

# Estrutura de Dados 2

## Árvores B



# Introdução

# Árvores B

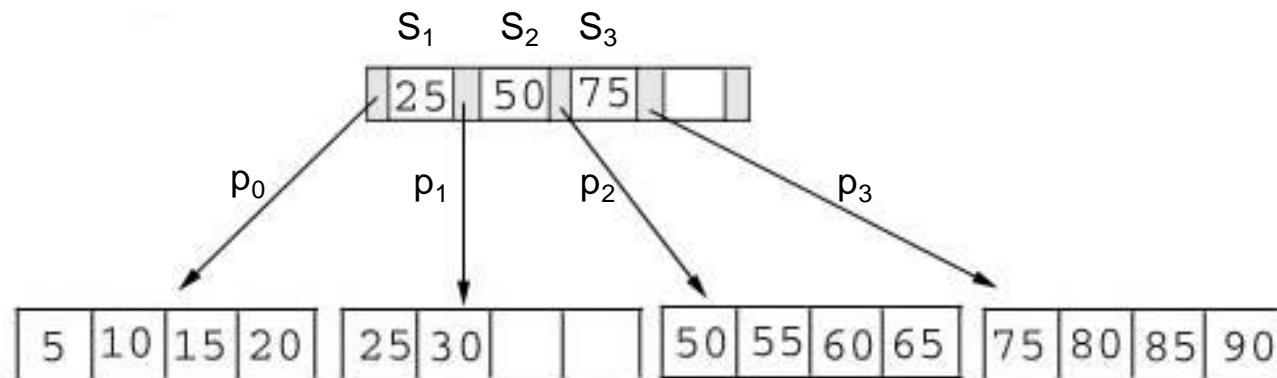
- Definição
  - Seja  $d$  um número natural. Uma *Árvore B de ordem  $d$*  é uma árvore **ordenada** que é vazia ou que satisfaz as seguintes condições:
    - A raiz é uma folha ou tem no mínimo dois filhos
    - Cada nó diferente da raiz e das folhas possui, no mínimo,  $d+1$  filhos
    - Cada nó tem, no máximo,  $2d+1$  filhos
    - Todas as folhas estão no mesmo nível

# Árvores B

- Um nó de uma Árvore B é chamado de *página*
  - Se uma página  $P$  não-folha possui  $m$  chaves, então  $P$  possui  $m+1$  filhos
  - A raiz possui entre 1 e  $2d$  chaves
  - Cada página diferente da raiz possui entre  $d$  e  $2d$  chaves
  - Em cada página  $P$ , as chaves estão ordenadas
  - $P$  contém  $m+1$  ponteiros  $p_0, p_1, \dots, p_{(m+1)}$  apontando para seus filhos
    - Nas folhas, estes ponteiros indicam NULL

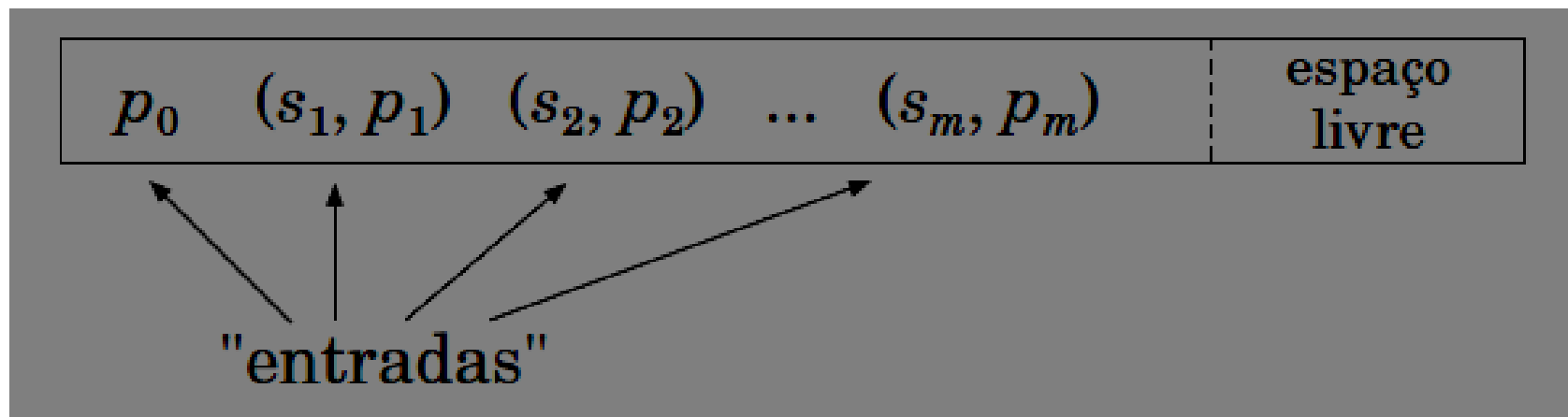
# Árvores B

- Exemplo:  $d = 2$



# Árvores B

- Estrutura de uma página
  - Para qualquer chave  $y$ , pertencente à página apontada por  $p_0$ ,  $y < S_1$
  - Para qualquer chave  $y$ , pertencente à página apontada por  $p_k$ ,  $1 \leq k \leq m-1$ ,  $S_k < y < S_{k+1}$
  - Para qualquer chave  $y$ , pertencente à página apontada por  $p_m$ ,  $y > S_m$

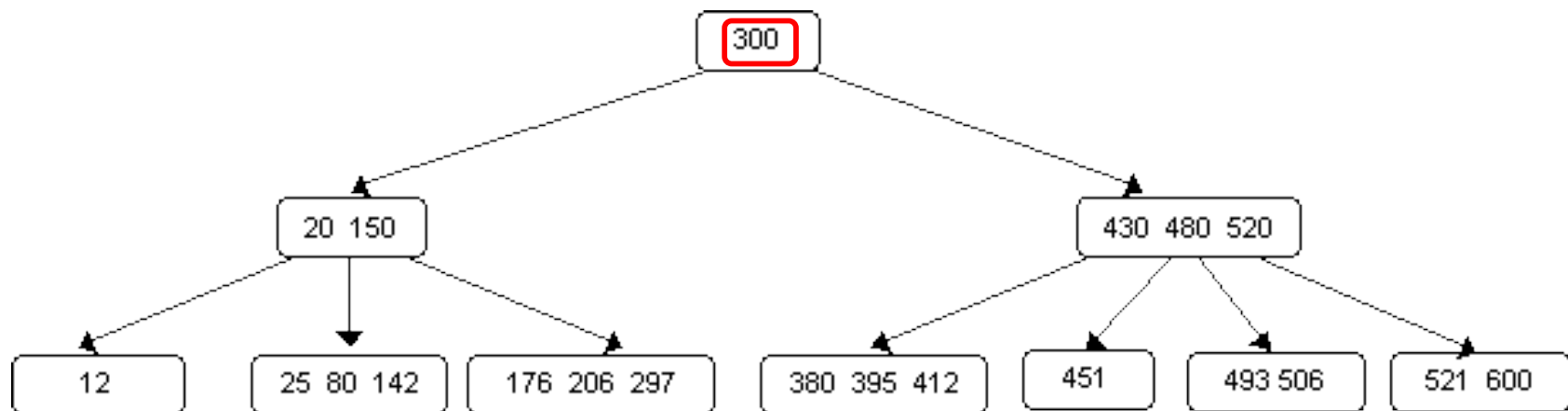


# Busca

- Seja  $s$  a chave procurada
- A busca inicia-se na primeira chave da raiz  $s_1$ :
- Compara-se  $s$  com o valor de  $s_i$ 
  - Se  $s = s_i$ , a busca termina
  - Se  $s < s_i$ , iniciar o procedimento de busca na página apontada por  $p_{i-1}$ 
    - Se  $p_{i-1} = \text{NULL}$  “ELEMENTO NÃO ENCONTRADO”
  - Se  $s > s_i$  e  $s_{i+1} \neq \text{NULL}$ , realizar a busca em  $s_{i+1}$
  - Se  $s > s_i$  e  $s_{i+1} = \text{NULL}$ , iniciar o procedimento de busca na página apontada por  $p_i$ 
    - Se  $p_i = \text{NULL}$  “ELEMENTO NÃO ENCONTRADO”

# Busca

- Exemplo: buscar o elemento 451

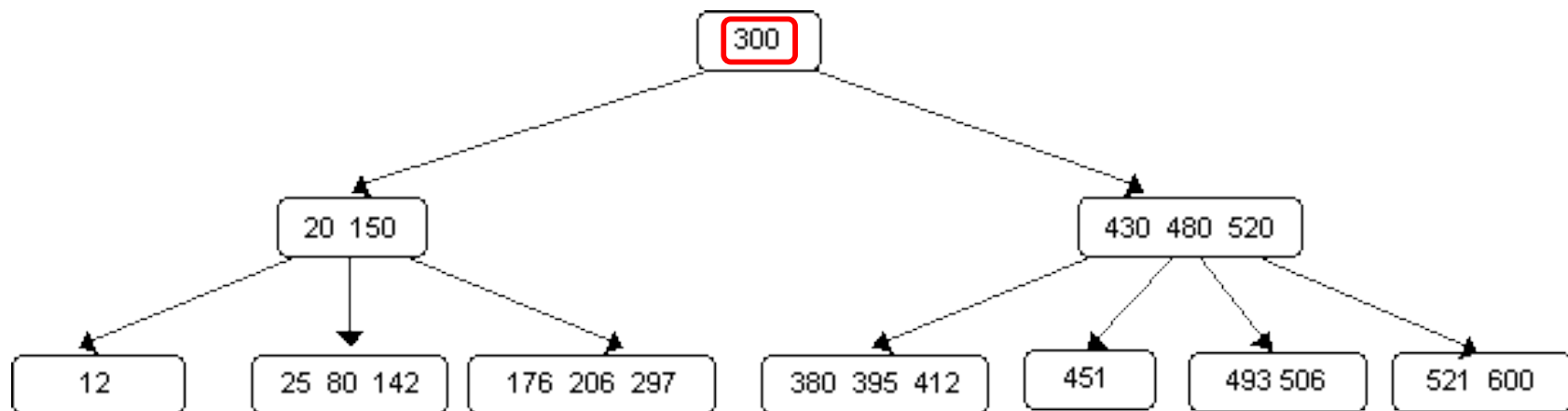


**Elemento encontrado.**



# Busca

- Exemplo: buscar o elemento 234



**A chave 297 é maior que a chave buscada. O ponteiro entre 206 e 297 é NULL.**

**ELEMENTO NÃO ENCONTRADO**

# Inserção

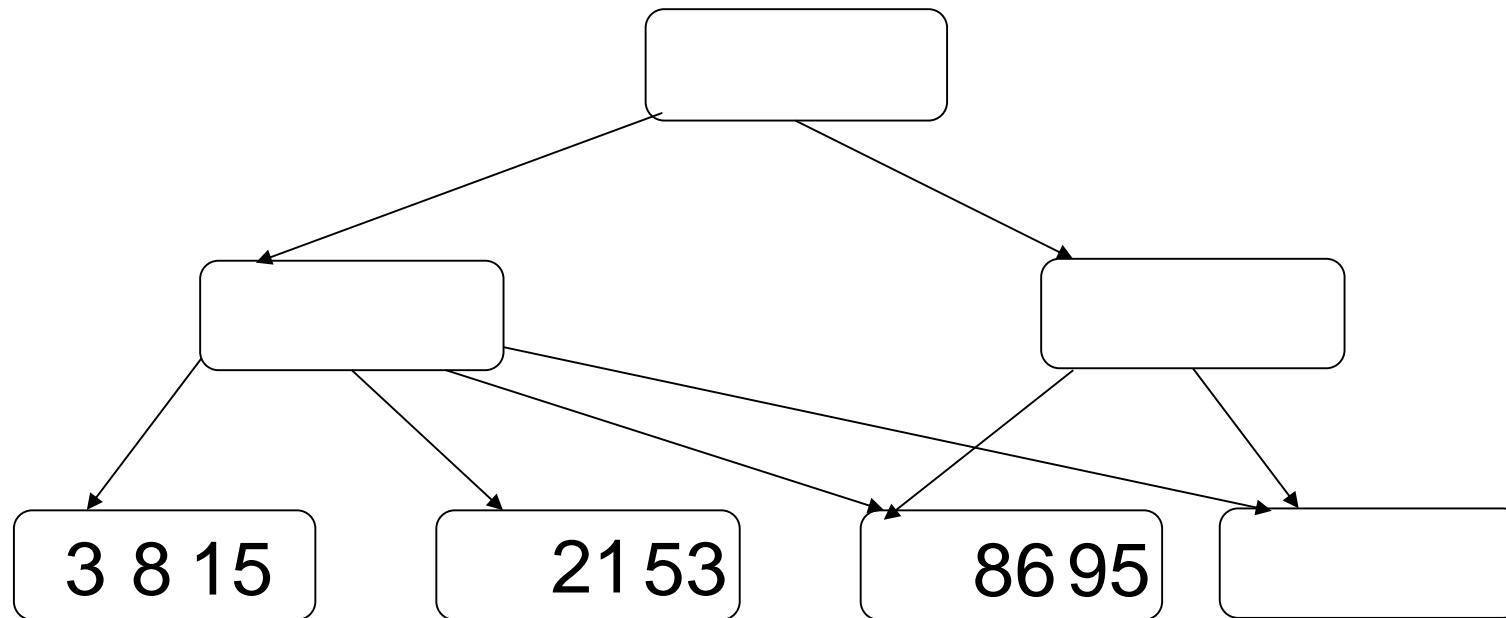
- Seja  $s$  a chave a ser inserida
- Utilizando o processo de busca, verifica-se:
  1. Se a chave já existe
  2. O local onde a chave deverá ser inserida, caso ainda não exista
- Insere-se a chave no ponto retornado no passo anterior
- Se o número de chaves da página onde  $s$  foi inserida for maior que  $2d$ , realizar **cisão** (*split*) da página

# Cisão

- Seja  $p$  a página que sofrerá cisão e  $m$  o número de elementos em  $p$  ( $m = 2d+1$ )
  - Cria-se uma nova página vazia  $p'$
  - Move-se os elementos  $p[d+1]$  a  $p[2d+1]$  para  $p'$
  - Move-se o elemento  $p[d]$  para a página-pai de  $p$ 
    - Após a inserção no pai, acertar os ponteiros para as páginas  $p$  e  $p'$
- A cisão pode se propagar no pai de  $p$  até a raiz, quando uma nova raiz será criada.

# Inserção

- Exemplo ( $d = 1$ )



**Página possui  $2d+1$  elementos. Realizar cisão**

# Exclusão

- Seja  $s$  a chave a ser removida
  - Busca-se  $s$  na árvore. Caso não seja encontrada, nada a fazer
  - Se  $s$  for encontrada numa página-folha, remova-a
  - Se  $s$  for encontrada numa página não-folha, remova-a e coloque no seu lugar a chave  $x$  imediatamente superior a  $s$  ( $x$  estará numa folha)

**A exclusão sempre ocorrerá em uma folha**

## Exclusão (2)

- Caso a folha onde tenha ocorrido a exclusão tenha ficado desbalanceada ( $m < d$ ), utilizar ou o procedimento de redistribuição ou de concatenação

# Redistribuição

- Se a folha  $f$  possui uma irmã  $f'$  com  $k$  elementos, tal que  $k+m > 2d$ 
  - Troca-se o primeiro (menor) elemento de  $f'$  pelo elemento do pai que aponta para  $f$
  - Este elemento do pai é inserido em  $f$ , tornando-o novamente balanceado
  - Não é propagável, por isso é o método preferível

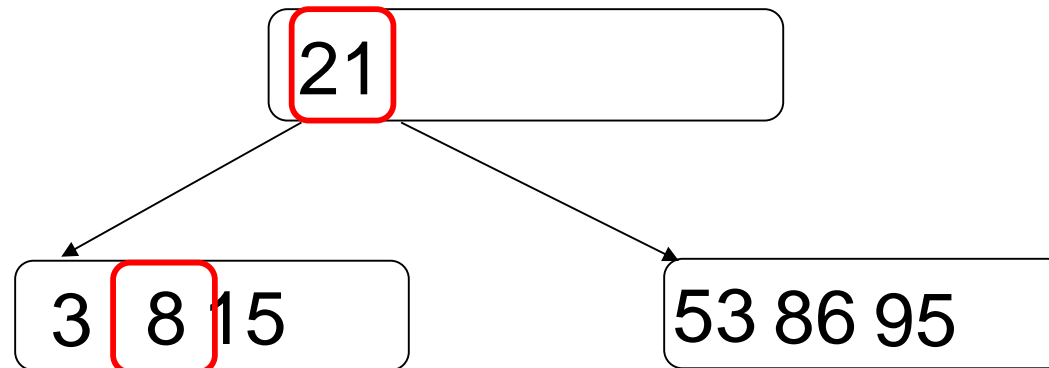
# Concatenação

- Se não for possível aplicar a redistribuição e  $f$  possuir uma irmã  $f'$  com  $k$  chaves tal que  $m+k \leq 2d$ 
  - Mover todas as chaves de  $f'$  para  $f$
  - Acertar os ponteiros no pai de  $f$ , que pode ficar desbalanceado
  - A concatenação é propagável
    - Se chegar até a raiz, a árvore terá diminuído sua altura



# Exclusão

- Exemplo ( $d = 2$ )



**Página desbalanceada. Utilizar redistribuição.**  
**Página desbalanceada. Utilizar concatenação.**

# Discussão

- Qual a complexidade, no pior caso, do algoritmo de busca em uma Árvore B?

$$O(\underbrace{2^d}_{\text{Resultado de uma busca linear}} \times h)$$

Resultado de uma busca linear

- É possível melhorá-la?
- E se utilizássemos uma busca binária? Como ficaria a complexidade?

$$O(\log 2^d \times h)$$

# Exercícios

- Qual o número máximo de páginas que uma Árvore B de altura  $h$  e ordem  $d$  pode ter?
- E qual o número máximo de elementos ?

Fórmula para os termos de uma PG:

$$S_n = a_1 (q^n - 1) / (q - 1)$$

- Determinar a expressão da altura máxima de uma Árvore B de ordem  $d$ .
- Implementar em Java uma Árvore B de ordem 4 completa, com as operações de inclusão, exclusão e busca.