



Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

## Parte V: Árvores B e B+

### Estrutura de Dados II

Sandy Porto  
[sandyporto@gmail.com](mailto:sandyporto@gmail.com)

Universidade Federal de Sergipe

13 de Março de 2017



# Conteúdo

Parte V:

Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

## 1 Introdução

## 2 Árvore B

## 3 Operações

- Busca
- Inserção
- Remoção

## 4 Árvores B+

- Operações



# Introdução

## Relembrando

Parte V:

Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

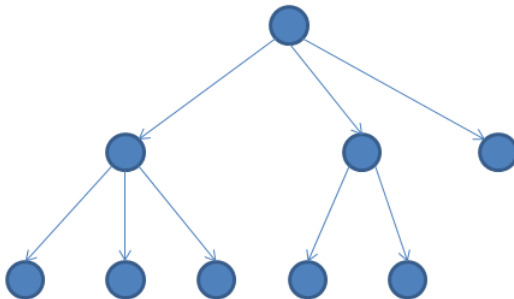
Inserção

Remoção

Árvores B+

Operações

- Árvores. O que significa os termos abaixo?
  - nó? ramo? grau? raiz? folha? nível? altura?





# Conteúdo

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

1 Introdução

2 Árvore B

3 Operações

- Busca
- Inserção
- Remoção

4 Árvores B+

- Operações



# Árvore B

## Parte V:

### Árvores B e B+

Sandy Porto

#### Introdução

#### Árvore B

#### Operações

Busca

Inserção

Remoção

#### Árvores B+

Operações

- São árvores cuja as operações de organização as conservam balanceadas.
  - Organização: Crescimento (Inserção) e Decrescimento (Remoção)
  - Crescimento: *bottom up*
- Cada nó de uma árvore B armazena mais de um registro e contém múltiplos filhos.



- Qual tamanho usar?
  - Proporcional ao tamanho da memória principal!



# Árvore B

Parte V:  
Árvores B e B+  
Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

- Diz-se que uma árvore B é de ordem 2 quando (Bayer e McCreight, 1972):
  - Armazena até  $4(2 * 2)$  registros por nó.
  - Cada nó pode conter até  $5(2 * 2 + 1)$  filhos.



- Árvores B são projetadas para funcionar em memória secundária.
- Suas propriedades permitem que as operações de inserção, remoção e busca de chave primária sejam em  $O(\log n)$ , se  $n$  for o número de registros dentro da árvore.



# Árvore B

## Generalizando

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

- Uma árvore B de **ordem**  $x$ :
  - Exceto a raiz, os nós devem armazenar entre  $x$  e  $2x$  registros.
  - A raiz pode armazenar 1 a  $2x$  registros.
  - Cada nó pode conter até  $2x + 1$  filhos.
  - Seja  $k$  o número de registros mantidos em um nó.
  - Todo nó: ou tem  $k + 1$  filhos ou é folha.
  - TODAS AS FOLHAS ESTÃO NO MESMO NÍVEL.



# Árvore B

Parte V:

Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

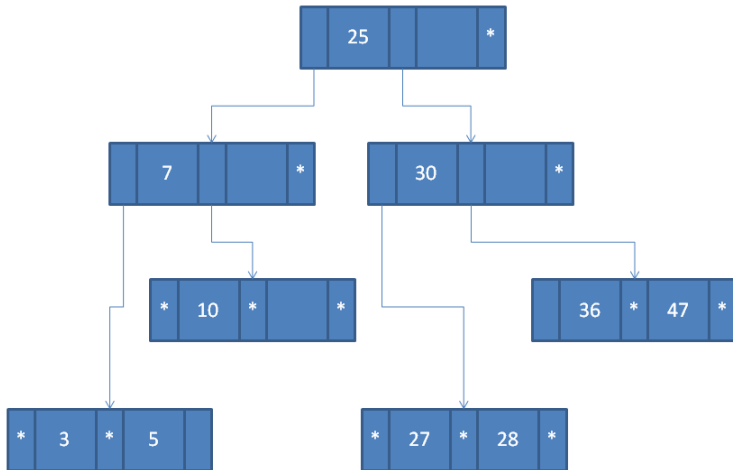
Busca

Inserção

Remoção

Árvores B+

Operações







# Árvore B

## Visão Geral

Parte V:

Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

- Normalmente, as árvores B armazenam grandes quantidades de dados e, portanto, possuem ordens elevadas!
  - Entre 50 e 500.
- Empregada em aplicações como banco de dados e sistemas de arquivos.
- Foi inventada em 1971 por **Rudolf Bayer** e **Edward Miyers McCreight**.
- Funcionam como uma generalização das árvores binárias de busca.



# Árvore B

## Memória Principal x Memória Secundária

### Parte V:

#### Árvores B e B+

Sandy Porto

#### Introdução

#### Árvore B

#### Operações

Busca

Inserção

Remoção

#### Árvores B+

Operações

- Normalmente cabe um nó da árvore B na memória principal.
  - Isso quer dizer que só dá pra carregar um nó por vez.
- A memória secundária guarda todos os nós e os organiza em formato de árvore B.
- Na busca de uma chave:
  - A memória principal carrega a raiz
    - Verifica se a chave está na raiz
    - Caso não, a própria estrutura indica o próximo nó a ser carregado.
  - Refaz o passo anterior até que a chave seja encontrada ou se chegue a uma folha.
- Então, a cada nó carregado desce um nível na árvore. Por isso,  $O(\log n)$ .



# Árvore B

## Struct em C

Parte V:  
Árvores B e B+  
Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

```
#define ordem 2
```

```
typedef struct noArvB {  
    //total de chaves armazenadas  
    int k;  
    //valores de chaves armazenados  
    int chaves[ordem*2];  
    //ponteiros para os filhos  
    struct noArvB filho[(ordem*2)+1];  
}noArvB;
```



# Árvore B

## Outras terminologias

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

- nó = página;
- folhas = páginas externas;
- demais nós = páginas internas;



# Conteúdo

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

1 Introdução

2 Árvore B

3 Operações

- Busca
- Inserção
- Remoção

4 Árvores B+

- Operações



# Operações em Árvores B

## Busca

Parte V:

Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

- Na busca de uma chave:
  - A memória principal carrega a raiz
    - Verifica se a chave está na raiz
    - Caso não, a própria estrutura indica o próximo nó a ser carregado.
  - Refaz o passo anterior até que a chave seja encontrada ou se chegue a uma folha.



# Operações em Árvores B

## Exemplo

Parte V:  
Árvores B e B+  
Sandy Porto

Introdução

Árvore B

Operações

Busca

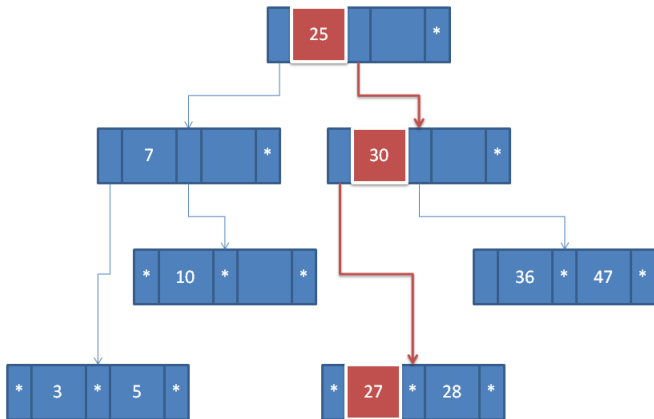
Inserção

Remoção

Árvores B+

Operações

- Busca da chave 27:





# Operações em Árvores B

## Exemplo

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

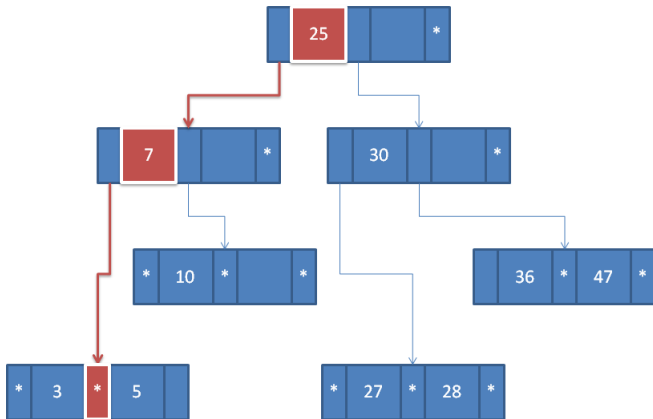
Inserção

Remoção

Árvores B+

Operações

- Busca da chave 4:







# Operações em Árvores B

## Inserção

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

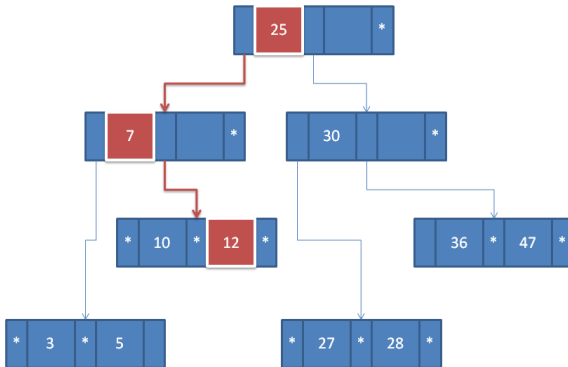
Inserção

Remoção

Árvores B+

Operações

- Primeiro efetua uma busca para achar em qual folha a chave deveria estar.
- Caso 1: há espaço na folha
  - Adiciona a nova chave de forma ordenada.





# Operações em Árvores B

## Inserção

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

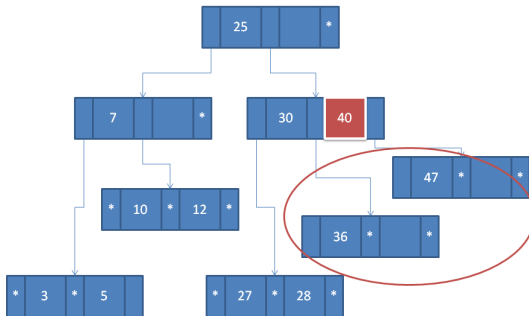
Inserção

Remoção

Árvores B+

Operações

- Caso 2: não há espaço na folha
  - Divide a folha em duas
  - A chave do meio é promovida





# Operações em Árvores B

## Inserção

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

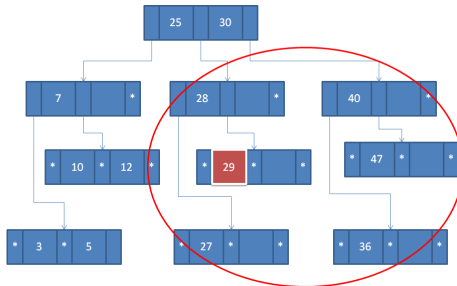
Inserção

Remoção

Árvores B+

Operações

- Caso 3: não há espaço no nó pai
  - Divide a folha em duas
  - A chave do meio é promovida
  - Divide o nó pai em dois
  - Promove a chave do meio do nó pai
  - Chaves maiores são movidas pro novo nó pai
  - Assim como os ponteiros pros nós filhos





# Operações em Árvores B

## Inserção

Parte V:

Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

- Caso 4: o nó pai não tem espaço e é o nó raiz
  - Crescimento *bottom up*
  - Divide o nó pai em dois
  - Nova raiz é criada
  - Chave do meio é promovida
  - Faz as transferências de chaves e ponteiros entre os dois nós pais
  - Ajusta os ponteiros da nova raiz.
- Esse é o pior caso da inserção.
- Somente neste caso a altura da árvore aumenta.



# Operações em Árvores B

## Relembrando

Parte V:

Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

- Propriedades de uma árvore B:
  - Exceto a raiz, os nós devem armazenar entre  $x$  e  $2x$  registros.
  - A raiz pode armazenar 1 a  $2x$  registros.
  - Cada nó pode conter até  $2x + 1$  filhos.
  - Seja  $k$  o número de registros mantidos em um nó.
  - Todo nó: ou tem  $k + 1$  filhos ou é folha.
  - Todas as folhas estão num mesmo nível.



# Operações em Árvores B

## Remoção

Parte V:  
Árvores B e B+  
Sandy Porto

Introdução

Árvore B

Operações

Busca

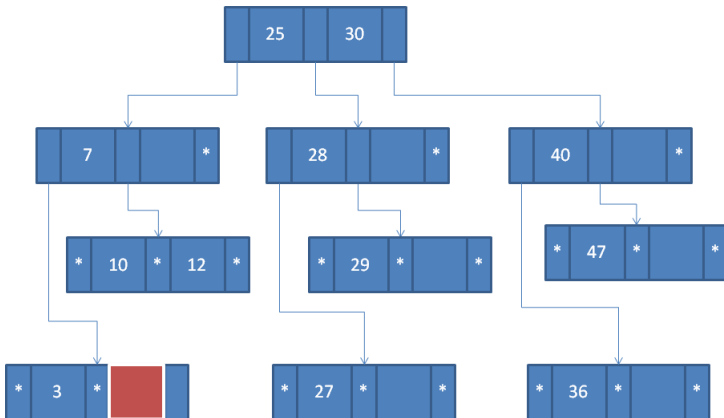
Inserção

Remoção

Árvores B+

Operações

- Caso 1: remoção em folha e a folha continua com  $k \geq \text{ordem}$





# Operações em Árvores B

## Remoção

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

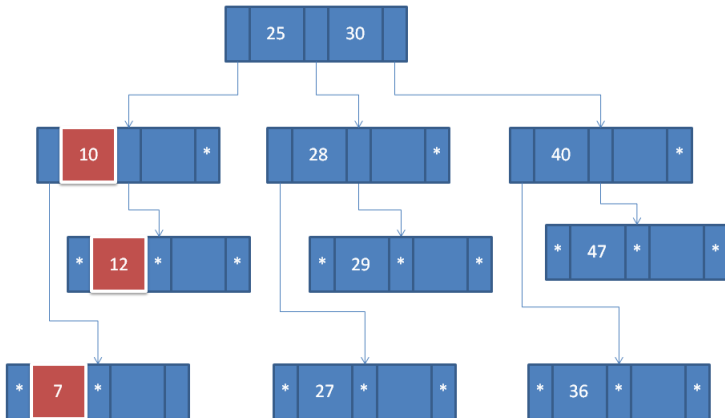
Inserção

Remoção

Árvores B+

Operações

- Caso 2: remoção em folha e a folha fica com  $k < ordem$ .
  - Usa a **Redistribuição**: verifica se a folha esquerda ou direita contém  $k > ordem$  e redistribui a posição das chaves.





# Operações em Árvores B

## Remoção

Parte V:  
Árvores B e B+  
Sandy Porto

Introdução

Árvore B

Operações

Busca

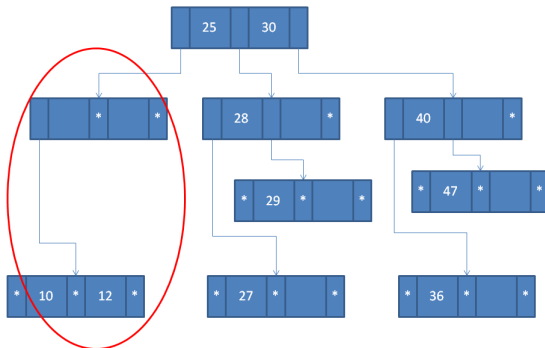
Inserção

Remoção

Árvores B+

Operações

- Caso 3: remoção em folha e a folha fica com  $k < ordem$  e não tem vizinho com  $k > ordem$ 
  - Usa a **Concatenação**: os conteúdos da folha, do vizinho à esquerda e da chave do meio são realocados em uma folha.







# Operações em Árvores B

## Remoção

Parte V:  
Árvores B e B+  
Sandy Porto

Introdução

Árvore B

Operações

Busca

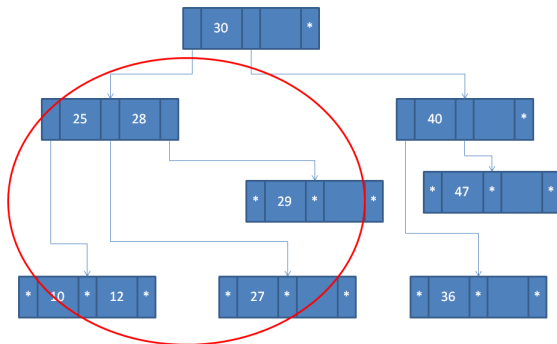
Inserção

Remoção

Árvores B+

Operações

- Caso 3: remoção em folha e a folha fica com  $k < ordem$  e não tem vizinho com  $k > ordem$ 
  - Usa a **Concatenação**: os conteúdos da folha, do vizinho à esquerda e da chave do meio são realocados em uma folha.
  - Se o nó pai ficar com  $k < ordem$  aplica redistribuição ou concatenação no nó pai.





# Operações em Árvores B

## Remoção

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

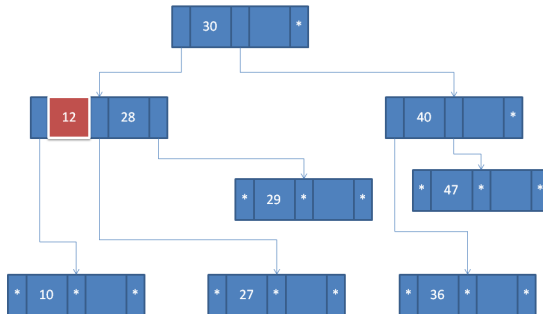
Inserção

Remoção

Árvores B+

Operações

- Caso 4: remoção em nó interno
  - Substitui a chave a ser removida pelo seu antecessor direto, ou seja, pela chave contida na folha mais a direita do ramo esquerdo.
  - Apaga o conteúdo na folha.





# Operações em Árvores B

## Remoção

Parte V:

Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

- Caso 4: remoção em nó interno e a folha do antecessor ficou com  $k < ordem$ 
  - Aplica redistribuição ou concatenação na folha.
- NOTA: somente a concatenação se propaga pela árvore.
- Deve-se tentar a redistribuição primeiro.



# Conteúdo

## Parte V:

### Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

## 1 Introdução

## 2 Árvore B

## 3 Operações

- Busca
- Inserção
- Remoção

## 4 Árvores B+

- Operações



# Introdução

## Parte V:

### Árvores B e B+

Sandy Porto

#### Introdução

#### Árvore B

#### Operações

Busca

Inserção

Remoção

### Árvores B+

Operações

- É uma variante de Árvore B.
- Apresenta algumas vantagens em relação à estrutura original.
- Páginas internas armazenam apenas cópias das chaves primárias.
  - Consequência: algumas chaves aparecerão mais de uma vez.
- Páginas internas funcionam como um índice.
- Páginas externas armazenam o registro por completo.
- Páginas externas são ligadas horizontalmente, e funcionam como uma lista encadeada.
  - Permite acesso sequencial às chaves, sem precisar fazer um percurso em ordem.
  - É a grande vantagem da árvore B+ em relação à árvore B.



# Árvores B+

## Representação

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

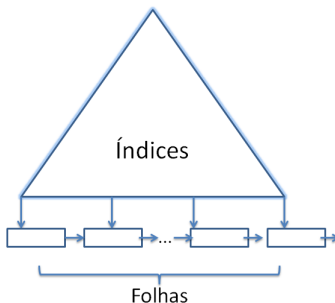
Inserção

Remoção

Árvores B+

Operações

- A árvore B+ é dividida em dois conjuntos: índices e sequencial



- Índices = páginas internas
- Folhas = páginas externas = sequencial



# Árvores B+

## Separador

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

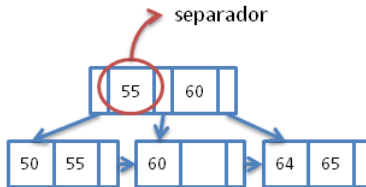
Inserção

Remoção

Árvores B+

Operações

- A “chave do meio” é chamada **separador** na árvore B+



- Geralmente é uma cópia da chave de maior valor do filho à esquerda.



# Árvores B+

## Características

Parte V:

Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

- Todas as chaves são mantidas em folhas.
- Algumas chaves são repetidas em nós internos formando um índice.
- As chaves em índices são chamadas *separadores*.
- As folhas são ligadas, como uma lista ligada.
- Oferece um caminho sequencial entre as folhas.
- Vantagens:
  - Mantém a eficiência de busca e da inserção da árvore B.
  - Eficiência da localização do próximo registro passa de  $O(\log_{ordem} N)$  para  $O(1)$ .





# Árvores B+

## Exemplo

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

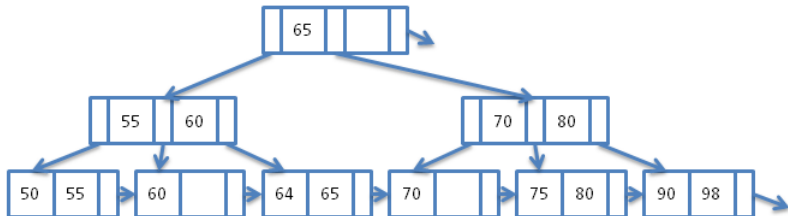
Busca

Inserção

Remoção

Árvores B+

Operações





# Operações em Árvores B+

Parte V:

Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

- **Busca**

- Semelhante à busca em Árvore B.
- Mas a busca sempre atinge uma folha
- Mas se necessário buscar um novo registro, segue o sequencial.

- **Inserção**

- Sempre insere na folha.
- Se ocorrer divisão da folha, a chave do meio é copiada para o índice e se torna um separador.
- O índice segue as regras e propriedades das árvores B.



# Operações em Árvores B+

Parte V:  
Árvores B e B+

Sandy Porto

Introdução

Árvore B

Operações

Busca

Inserção

Remoção

Árvores B+

Operações

## ● Remoção

- Semelhante à remoção em Árvore B.
- Se necessário usar a redistribuição, *altera* o separador.
- Se necessário usar a concatenação, *remove* o separador.
- A remoção das chaves ocorre somente nas folhas, os separadores só são modificados durante redistribuição e concatenação.