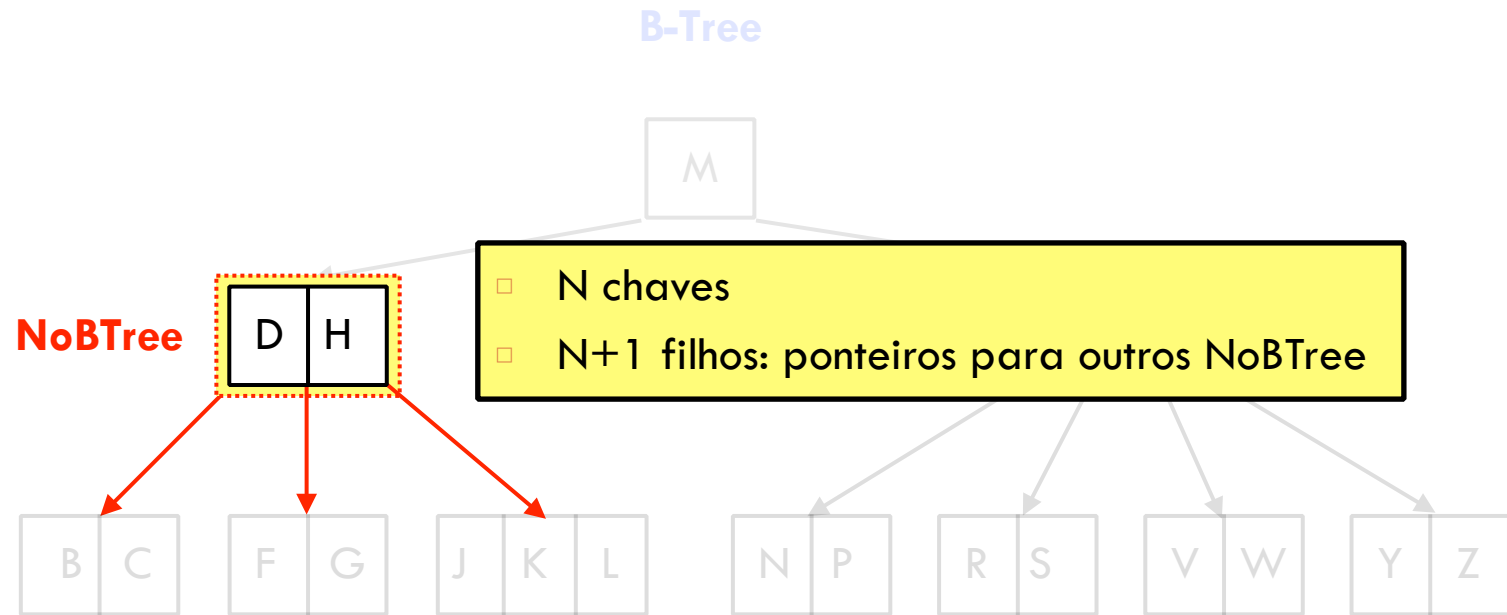
Folhas



Introdução

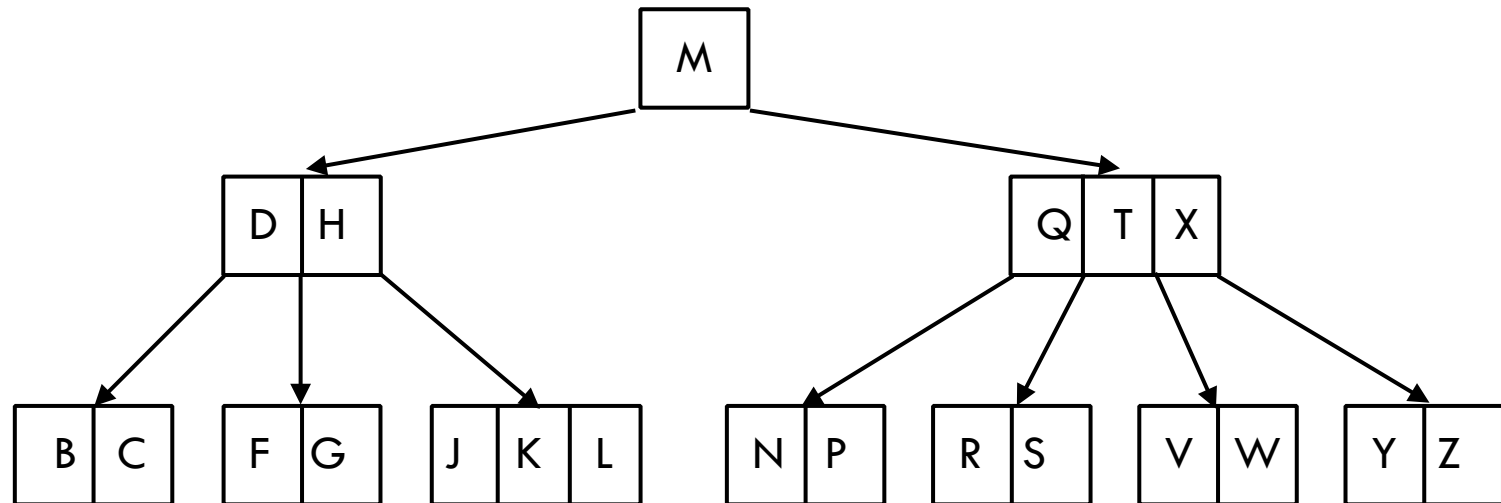


Introdução



Introdução

B-Tree

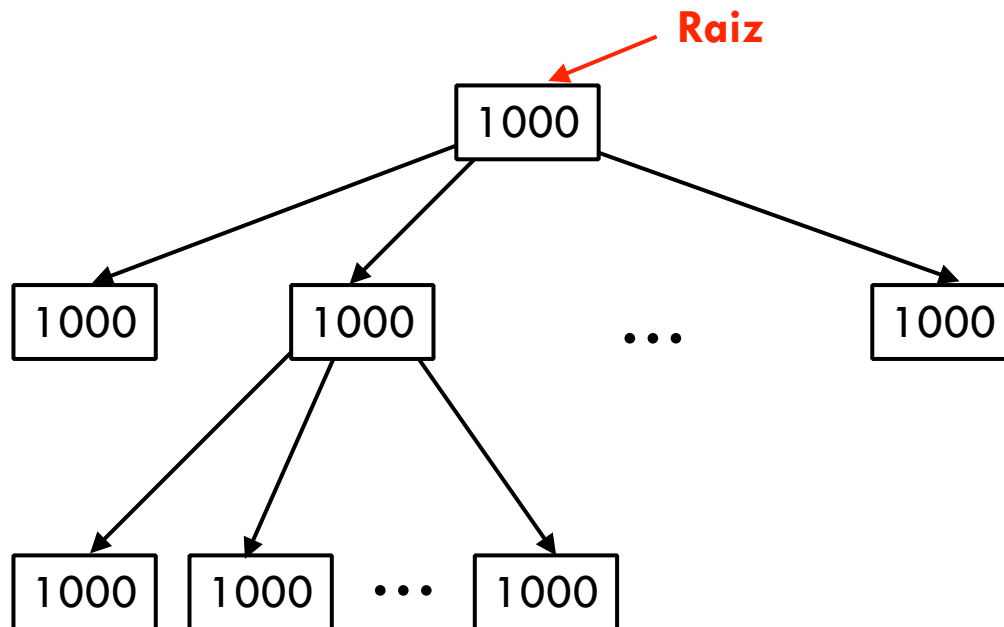


- Minimizam as operações de I/O no disco
- Muitos sistemas de banco de dados usam B-Tree ou variações para indexação de arquivos e armazenamento de informações

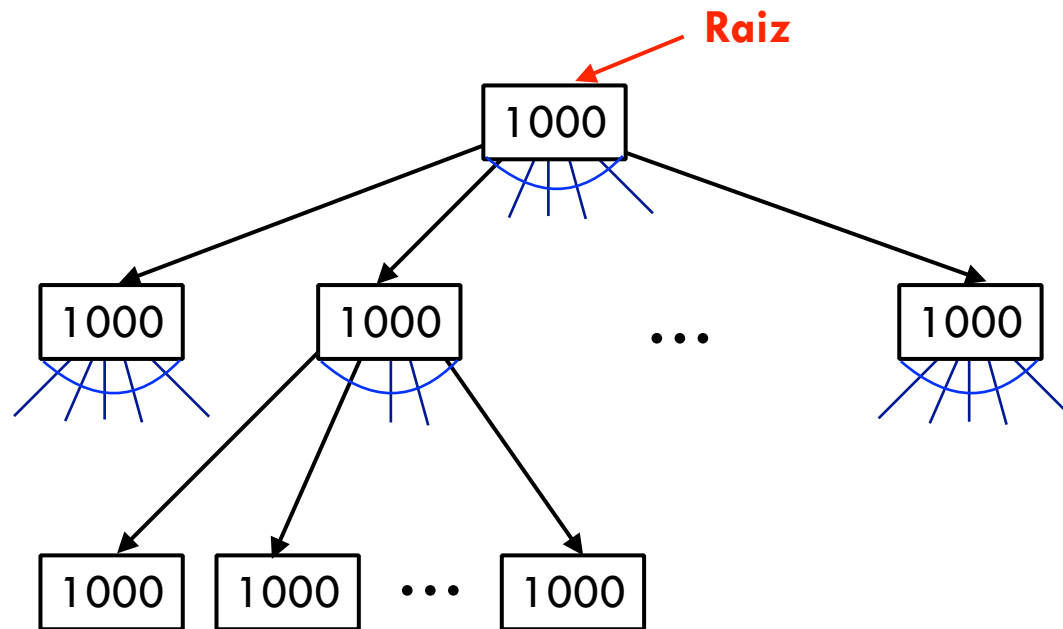
Introdução

- B-Trees para armazenamento secundário:
 - Informação é dividida em “**páginas**” de tamanho igual
 - Cada leitura/escrita altera/afeta uma ou mais páginas
 - B-Trees copiam algumas páginas na memória, e as escreve de volta no disco se foram modificadas
 - Geralmente temos B-Trees com nós de ordem 50 a 2000 (maior, melhor)
 - Maiores ordens tornam as árvores **mais baixas**, e requerem **menos operações**
 - O **nó raiz** é sempre mantido na **memória principal**

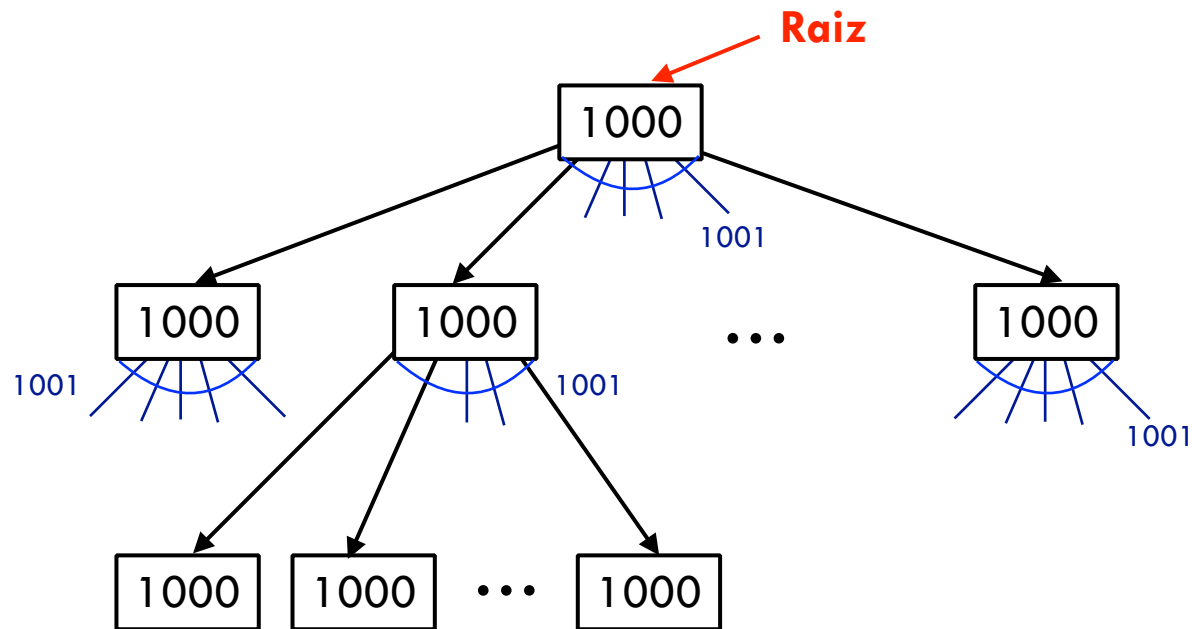
Introdução



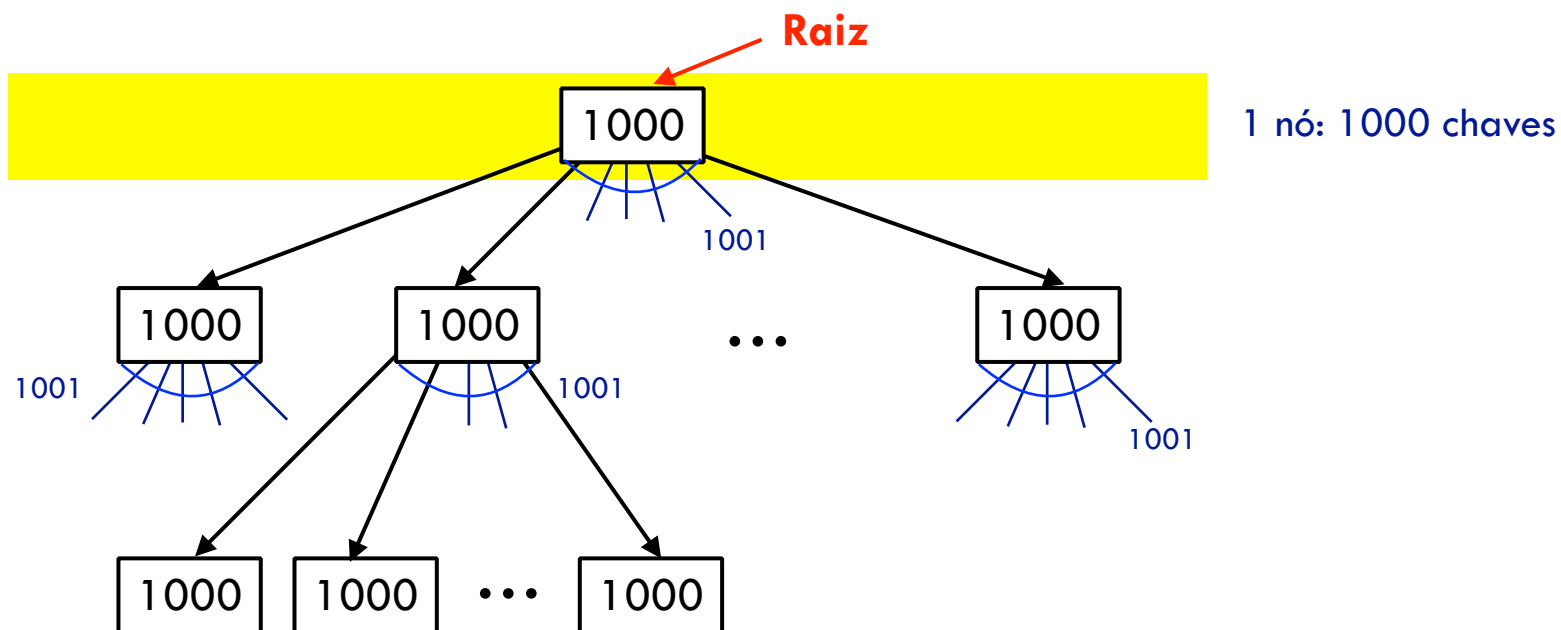
Introdução



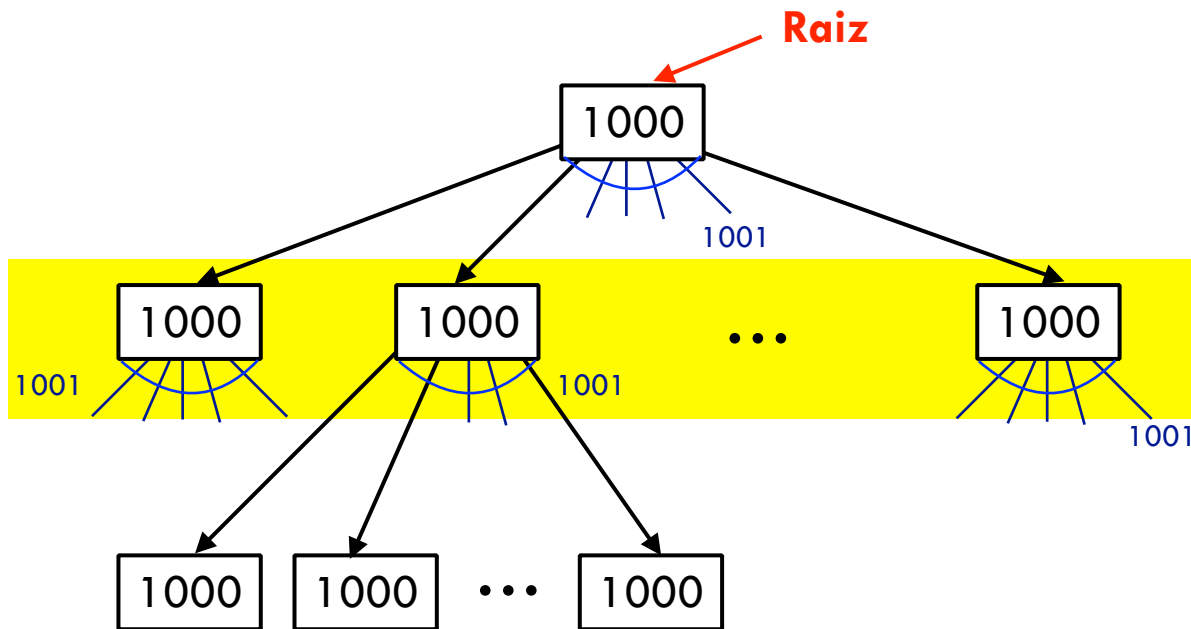
Introdução



Introdução



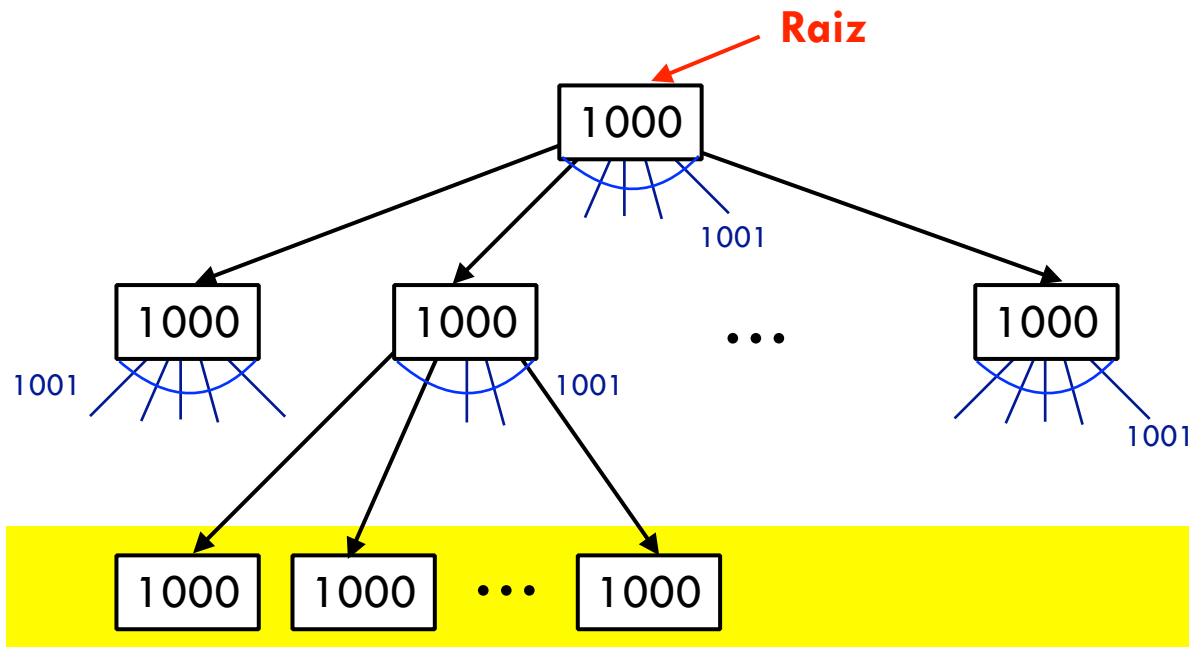
Introdução



1 nó: 1000 chaves

1001 nós:
1.001.000 chaves

Introdução

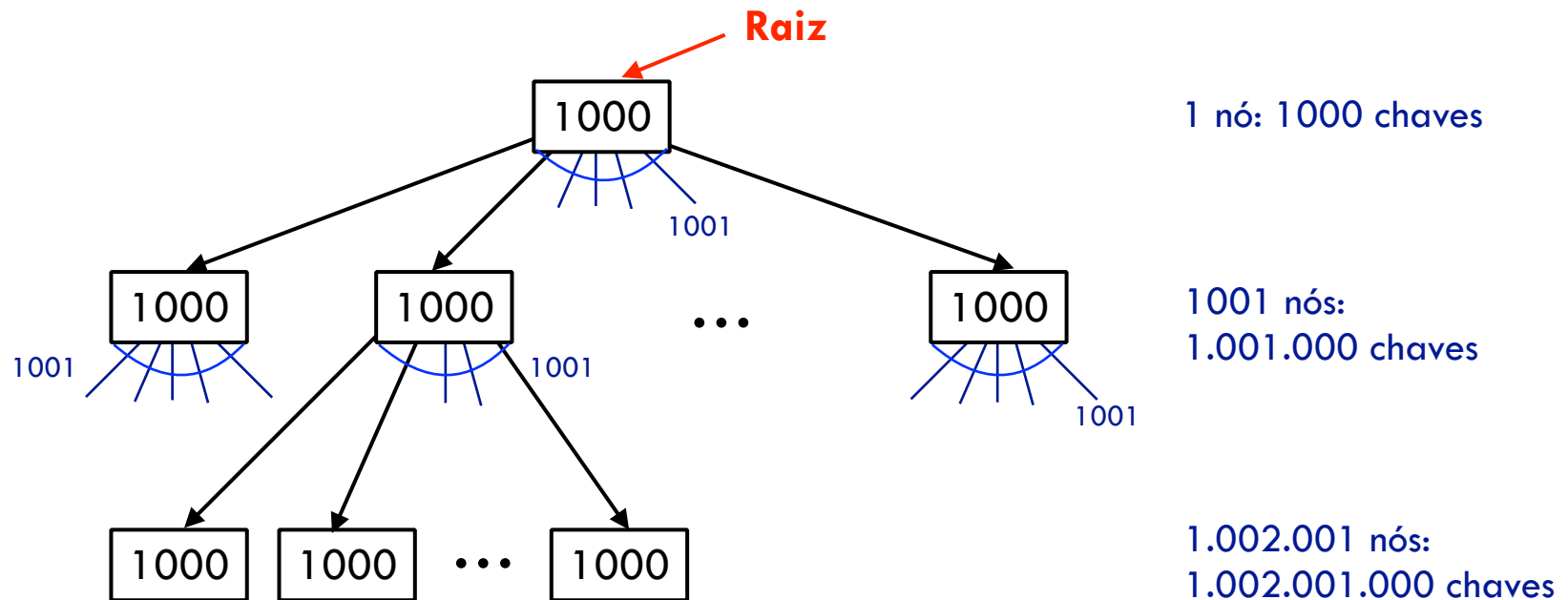


1 nó: 1000 chaves

1001 nós:
1.001.000 chaves

1.002.001 nós:
1.002.001.000 chaves

Introdução



* B-Tree de altura 2 com 1 bilhão de chaves

* Raiz é mantida na memória. No **máximo 2 leituras** para recuperar qualquer informação

Roteiro



- 1 Introdução
- 2 Definição de B-Tree
- 3 Inserção de elementos
- 4 Exercícios
- 5 Referências

B-Tree



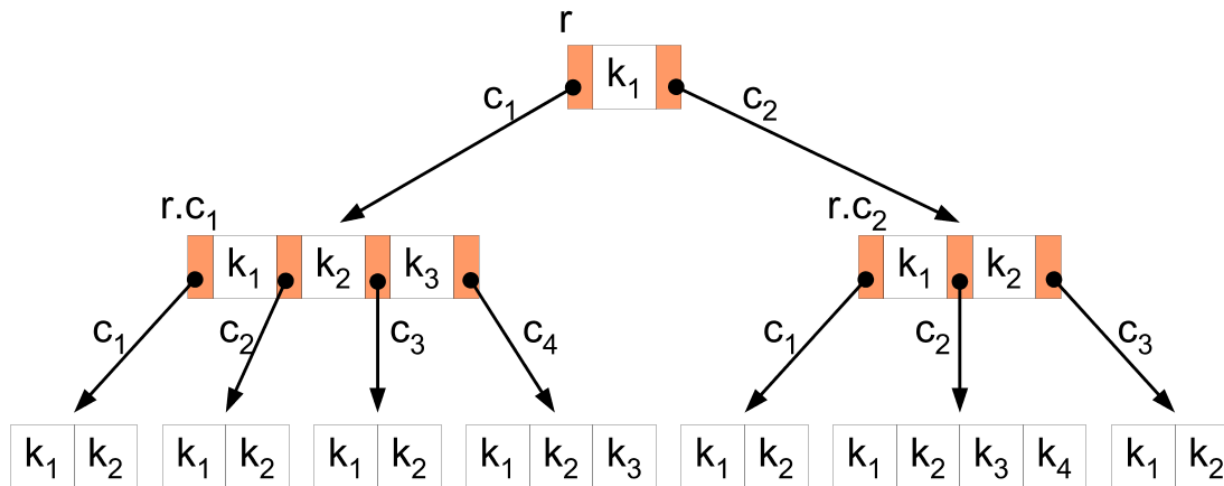
B-Tree



- B-TREE é uma árvore enraizada que possui as seguintes propriedades:

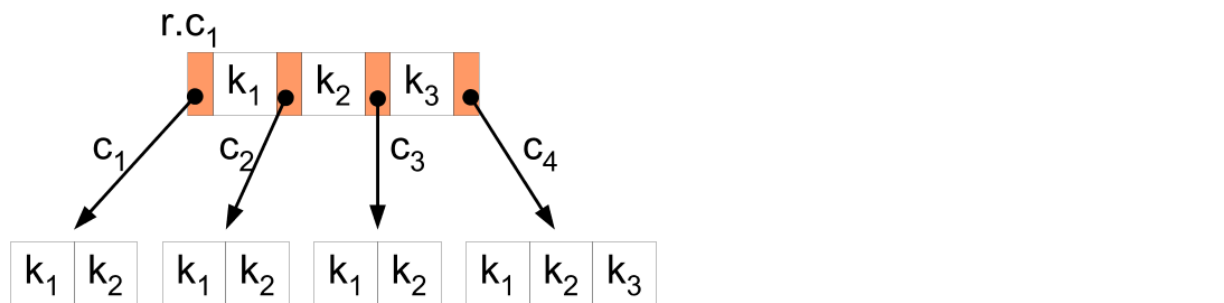
B-Tree

- B-TREE é uma árvore enraizada que possui as seguintes propriedades:



B-Tree

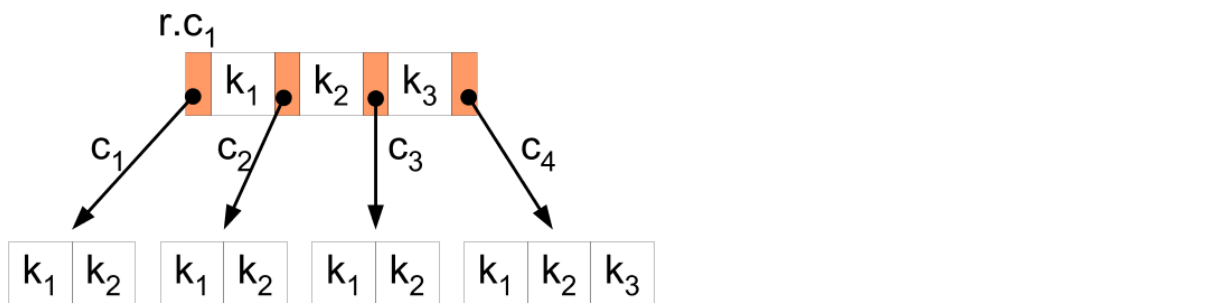
- B-TREE é uma árvore enraizada que possui as seguintes propriedades:



B-Tree

- B-TREE é uma árvore enraizada que possui as seguintes propriedades:

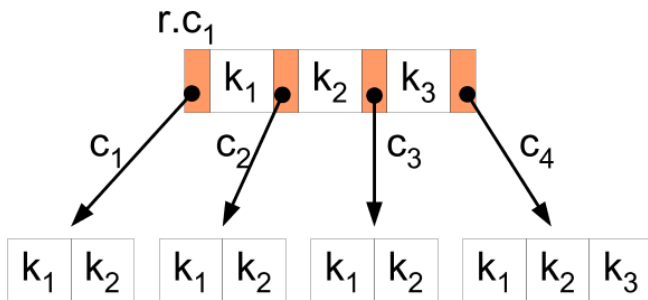
1) Todo nó possui alguns atributos:



B-Tree

- B-TREE é uma árvore enraizada que possui as seguintes propriedades:

1) Todo nó possui alguns atributos:



NoBTree

k_i : conjunto de chaves

c_i : conjunto de ponteiros

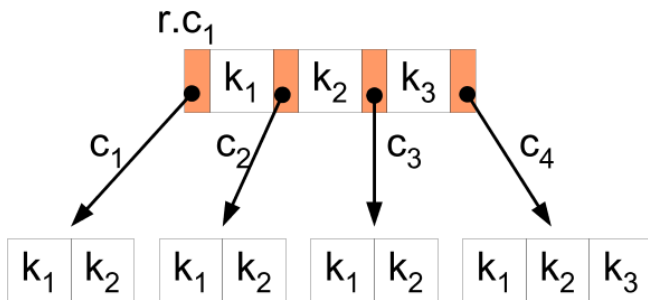
n : quantidade de chaves armazenadas no nó

folha: se é folha ou nó interno

B-Tree

- B-TREE é uma árvore enraizada que possui as seguintes propriedades:

1) Todo nó possui alguns atributos:



NoBTree

k_i : conjunto de chaves

c_i : conjunto de ponteiros

n : quantidade de chaves armazenadas no nó

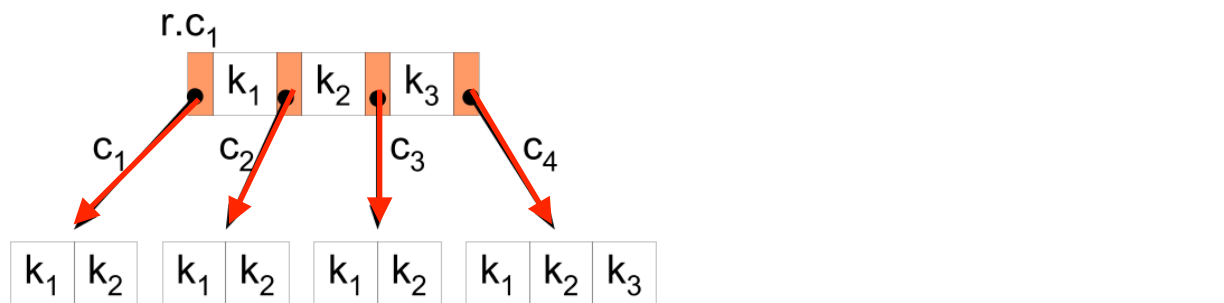
folha: se é folha ou nó interno

* As chaves estão armazenadas em ordem crescente

$$x.key_1 \leq x.key_2 \leq x.key_3 \leq \dots \leq x.key_{xn}$$

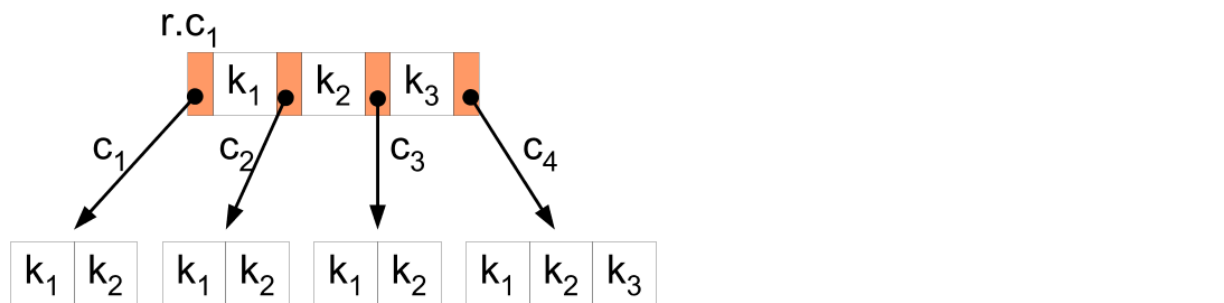
B-Tree

2) Cada **nó interno** contém também **$x+1$ ponteiros** para acesso aos seus filhos.



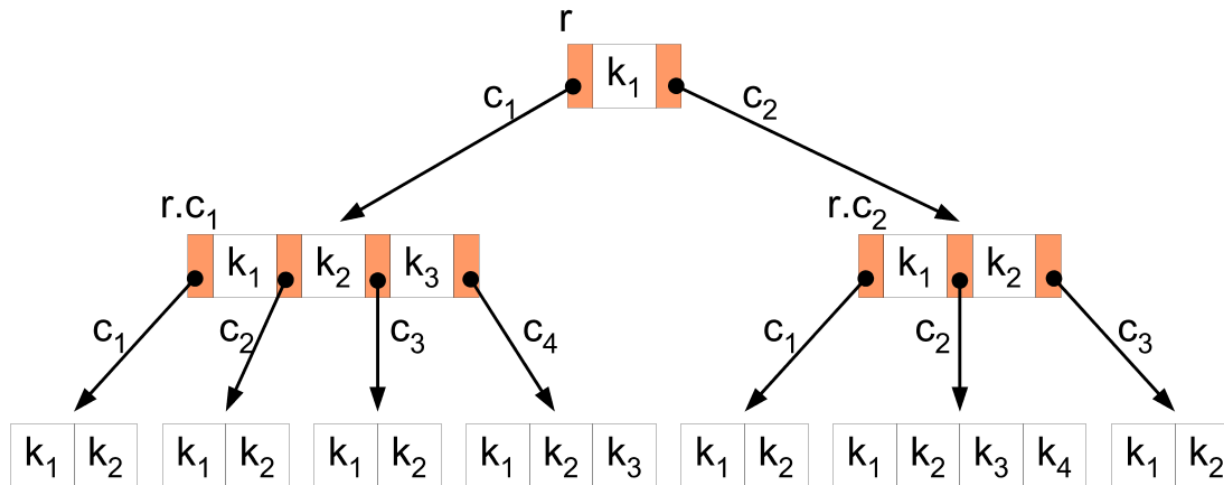
B-Tree

3) As chaves k_i separam os intervalos de chaves armazenados em cada sub-árvore



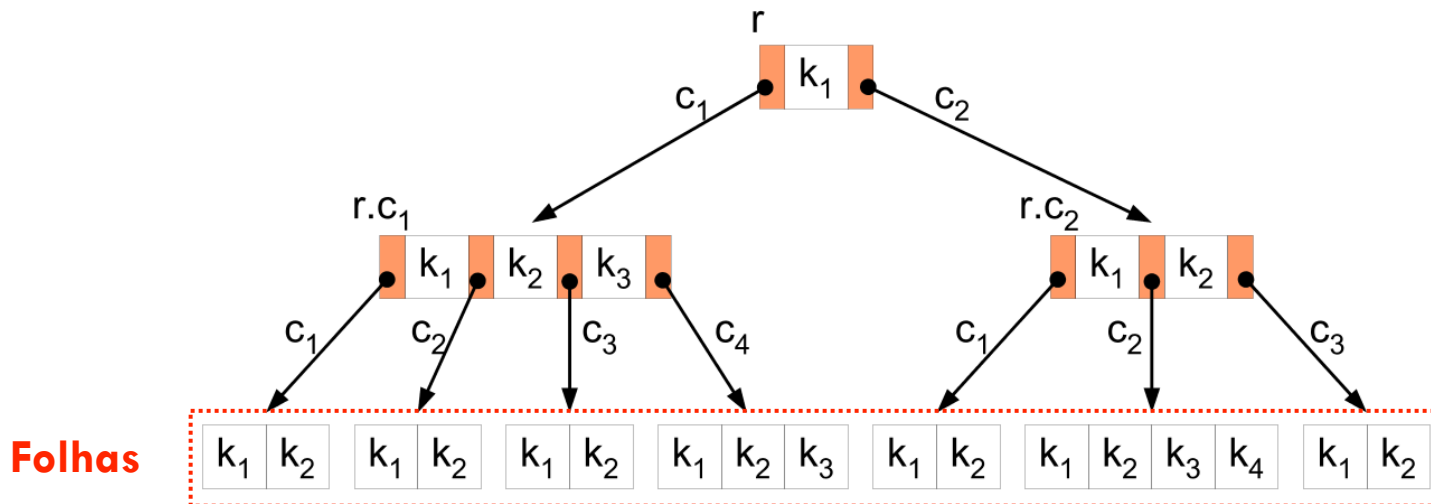
B-Tree

4) **Todas as folhas** tem a **mesma profundidade**, que é a altura da árvore



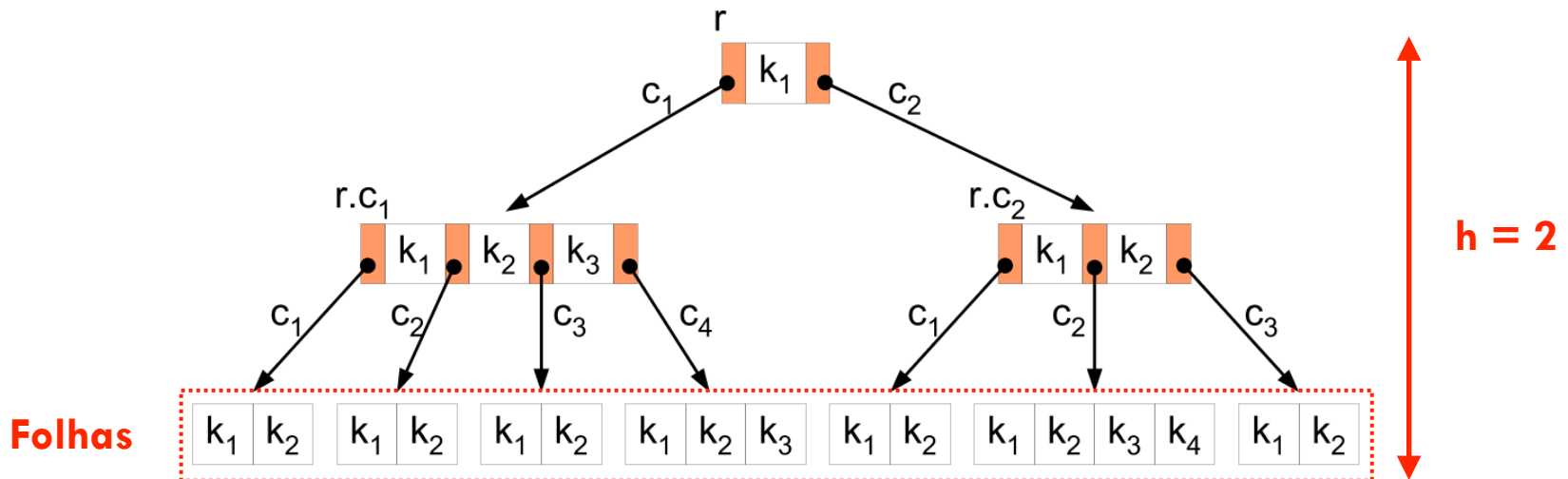
B-Tree

4) Todas as folhas tem a mesma profundidade, que é a altura da árvore



B-Tree

4) Todas as folhas tem a mesma profundidade, que é a altura da árvore

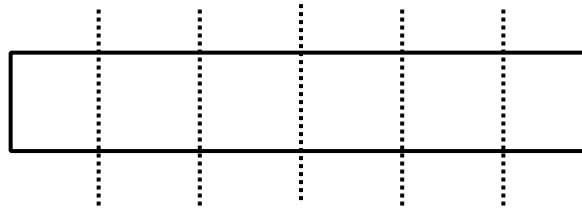


B-Tree

5) Os nós possuem quantidades mínimas e máximas de chaves que podem conter. Esse intervalo é definido com base em um inteiro fixo (t) chamado de **grau mínimo** da B-Tree

B-Tree

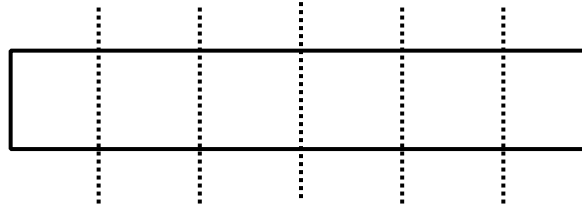
5) Os nós possuem quantidades mínimas e máximas de chaves que podem conter. Esse intervalo é definido com base em um inteiro fixo (t) chamado de **grau mínimo** da B-Tree



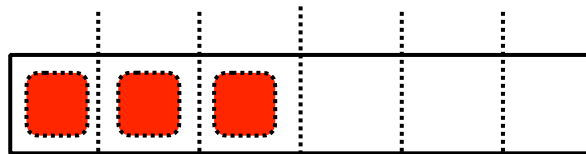
B-Tree

5) Os nós possuem quantidades mínimas e máximas de chaves que podem conter. Esse intervalo é definido com base em um inteiro fixo (t) chamado de **grau mínimo** da B-Tree

$t = 3$



MIN

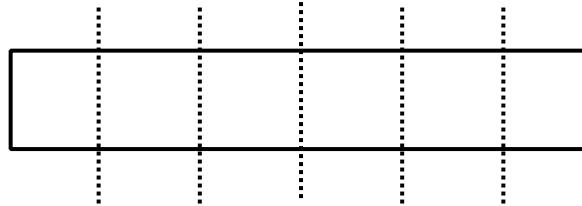


Todo nó interno diferente da raiz tem no **mínimo** t filhos

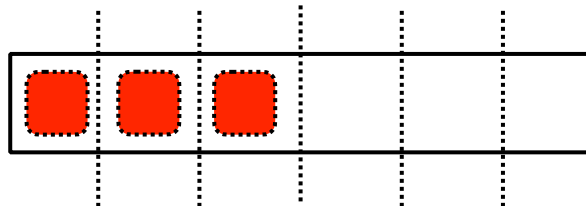
B-Tree

5) Os nós possuem quantidades mínimas e máximas de chaves que podem conter. Esse intervalo é definido com base em um inteiro fixo (t) chamado de **grau mínimo** da B-Tree

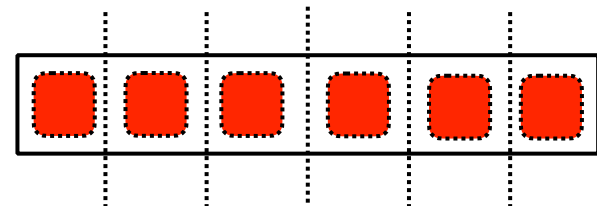
$t = 3$



MIN



MAX



Todo nó pode contar no máximo $2t - 1$ chaves
(CHEIO)

Roteiro



- 1 Introdução
- 2 Definição de B-Tree
- 3 Inserção de elementos
- 4 Exercícios
- 5 Referências

Inserção

- Inserir uma chave em uma B-Tree é mais complicado do que em uma árvore binária:

Inserção

- Inserir uma chave em uma B-Tree é mais complicado do que em uma árvore binária:
 1. **Inserir** a nova chave em um nó **folha** existente. Se árvore estiver vazia, criar um nó folha (raiz)
 2. **Não** podemos **inserir** uma chave **em** um **nó cheio**. Isso demanda uma operação de **split** (divisão), e um novo nó é criado
 3. O elemento mediano sobe para um novo nó pai, e o vetor cheio é dividido em dois filhos com $t-1$ chaves cada
 4. Se o pai, que recebe o elemento mediano, também está cheio, repete-se o processo para cima (em direção à raiz da árvore)

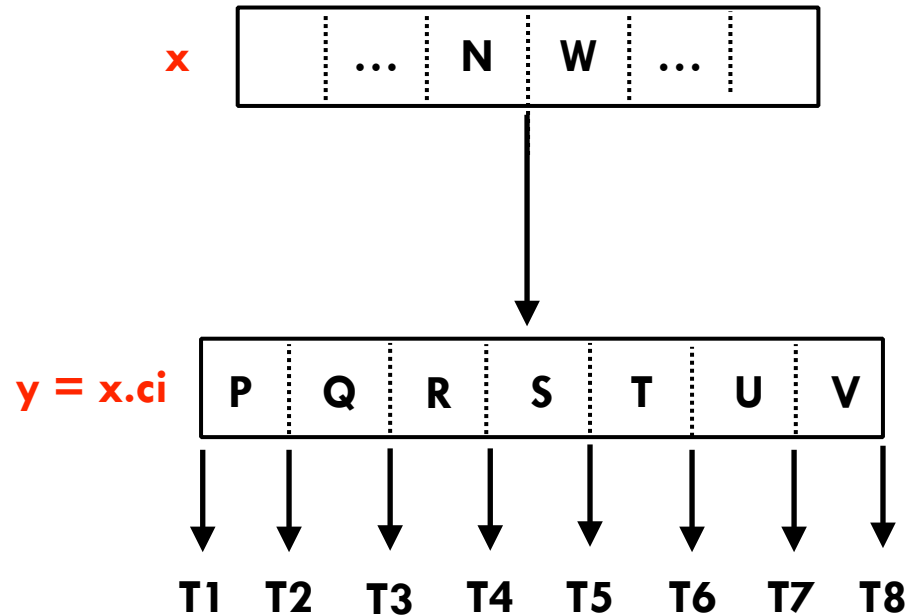
Inserção



- Split:

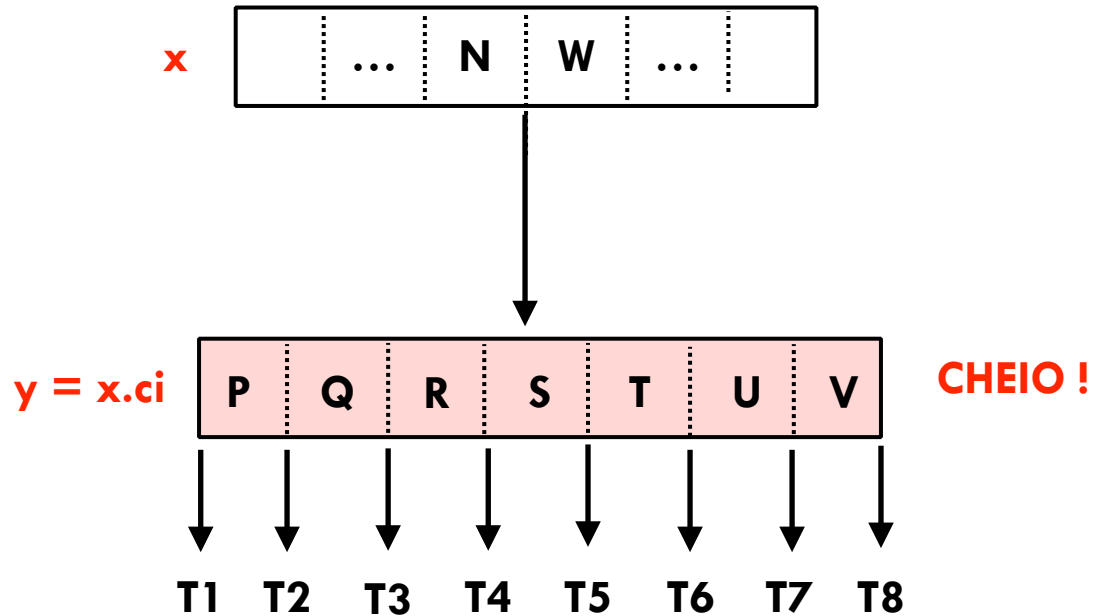
Insertão

- Split:



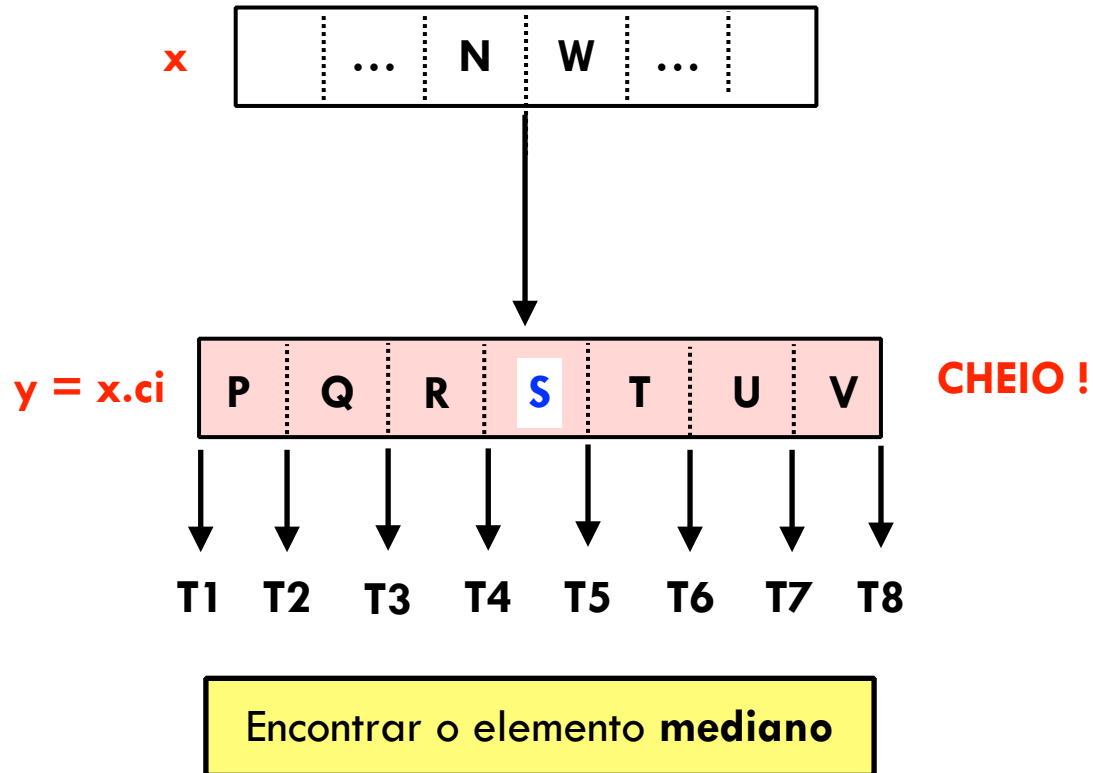
Insertão

- Split:



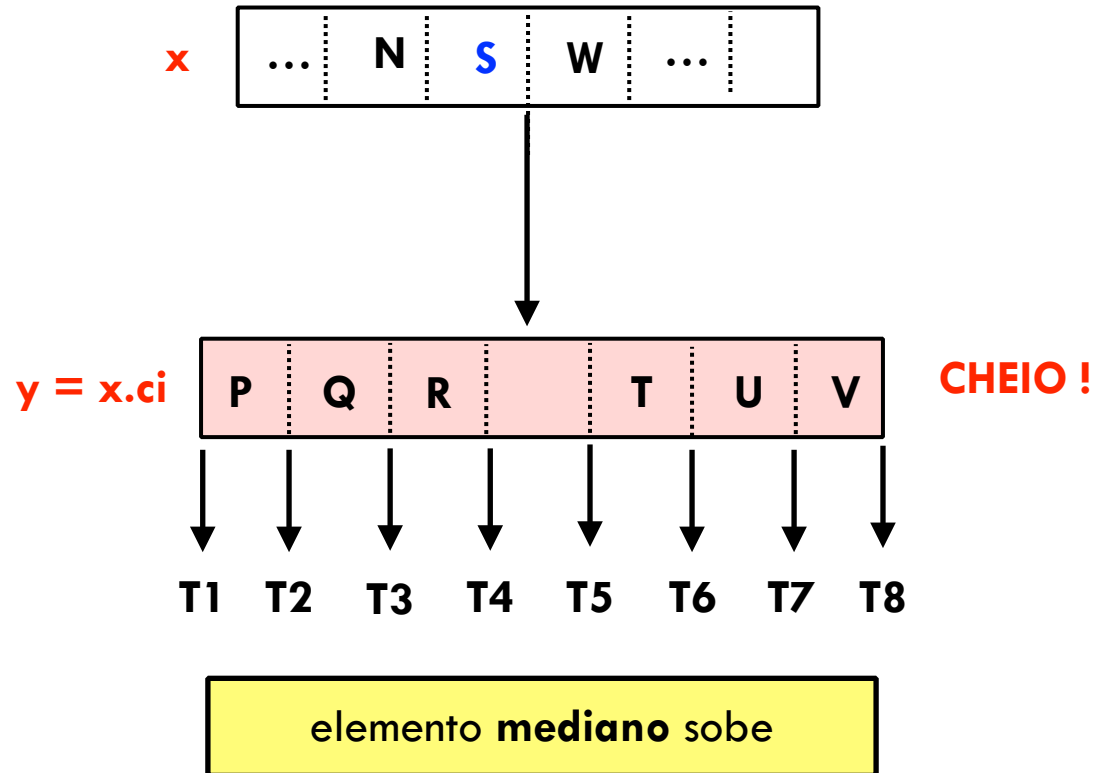
Inserção

- Split:



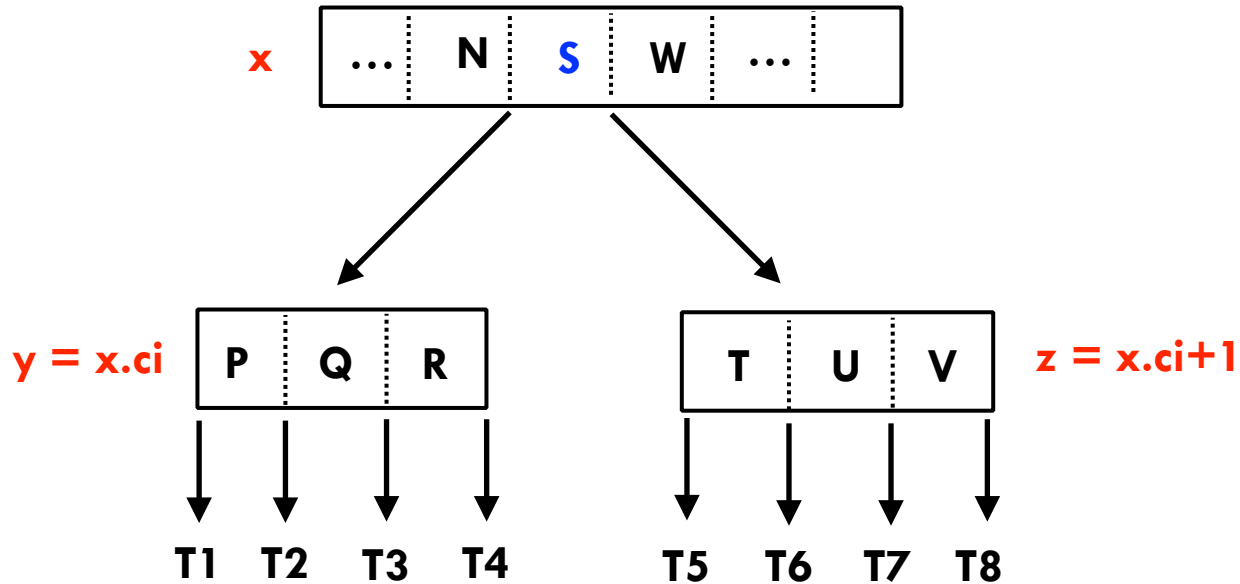
Inserção

- Split:



Inserção

□ Split:



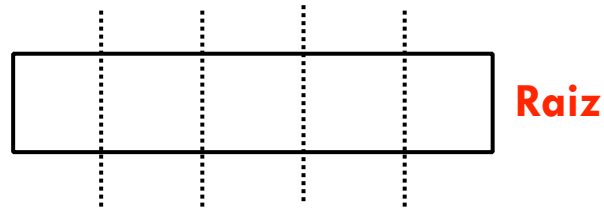
Dividimos (split) o vetor cheio em dois filhos com $\lceil t \rceil$ chaves cada

Exemplo

- $S = \{51, 69, -64, 34, -70, 35, -4, -97, -21, 41\}$, $\text{Ordem} = 5$

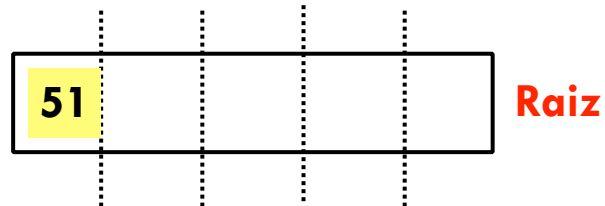
Exemplo

- $S = \{51, 69, -64, 34, -70, 35, -4, -97, -21, 41\}$, Ordem = 5



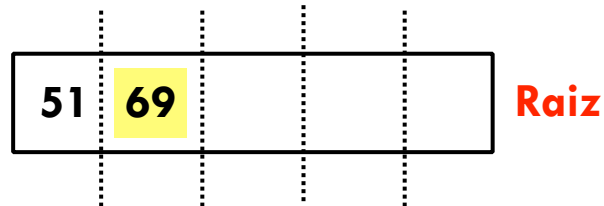
Exemplo

- $S = \{\cancel{51}, 69, -64, 34, -70, 35, -4, -97, -21, 41\}$, Ordem = 5



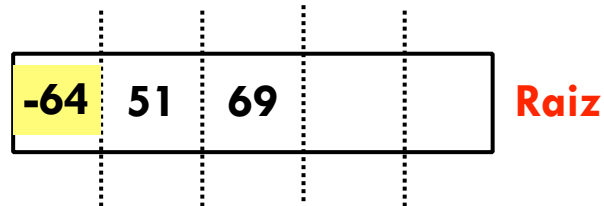
Exemplo

- $S = \{\cancel{51}, \cancel{69}, -64, 34, -70, 35, -4, -97, -21, 41\}$, Ordem = 5



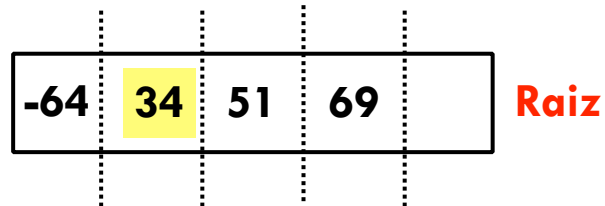
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, 34, -70, 35, -4, -97, -21, 41\}$, Ordem = 5



Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, -70, 35, -4, -97, -21, 41\}$, Ordem = 5



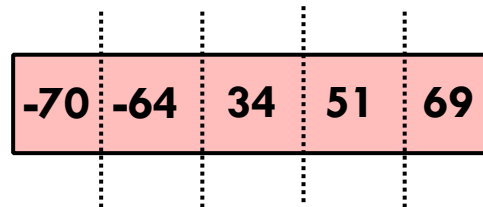
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, -70, 35, -4, -97, -21, 41\}$, Ordem = 5

-70	-64	34	51	69	Raiz
------------	------------	-----------	-----------	-----------	-------------

Exemplo

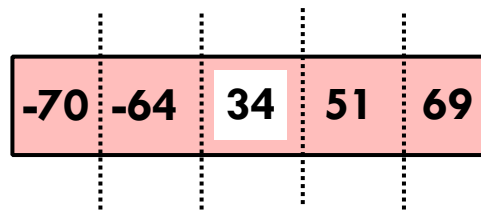
- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, -70, 35, -4, -97, -21, 41\}$, Ordem = 5



Raiz (cheio -> split)

Exemplo

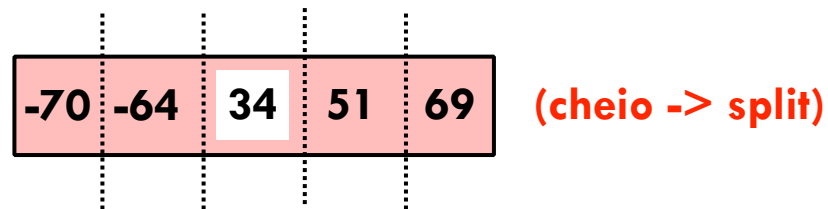
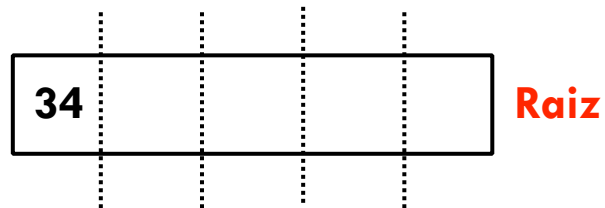
- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, 35, -4, -97, -21, 41\}$, Ordem = 5



Raiz (cheio -> split)

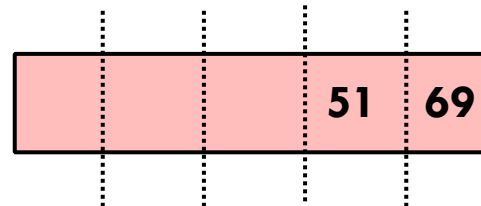
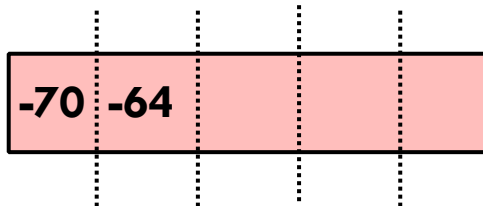
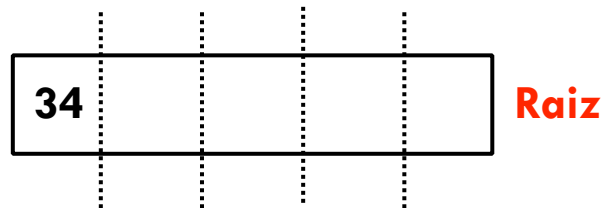
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, 35, -4, -97, -21, 41\}$, Ordem = 5



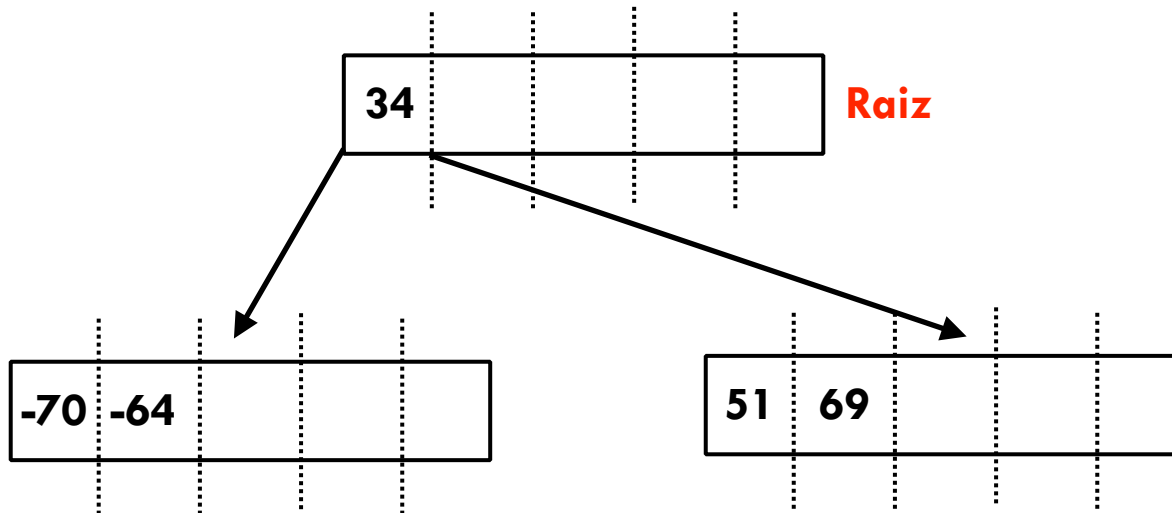
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, 35, -4, -97, -21, 41\}$, Ordem = 5



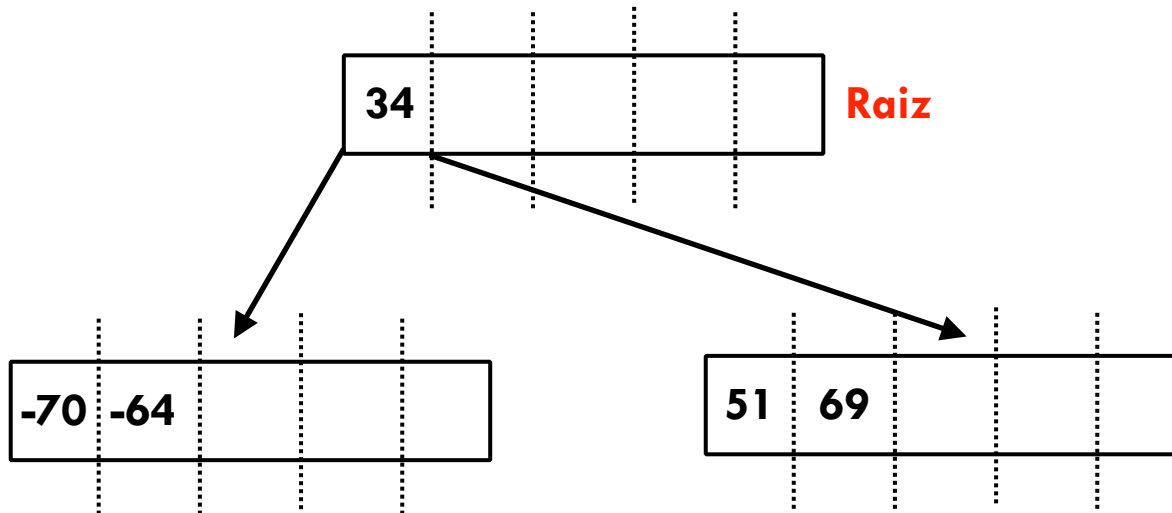
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, 35, -4, -97, -21, 41\}$, Ordem = 5



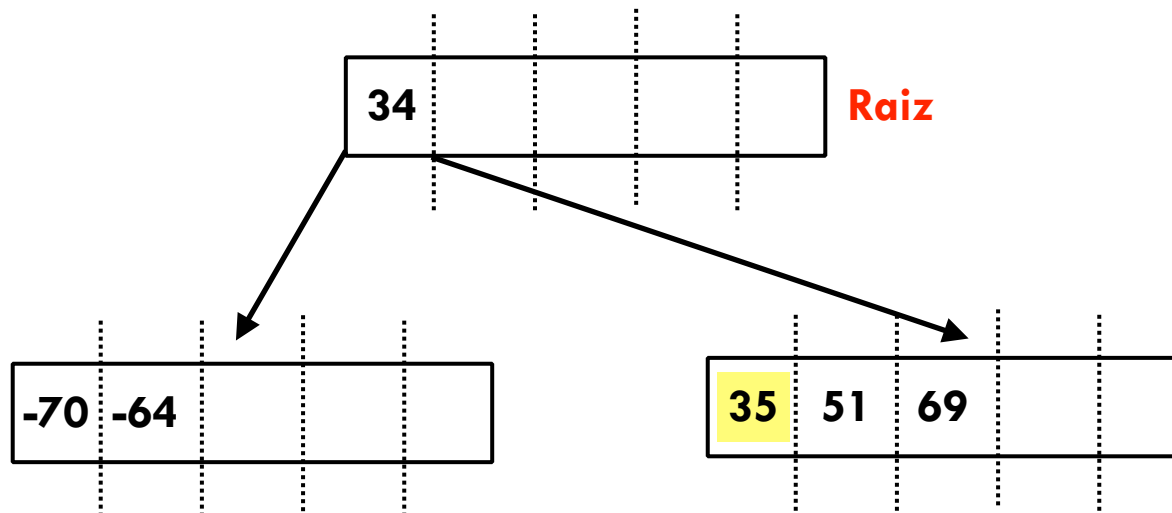
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, 35, -4, -97, -21, 41\}$, Ordem = 5



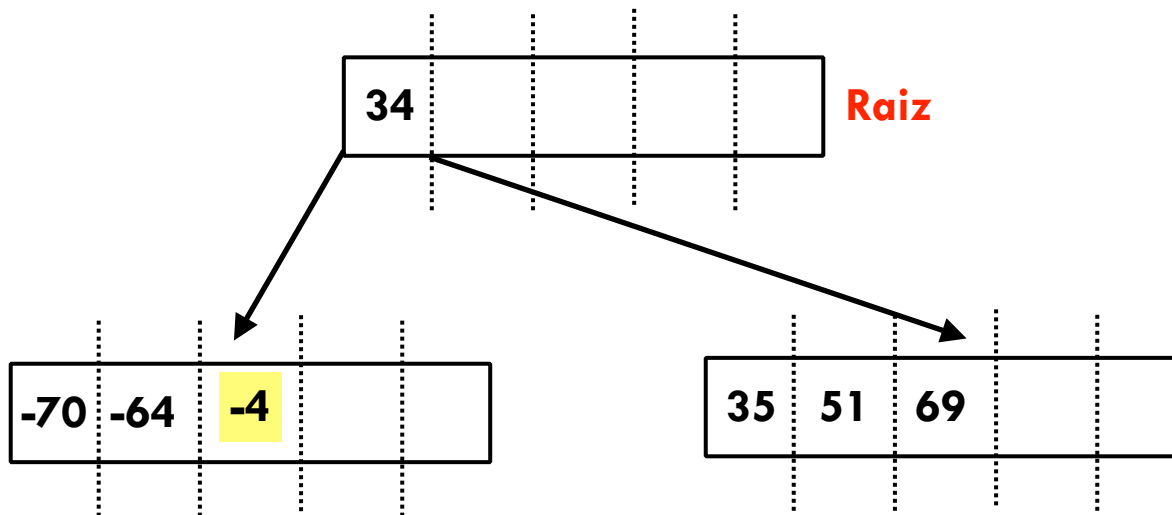
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, \cancel{35}, -4, -97, -21, 41\}$, Ordem = 5



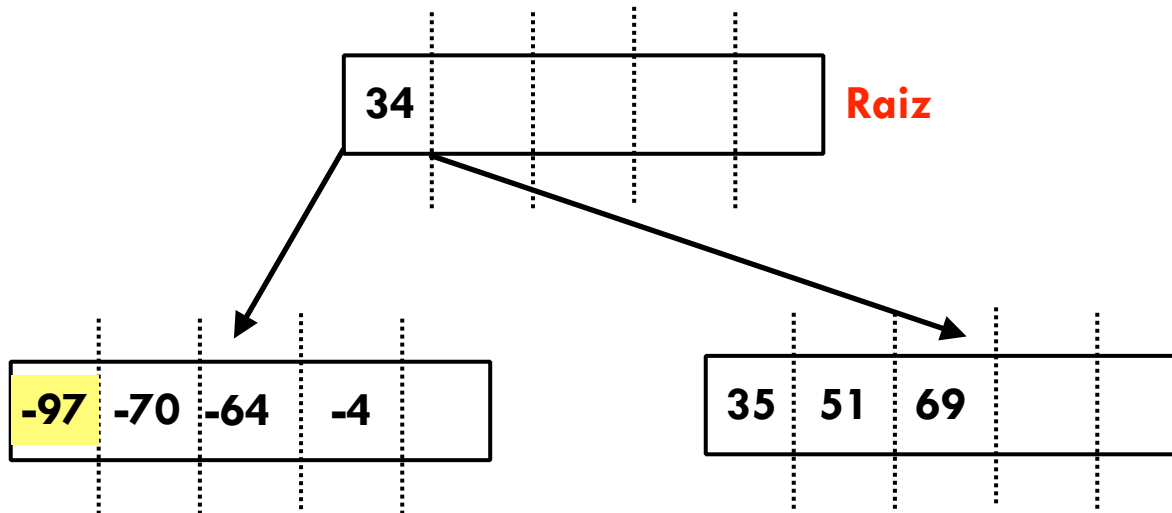
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, 34, \cancel{-70}, \cancel{35}, \cancel{-4}, \cancel{-97}, -21, 41\}$, Ordem = 5



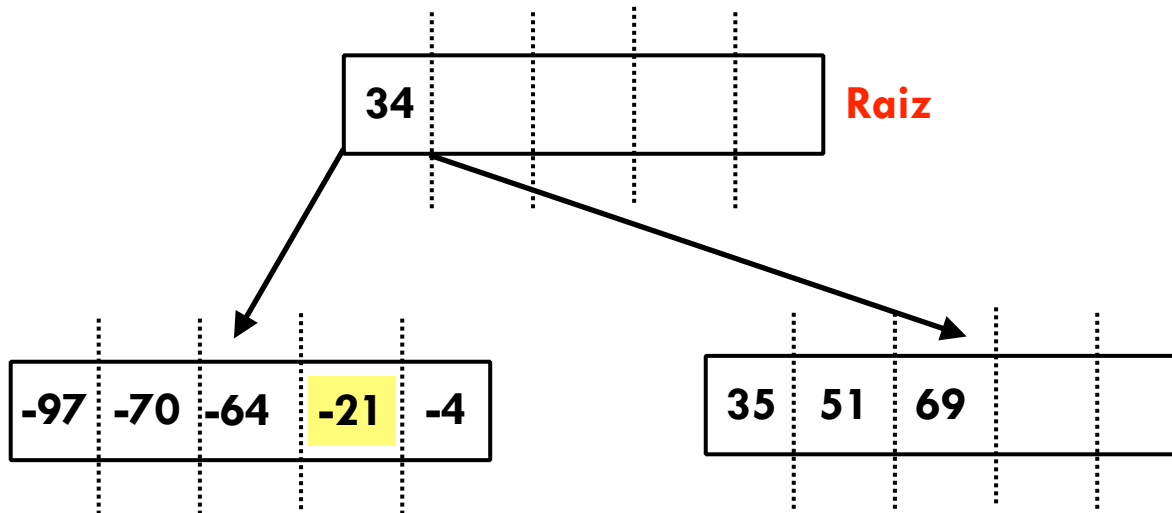
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, \cancel{35}, \cancel{-4}, \cancel{-97}, -21, 41\}$, Ordem = 5



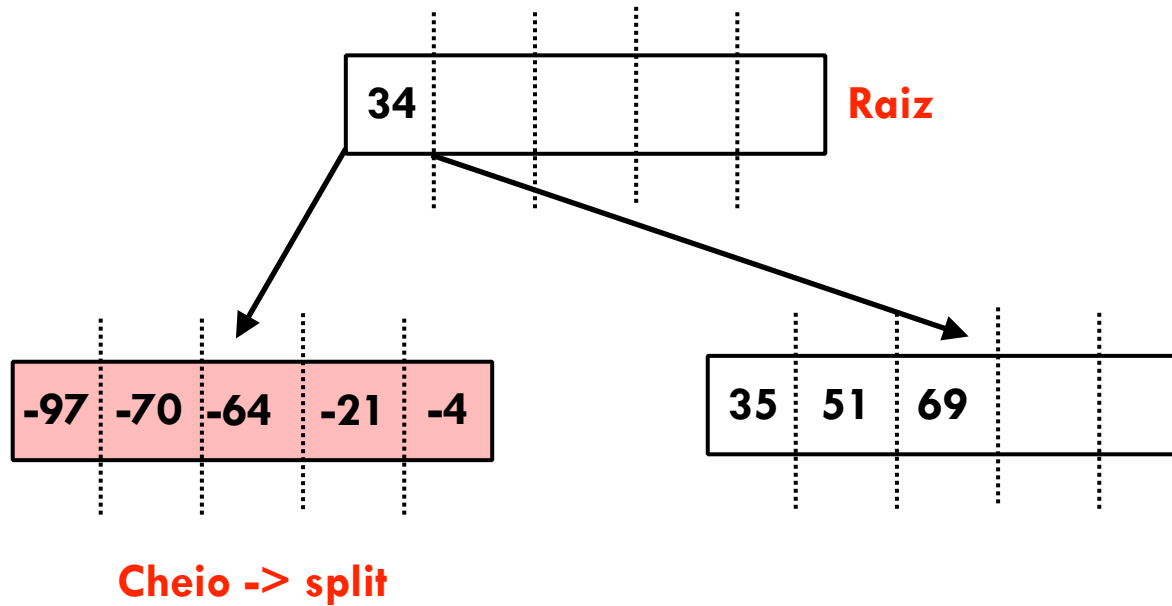
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, \cancel{35}, \cancel{-4}, \cancel{-97}, \cancel{-21}, 41\}$, Ordem = 5



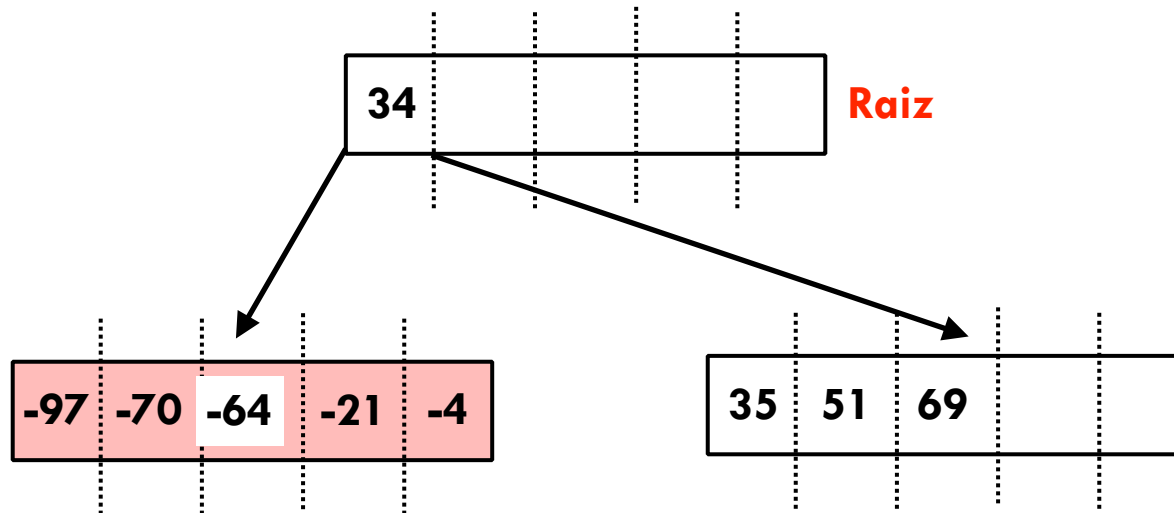
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, 34, \cancel{-70}, \cancel{35}, \cancel{-4}, \cancel{-97}, \cancel{-21}, 41\}$, Ordem = 5



Exemplo

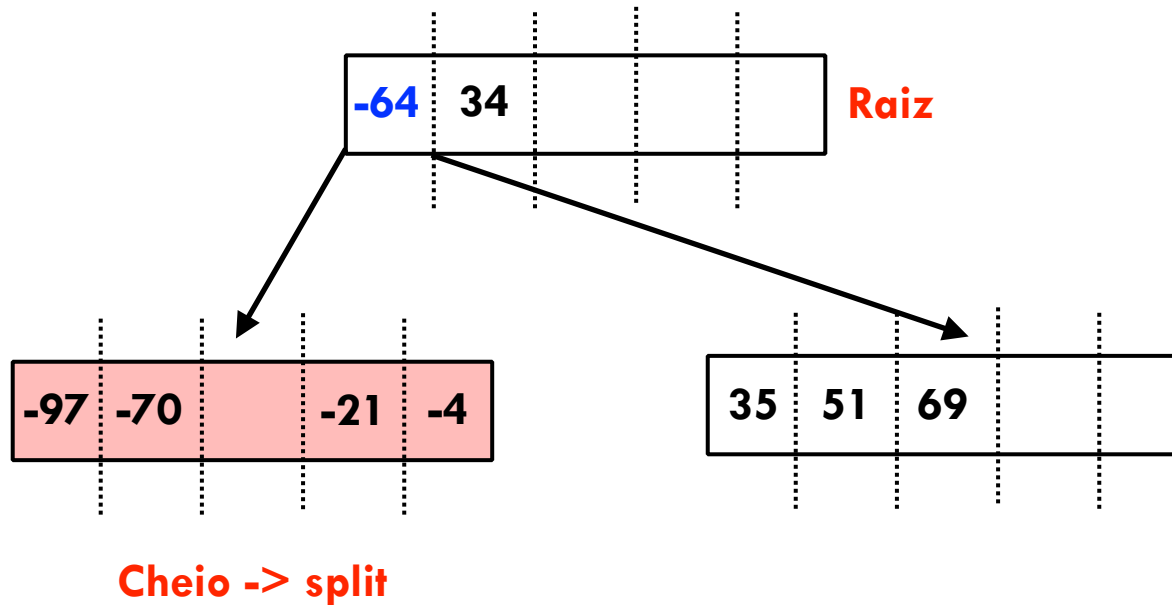
- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, 34, \cancel{-70}, \cancel{35}, \cancel{-4}, \cancel{-97}, \cancel{-21}, 41\}$, Ordem = 5



Cheio -> split

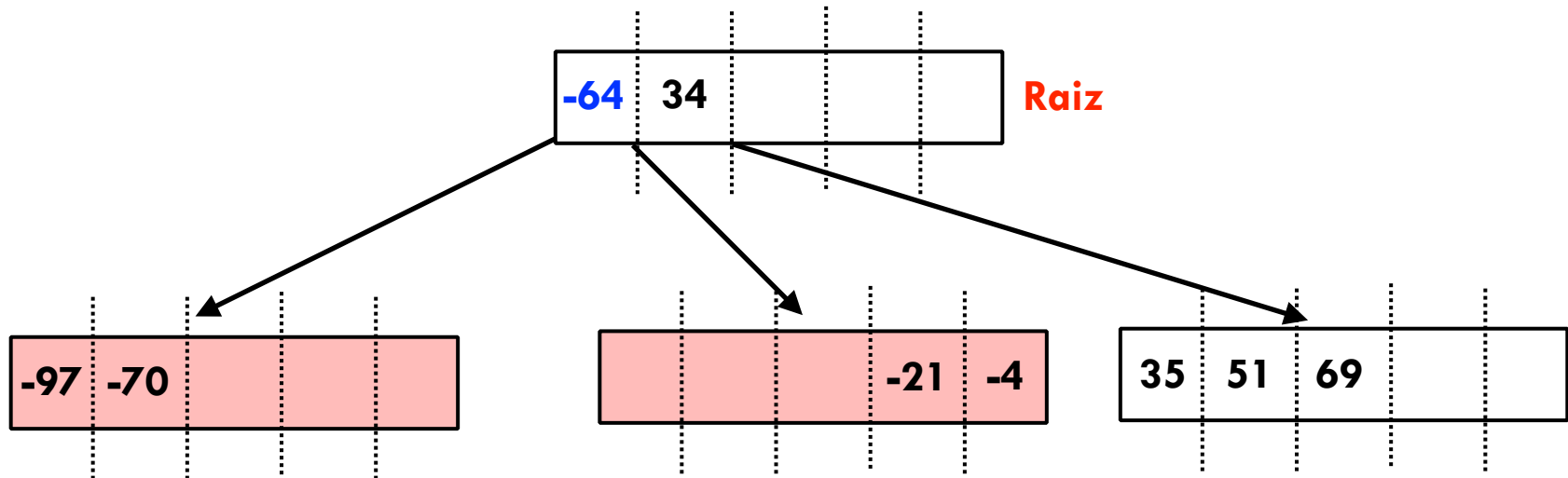
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, \cancel{35}, \cancel{-4}, \cancel{-97}, \cancel{-21}, 41\}$, Ordem = 5



Exemplo

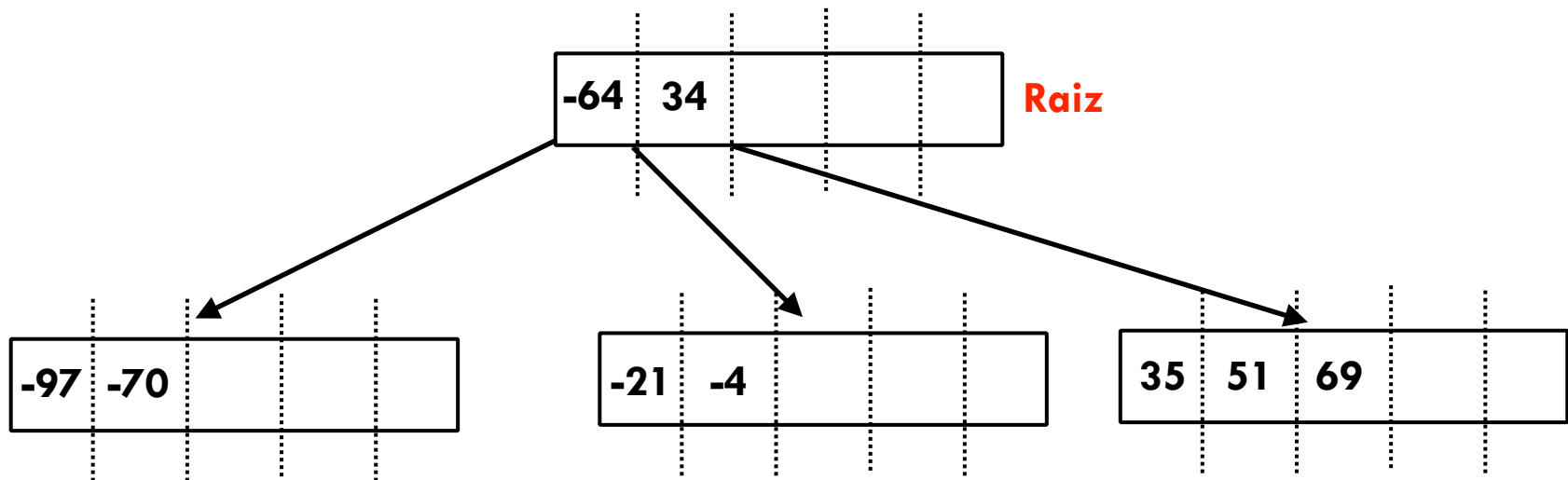
- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, \cancel{35}, \cancel{-4}, \cancel{-97}, \cancel{-21}, 41\}$, Ordem = 5



Cheio -> split

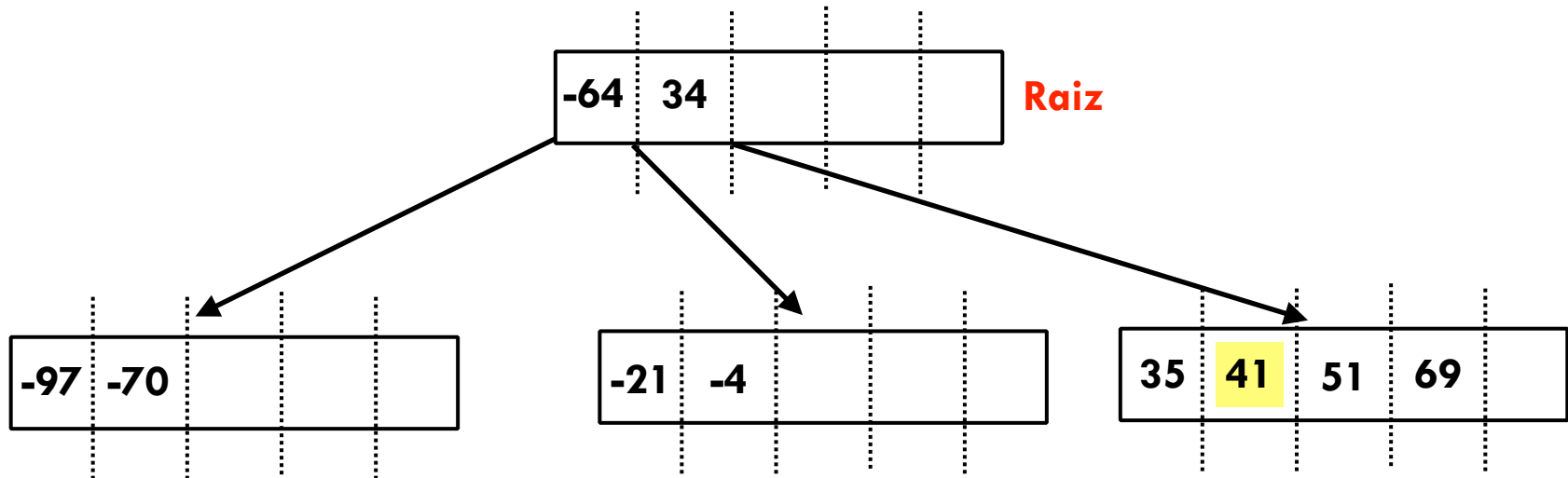
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, \cancel{35}, \cancel{-4}, \cancel{-97}, \cancel{-21}, 41\}$, Ordem = 5



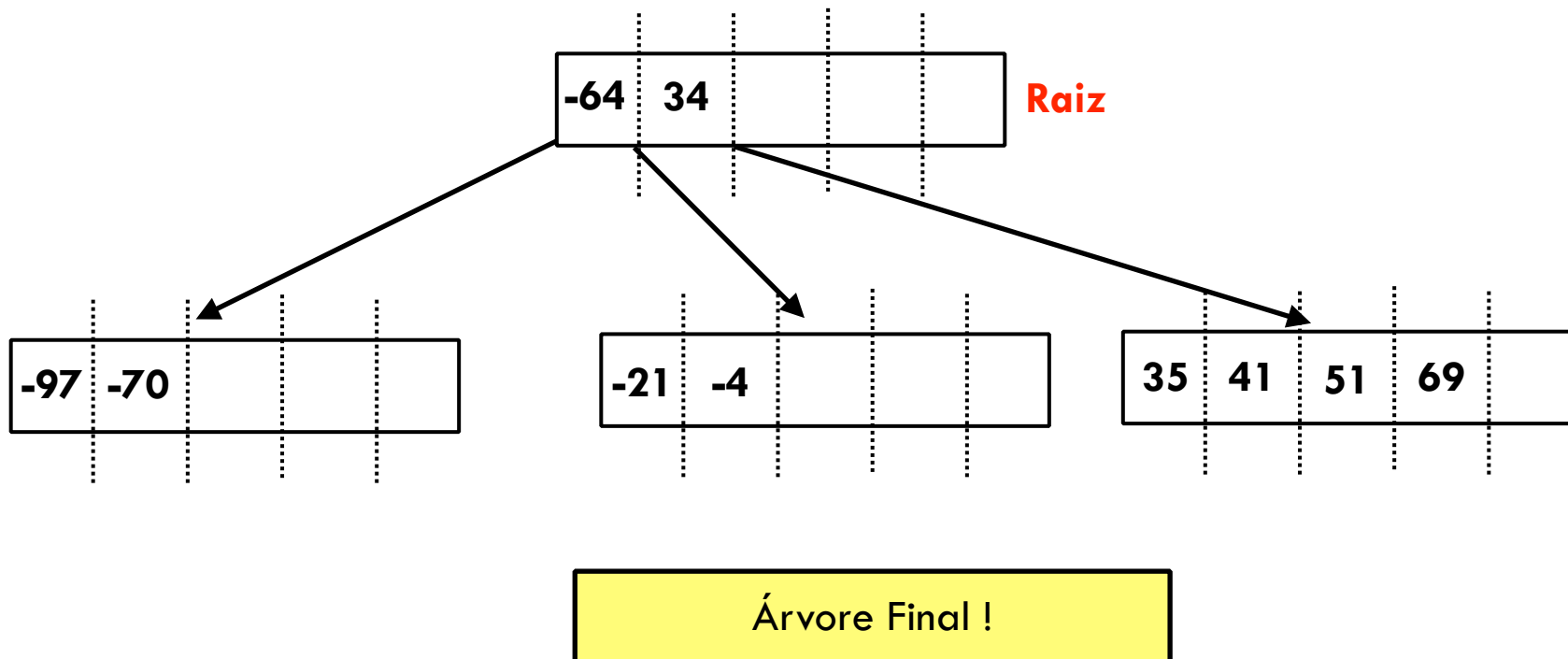
Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, \cancel{35}, \cancel{-4}, \cancel{-97}, \cancel{-21}, \cancel{41}\}$, Ordem = 5



Exemplo

- $S = \{\cancel{51}, \cancel{69}, \cancel{-64}, \cancel{34}, \cancel{-70}, \cancel{35}, \cancel{-4}, \cancel{-97}, \cancel{-21}, \cancel{41}\}$, Ordem = 5



Roteiro

- 1 Introdução
- 2 Definição de B-Tree
- 3 Inserção de elementos
- 4 Exercícios
- 5 Referências

Exercícios

1) Inserir a sequência = $\{-6, -32, 68, 33, -1, 22, 62, -34, 90, -26, -95, 17, 44, 93, 76, -20, -56, 79, -74, -81\}$ em uma B-Tree de ordem 5.

Exercícios

3) Faça o resultado das inserções das seguintes chaves, em ordem:

F, S, Q, K, C, L, H, T, V, W, M, R, N, P, A, B, X, Y, D, Z, E

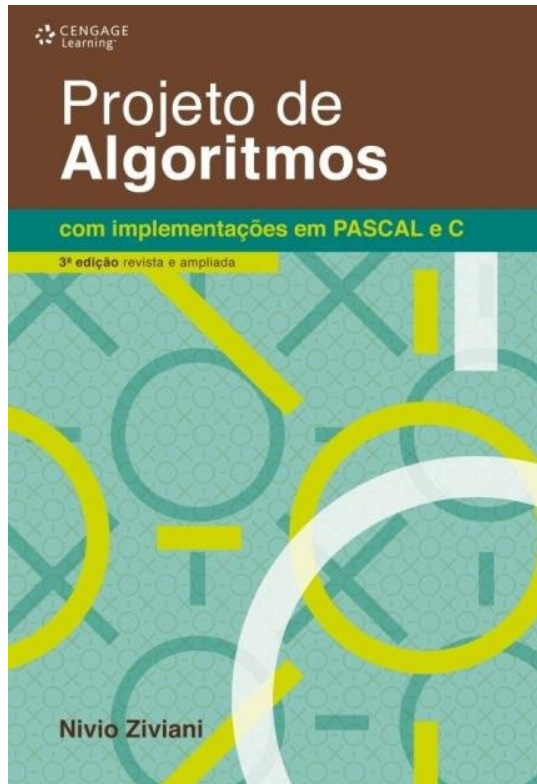
em uma B-Tree vazia com grau mínimo = 2. Desenhe novas árvores apenas quando operações de split sejam necessárias. Desenhe também a árvore final.

Roteiro

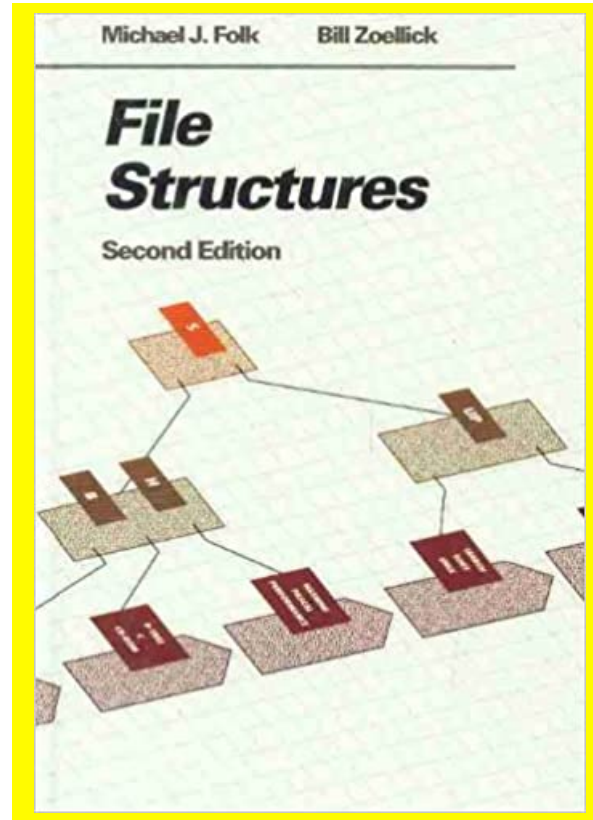


- 1** Introdução
- 2** Definição de B-Tree
- 3** Inserção de elementos
- 4** Exercícios
- 5** Referências

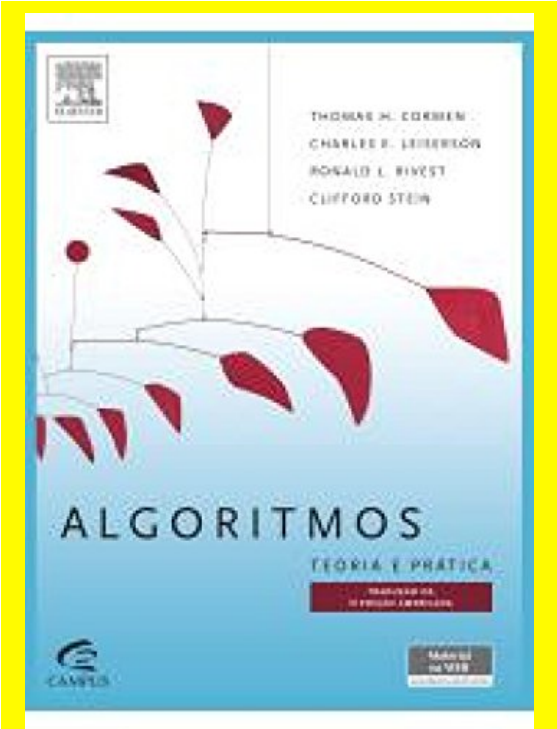
Referências sugeridas



[Ziviani, 2010]



[Folk & Zoellick, 1992]



Perguntas?

Prof. Rafael G. **Mantovani**

rafaelmantovani@utfpr.edu.br