

MATA54 - Estruturas de Dados e Algoritmos II

Árvores B e Árvores B+

Flávio Assis

Versão gerada a partir de slides do Prof. George Lima

IC - Instituto de Computação

Salvador, setembro de 2021

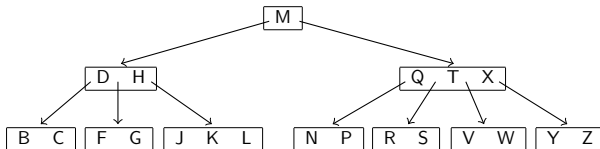
Árvores B

Motivações: desempenho e grande volume de dados

Árvore B

Uma árvore k -ária balanceada

- ▶ Vários registros e ponteiros por nó
- ▶ Cresce das folhas para a raiz
- ▶ Balanceamento é consequência do processo de inserção de chaves
- ▶ Balanceamento: os caminhos da raiz às folhas possuem o mesmo comprimento



Usaremos a definição de árvores B apresentada em [Cormen et al. 2009].

Árvore B: Propriedades [Cormen *et al.* 2009]

Uma árvore B é uma árvore enraizada com as seguintes propriedades:

1. Cada nó x tem os seguintes atributos:
 - 1.1 $x.n$, o número de chaves armazenadas no nó x
 - 1.2 as $x.n$ chaves, $x.key_1, x.key_2, \dots, x.key_n$, armazenadas em ordem não decrescente, de forma que $x.key_1 \leq x.key_2 \leq \dots \leq x.key_n$
 - 1.3 $x.leaf$, um valor *booleano* que é TRUE se x é uma folha e FALSE se x é um nó interno
2. Cada nó x contém $x.n + 1$ apontadores $x.c_1, x.c_2, \dots, x.c_{n+1}$. Se x é nó interno, esses apontadores apontam para seus nós filhos. Se é nó folha, esses atributos c_i são não definidos

c_1	k_1	c_2	k_2	\dots	$c_{x.n}$	$k_{x.n}$	$c_{x.n+1}$
-------	-------	-------	-------	---------	-----------	-----------	-------------

Árvore B: Propriedades (cont)

3. As chaves $x.key_i$ separam as faixas de chaves armazenadas em cada subárvore: se a chave k_i é uma chave armazenada na subárvore com raiz $x.c_i$ então

$$k_1 \leq x.key_1 \leq k_2 \leq x.key_2 \leq \dots \leq x.key_{x.n} \leq k_{x.n+1}$$

4. Todas as folhas possuem a mesma profundidade, que é igual à altura h da árvore
5. Nós possuem um limite inferior e um limite superior no número de chaves que podem conter. Estes limites são expressos em termos de um número inteiro fixo $t \geq 2$, chamado o **grau mínimo** da árvore B:
- 5.1 todo nó, com exceção da raiz, tem que ter pelo menos $t - 1$ chaves. Cada nó interno tem que ter então pelo menos t nós filhos. Se a árvore é não vazia a raiz tem que ter pelo menos uma chave.
- 5.2 Todo nó pode conter no máximo $2t - 1$ chaves. Por isso, um nó interno pode ter no máximo $2t$ nós filhos.

Ilustração de Inserção em Árvore B

Sequência de chaves a serem inseridas em uma árvore B com $t = 2$:

10, 75, 62, 69

Número mínimo de chaves por nó (com exceção da raiz): $t - 1$: 1

Número máximo de chaves por nó: $2t - 1$: 3

Inserção da chave **10** em uma árvore B vazia:

raiz \rightarrow

λ	10	λ
-----------	----	-----------

Ilustração de Inserção em Árvore B

Sequência de chaves a serem inseridas em uma árvore B com $t = 2$:

10, 75, 62, 69

Número mínimo de chaves por nó (com exceção da raiz): $t - 1$: 1

Número máximo de chaves por nó: $2t - 1$: 3

Inserção da chave **10** em uma árvore B vazia:

raiz \rightarrow

λ	10	λ
-----------	----	-----------

Inserção das chaves **75** e **62**:

Ilustração de Inserção em Árvore B

Sequência de chaves a serem inseridas em uma árvore B com $t = 2$:

10, 75, 62, 69

Número mínimo de chaves por nó (com exceção da raiz): $t - 1$: 1

Número máximo de chaves por nó: $2t - 1$: 3

Inserção da chave **10** em uma árvore B vazia:

raiz \rightarrow

λ	10	λ
-----------	----	-----------

Inserção das chaves **75** e **62**:

raiz \rightarrow

λ	10	λ	62	λ	75	λ
-----------	----	-----------	----	-----------	----	-----------

Ilustração de inserção em árvore B

raiz →

λ	10	λ	62	λ	75	λ
-----------	----	-----------	----	-----------	----	-----------

Inserção da chave **69**:

Ilustração de inserção em árvore B

raiz →

λ	10	λ	62	λ	75	λ
-----------	----	-----------	----	-----------	----	-----------

Inserção da chave **69**: **Quebra do nó!**

Ilustração de inserção em árvore B

raiz →

λ	10	λ	62	λ	75	λ
-----------	----	-----------	----	-----------	----	-----------

Inserção da chave **69**: **Quebra do nó!**

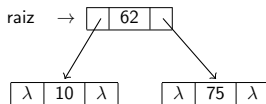
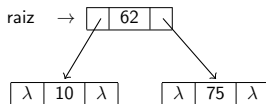


Ilustração de inserção em árvore B

raiz →

λ	10	λ	62	λ	75	λ
-----------	----	-----------	----	-----------	----	-----------

Inserção da chave **69**: **Quebra do nó!**



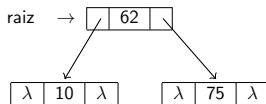
Inserção da chave 69 na árvore:

Ilustração de inserção em árvore B

raiz →

λ	10	λ	62	λ	75	λ
---	----	---	----	---	----	---

Inserção da chave **69**: **Quebra do nó!**



Inserção da chave 69 na árvore:

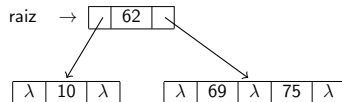


Ilustração de Inserção em Árvore B

Inserção da chave **80**:

Ilustração de Inserção em Árvore B

Inserção da chave **80**:

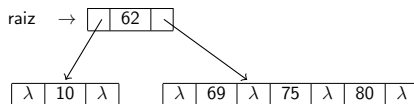
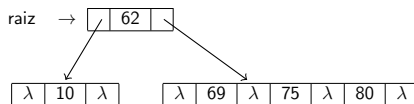


Ilustração de Inserção em Árvore B

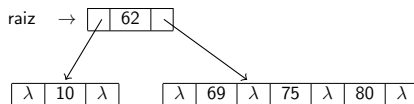
Inserção da chave **80**:



Inserção da chave **90**:

Ilustração de Inserção em Árvore B

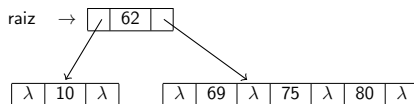
Inserção da chave **80**:



Inserção da chave **90**: **Quebra de nó!**

Ilustração de Inserção em Árvore B

Inserção da chave **80**:



Inserção da chave **90**: **Quebra de nó!**

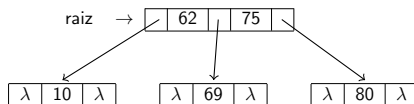
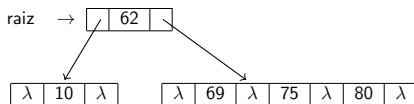
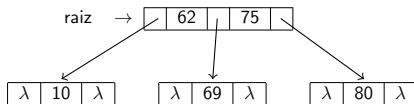


Ilustração de Inserção em Árvore B

Inserção da chave **80**:



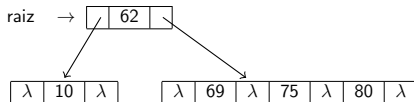
Inserção da chave **90**: **Quebra de nó!**



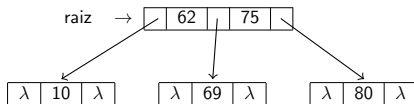
Inserção do 90 na árvore:

Ilustração de Inserção em Árvore B

Inserção da chave **80**:



Inserção da chave **90**: **Quebra de nó!**



Inserção do 90 na árvore:

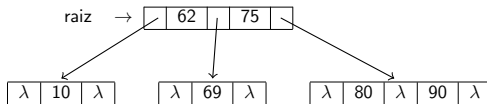


Ilustração de inserção em árvore B

Inserção das chaves **20**, **30** e **100**:

Ilustração de inserção em árvore B

Inserção das chaves **20**, **30** e **100**:

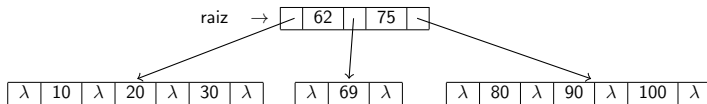
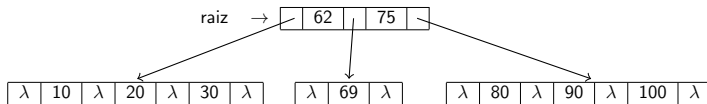


Ilustração de inserção em árvore B

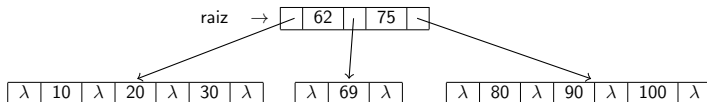
Inserção das chaves **20**, **30** e **100**:



Inserção da chave **25**:

Ilustração de inserção em árvore B

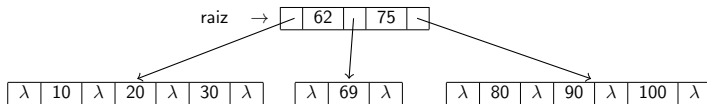
Inserção das chaves **20**, **30** e **100**:



Inserção da chave **25**: **Quebra de nó!**

Ilustração de inserção em árvore B

Inserção das chaves **20**, **30** e **100**:



Inserção da chave **25**: **Quebra de nó!**

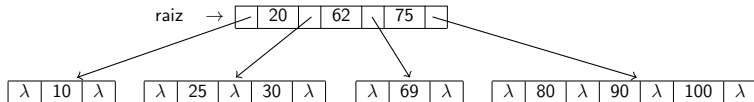


Ilustração de Inserção em Árvore B

A quebra de nó faz com que uma chave suba na árvore. Como evitar que uma inserção gere quebras sucessivas no caminho de volta para a raiz?

Ilustração de Inserção em Árvore B

A quebra de nó faz com que uma chave suba na árvore. Como evitar que uma inserção gere quebras sucessivas no caminho de volta para a raiz?

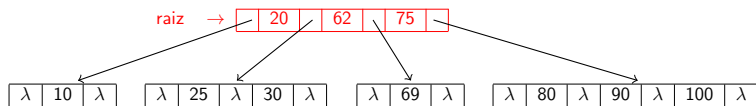
No processo de busca pelo nó onde a chave será inserida, quebra-se sempre um nó que estiver com capacidade máxima!

Ilustração de Inserção em Árvore B

A quebra de nó faz com que uma chave suba na árvore. Como evitar que uma inserção gere quebras sucessivas no caminho de volta para a raiz?

No processo de busca pelo nó onde a chave será inserida, quebra-se sempre um nó que estiver com capacidade máxima!

Inserção da chave **85**:



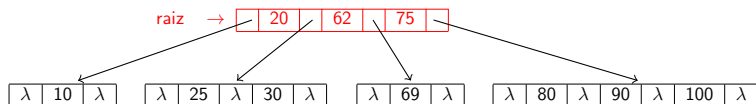
Nó raiz com capacidade máxima!

Ilustração de Inserção em Árvore B

A quebra de nó faz com que uma chave suba na árvore. Como evitar que uma inserção gere quebras sucessivas no caminho de volta para a raiz?

No processo de busca pelo nó onde a chave será inserida, quebra-se sempre um nó que estiver com capacidade máxima!

Inserção da chave **85**:



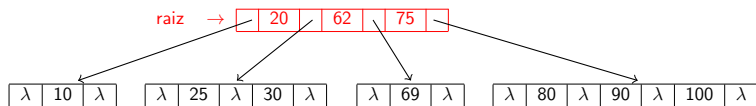
Nó raiz com capacidade máxima! **Quebra do nó!**

Ilustração de Inserção em Árvore B

A quebra de nó faz com que uma chave suba na árvore. Como evitar que uma inserção gere quebras sucessivas no caminho de volta para a raiz?

No processo de busca pelo nó onde a chave será inserida, quebra-se sempre um nó que estiver com capacidade máxima!

Inserção da chave **85**:



Nó raiz com capacidade máxima! **Quebra do nó!**

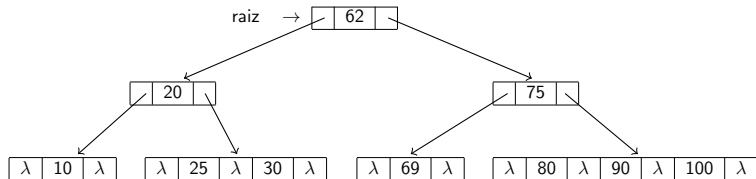


Ilustração de Inserção em Árvore B

Nó folha com capacidade máxima!

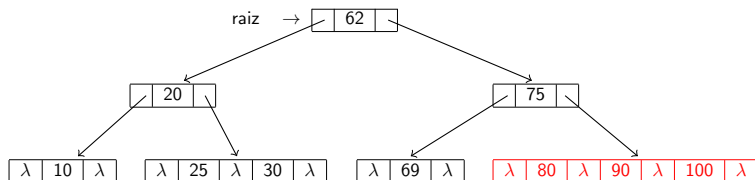
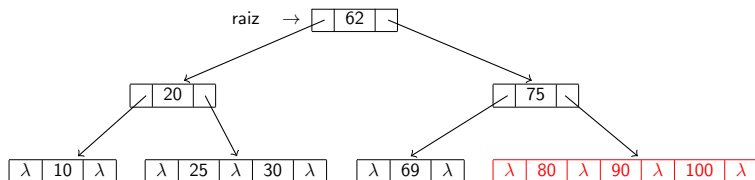


Ilustração de Inserção em Árvore B

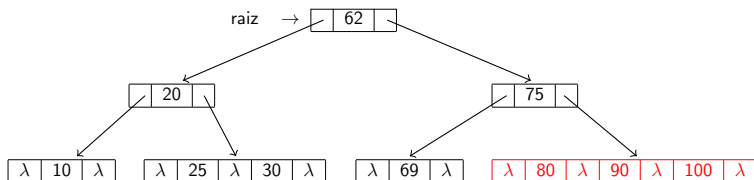
Nó folha com capacidade máxima!



Nova quebra de nó!

Ilustração de Inserção em Árvore B

Nó folha com capacidade máxima!



Nova quebra de nó!

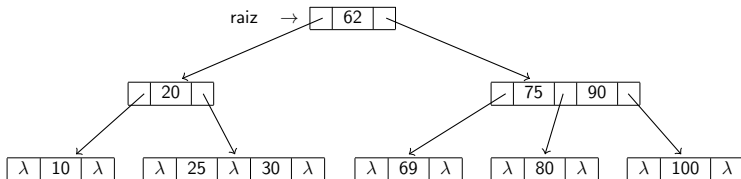
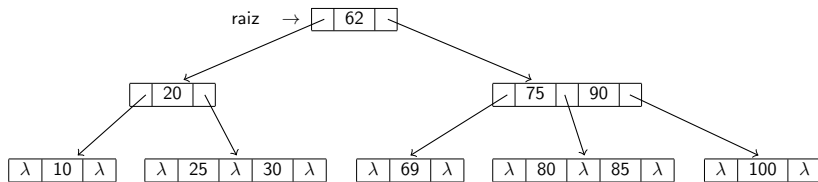


Ilustração de Inserção em Árvore B

Inserção do 85 na árvore:



Árvore B - Algoritmo de Inserção

Ver descrição em [Cormen *et al.* 2009]

Árvore B - Algoritmo de Busca

Algorithm 1: Busca em Árvore B

entrada: Apontador x para a raiz da árvore (não vazia) e chave k

saida : Se x está na árvore, retorna o par (y, i) , em que y é um nó e i é um índice tal que $y.key_i = k$. Se não, retorna NULL

```
1  $i \leftarrow 1$ ;  
2 while  $(i \leq x.n) \wedge (k > x.key_i)$  do  
3    $i \leftarrow i + 1$   
4 if  $(i \leq x.n) \wedge (k = x.key_i)$  then  
5   return  $(x, i)$   
6 else if  $x.leaf$  then return NULL;  
7 else  
8   DiskRead( $x.c_i$ );  
9   return B-TREE-SEARCH( $x.c_i, k$ )
```

Propriedade da Altura de Árvores B

Teorema:

Seja n o número de nós em uma árvore B. Se $n \geq 1$ então para qualquer árvore B de altura h e grau mínimo $t \geq 2$:

$$\log_{2t} (n + 1) - 1 \leq h \leq \log_t \frac{n+1}{2}$$

Para o teorema acima, o nível da raiz é zero. Parte do teorema acima aparece como Teorema 18.1 em [Cormen *et al.* 2009].

Propriedade da Altura de Árvores B

Teorema:

Seja n o número de nós em uma árvore B. Se $n \geq 1$ então para qualquer árvore B de altura h e grau mínimo $t \geq 2$:

$$\log_{2t} (n + 1) - 1 \leq h \leq \log_t \frac{n+1}{2}$$

Para o teorema acima, o nível da raiz é zero. Parte do teorema acima aparece como Teorema 18.1 em [Cormen *et al.* 2009].

Como consequência do teorema acima, as operações de consulta e inserção fazem $\Theta(\log n)$ acessos a nós no pior caso.

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0		

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1		

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2		

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$
3	$(2t)^3$	$(2t)^3(2t - 1)$
\vdots	\vdots	\vdots

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$
3	$(2t)^3$	$(2t)^3(2t - 1)$
\vdots	\vdots	\vdots
h	$(2t)^h$	$(2t)^h(2t - 1)$

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$
3	$(2t)^3$	$(2t)^3(2t - 1)$
\vdots	\vdots	\vdots
h	$(2t)^h$	$(2t)^h(2t - 1)$

Núm. máx. chaves para altura h :

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$
3	$(2t)^3$	$(2t)^3(2t - 1)$
\vdots	\vdots	\vdots
h	$(2t)^h$	$(2t)^h(2t - 1)$

Núm. máx. chaves para altura h :

$$n = (2t - 1) + (2t)(2t - 1) + (2t)^2(2t - 1) + \dots + (2t)^h(2t - 1)$$

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$
3	$(2t)^3$	$(2t)^3(2t - 1)$
\vdots	\vdots	\vdots
h	$(2t)^h$	$(2t)^h(2t - 1)$

Núm. máx. chaves para altura h :

$$n = (2t - 1) + (2t)(2t - 1) + (2t)^2(2t - 1) + \dots + (2t)^h(2t - 1)$$

$$n = (2t - 1)(1 + (2t) + (2t)^2 + \dots + (2t)^h)$$

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$
3	$(2t)^3$	$(2t)^3(2t - 1)$
\vdots	\vdots	\vdots
h	$(2t)^h$	$(2t)^h(2t - 1)$

Núm. máx. chaves para altura h :

$$n = (2t - 1) + (2t)(2t - 1) + (2t)^2(2t - 1) + \dots + (2t)^h(2t - 1)$$

$$n = (2t - 1)(1 + (2t) + (2t)^2 + \dots + (2t)^h)$$

$$n = (2t - 1) \frac{(2t)^{h+1} - 1}{(2t - 1)}$$

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$
3	$(2t)^3$	$(2t)^3(2t - 1)$
\vdots	\vdots	\vdots
h	$(2t)^h$	$(2t)^h(2t - 1)$

Núm. máx. chaves para altura h :

$$n = (2t - 1) + (2t)(2t - 1) + (2t)^2(2t - 1) + \dots + (2t)^h(2t - 1)$$

$$n = (2t - 1)(1 + (2t) + (2t)^2 + \dots + (2t)^h)$$

$$n = (2t - 1) \frac{(2t)^{h+1} - 1}{(2t - 1)}$$

$$n + 1 = (2t)^{h+1}$$

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$
3	$(2t)^3$	$(2t)^3(2t - 1)$
\vdots	\vdots	\vdots
h	$(2t)^h$	$(2t)^h(2t - 1)$

Núm. máx. chaves para altura h :

$$n = (2t - 1) + (2t)(2t - 1) + (2t)^2(2t - 1) + \dots + (2t)^h(2t - 1)$$

$$n = (2t - 1)(1 + (2t) + (2t)^2 + \dots + (2t)^h)$$

$$n = (2t - 1) \frac{(2t)^{h+1} - 1}{(2t - 1)}$$

$$n + 1 = (2t)^{h+1}$$

$$\log_{2t}(n + 1) = h + 1$$

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$
3	$(2t)^3$	$(2t)^3(2t - 1)$
\vdots	\vdots	\vdots
h	$(2t)^h$	$(2t)^h(2t - 1)$

Núm. máx. chaves para altura h :

$$n = (2t - 1) + (2t)(2t - 1) + (2t)^2(2t - 1) + \dots + (2t)^h(2t - 1)$$

$$n = (2t - 1)(1 + (2t) + (2t)^2 + \dots + (2t)^h)$$

$$n = (2t - 1) \frac{(2t)^{h+1} - 1}{(2t - 1)}$$

$$n + 1 = (2t)^{h+1}$$

$$\log_{2t}(n + 1) = h + 1$$

$$h = \log_{2t}(n + 1) - 1$$

Esquema da Prova

Nível	Núm. Máx. Nós	Núm. Máx. Chaves
0	1	$2t - 1$
1	$2t$	$2t(2t - 1)$
2	$(2t)^2$	$(2t)^2(2t - 1)$
3	$(2t)^3$	$(2t)^3(2t - 1)$
\vdots	\vdots	\vdots
h	$(2t)^h$	$(2t)^h(2t - 1)$

Núm. máx. chaves para altura h :

$$n = (2t - 1) + (2t)(2t - 1) + (2t)^2(2t - 1) + \dots + (2t)^h(2t - 1)$$

$$n = (2t - 1)(1 + (2t) + (2t)^2 + \dots + (2t)^h)$$

$$n = (2t - 1) \frac{(2t)^{h+1} - 1}{(2t - 1)}$$

$$n + 1 = (2t)^{h+1}$$

$$\log_{2t}(n + 1) = h + 1$$

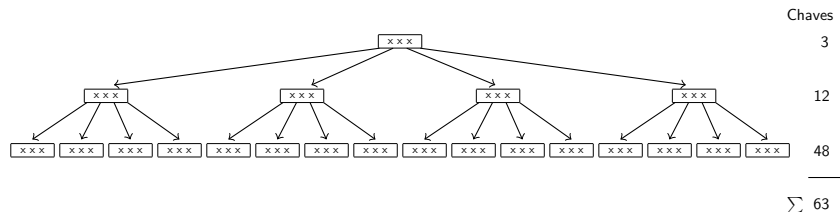
$$h = \log_{2t}(n + 1) - 1$$

Como é o número máximo de chaves para a altura h , tem-se que:

$$\log_{2t}(n + 1) - 1 \leq h, \text{ para um dado valor de } n$$

Esquema da Prova - Ilustração

Número máximo de chaves por nó: $t = 2$:



Por exemplo:

$h \geq \log_{2t}(n+1) - 1$

Para $t = 2$ e $n = 15$: $h \geq 1$ e $\log_4 16 - 1 = 1$
Para $t = 2$ e $n = 63$: $h \geq 2$ e $\log_4 64 - 1 = 2$
Para $t = 2$ e $n = 20$: $h \geq 2$ e $\log_4 21 - 1 = 1.196$

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0		

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1		

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t - 1)$
2		

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t - 1)$
2	$2t$	

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t - 1)$
2	$2t$	$2t(t - 1)$

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t - 1)$
2	$2t$	$2t(t - 1)$
3	$2t^2$	$2t^2(t - 1)$
\vdots	\vdots	\vdots

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t - 1)$
2	$2t$	$2t(t - 1)$
3	$2t^2$	$2t^2(t - 1)$
\vdots	\vdots	\vdots
h	$2t^{h-1}$	$2t^{h-1}(t - 1)$

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t-1)$
2	$2t$	$2t(t-1)$
3	$2t^2$	$2t^2(t-1)$
\vdots	\vdots	\vdots
h	$2t^{h-1}$	$2t^{h-1}(t-1)$

Núm. mín. chaves para altura h :

$$n = 1 + 2(t-1) + 2t(t-1) + 2t^2(t-1) + \dots + 2t^{h-1}(t-1)$$

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t-1)$
2	$2t$	$2t(t-1)$
3	$2t^2$	$2t^2(t-1)$
\vdots	\vdots	\vdots
h	$2t^{h-1}$	$2t^{h-1}(t-1)$

Núm. mín. chaves para altura h :

$$n = 1 + 2(t-1) + 2t(t-1) + 2t^2(t-1) + \dots + 2t^{h-1}(t-1)$$

$$n = 1 + 2(t-1)(1 + t + t^2 + \dots + t^{h-1})$$

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t-1)$
2	$2t$	$2t(t-1)$
3	$2t^2$	$2t^2(t-1)$
\vdots	\vdots	\vdots
h	$2t^{h-1}$	$2t^{h-1}(t-1)$

Núm. mín. chaves para altura h :

$$n = 1 + 2(t-1) + 2t(t-1) + 2t^2(t-1) + \dots + 2t^{h-1}(t-1)$$

$$n = 1 + 2(t-1)(1 + t + t^2 + \dots + t^{h-1})$$

$$n = 1 + 2(t-1) \frac{t^h - 1}{(t-1)}$$

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t-1)$
2	$2t$	$2t(t-1)$
3	$2t^2$	$2t^2(t-1)$
\vdots	\vdots	\vdots
h	$2t^{h-1}$	$2t^{h-1}(t-1)$

Núm. mín. chaves para altura h :

$$n = 1 + 2(t-1) + 2t(t-1) + 2t^2(t-1) + \dots + 2t^{h-1}(t-1)$$

$$n = 1 + 2(t-1)(1 + t + t^2 + \dots + t^{h-1})$$

$$n = 1 + 2(t-1) \frac{t^h - 1}{t - 1}$$

$$n = 2t^h - 1$$

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t-1)$
2	$2t$	$2t(t-1)$
3	$2t^2$	$2t^2(t-1)$
\vdots	\vdots	\vdots
h	$2t^{h-1}$	$2t^{h-1}(t-1)$

Núm. mín. chaves para altura h :

$$n = 1 + 2(t-1) + 2t(t-1) + 2t^2(t-1) + \dots + 2t^{h-1}(t-1)$$

$$n = 1 + 2(t-1)(1 + t + t^2 + \dots + t^{h-1})$$

$$n = 1 + 2(t-1) \frac{t^h - 1}{t - 1}$$

$$n = 2t^h - 1$$

$$\frac{n+1}{2} = t^h$$

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t-1)$
2	$2t$	$2t(t-1)$
3	$2t^2$	$2t^2(t-1)$
\vdots	\vdots	\vdots
h	$2t^{h-1}$	$2t^{h-1}(t-1)$

Núm. mín. chaves para altura h :

$$n = 1 + 2(t-1) + 2t(t-1) + 2t^2(t-1) + \dots + 2t^{h-1}(t-1)$$

$$n = 1 + 2(t-1)(1 + t + t^2 + \dots + t^{h-1})$$

$$n = 1 + 2(t-1) \frac{t^h - 1}{t - 1}$$

$$n = 2t^h - 1$$

$$\frac{n+1}{2} = t^h$$

$$h = \log_t \frac{n+1}{2}$$

Esquema da Prova (cont)

Nível	Núm. Mín. Nós	Núm. Mín. Chaves
0	1	1
1	2	$2(t-1)$
2	$2t$	$2t(t-1)$
3	$2t^2$	$2t^2(t-1)$
\vdots	\vdots	\vdots
h	$2t^{h-1}$	$2t^{h-1}(t-1)$

Núm. mín. chaves para altura h :

$$n = 1 + 2(t-1) + 2t(t-1) + 2t^2(t-1) + \dots + 2t^{h-1}(t-1)$$

$$n = 1 + 2(t-1)(1 + t + t^2 + \dots + t^{h-1})$$

$$n = 1 + 2(t-1) \frac{t^h - 1}{t - 1}$$

$$n = 2t^h - 1$$

$$\frac{n+1}{2} = t^h$$

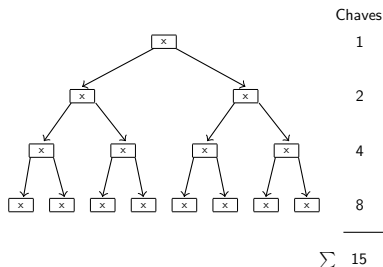
$$h = \log_t \frac{n+1}{2}$$

Como é o número mínimo de chaves para altura h , tem-se que:

$$h \leq \log_t \frac{n+1}{2}, \text{ para um dado valor de } n$$

Esquema da Prova - Ilustração

Número mínimo de chaves por nó: $t = 2$:



Por exemplo:

$$h \leq \log_t \frac{n+1}{2}$$

Para $t = 2$ e $n = 7$: $h \leq 2$ e $\log_2 \left(\frac{7+1}{2} \right) = 2$

Para $t = 2$ e $n = 15$: $h \leq 3$ e $\log_2 \left(\frac{15+1}{2} \right) = 3$

Para $t = 2$ e $n = 10$: $h \leq 2$ e $\log_2 \left(\frac{10+1}{2} \right) = 2.46$

Exercícios

1. Apresentar o estado final de uma árvore B de grau mínimo $t = 3$ após a inserção da seguinte sequência de chaves:

10, 75, 62, 69, 80, 90, 20, 30, 100, 25, 85, 95

Exercícios

1. Apresentar o estado final de uma árvore B de grau mínimo $t = 3$ após a inserção da seguinte sequência de chaves:

10, 75, 62, 69, 80, 90, 20, 30, 100, 25, 85, 95

Resposta:

Inserção das chaves 10, 75, 62, 69, 80:

10	62	69	75	80
----	----	----	----	----

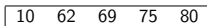
Exercícios

1. Apresentar o estado final de uma árvore B de grau mínimo $t = 3$ após a inserção da seguinte sequência de chaves:

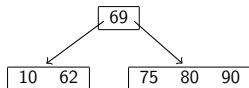
10, 75, 62, 69, 80, 90, 20, 30, 100, 25, 85, 95

Resposta:

Inserção das chaves 10, 75, 62, 69, 80:

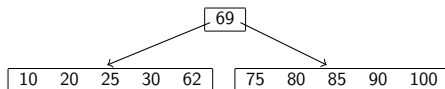


Inserção da chave 90:



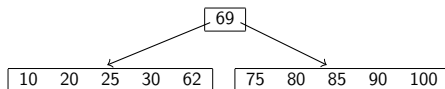
Exercícios

Inserção das chaves 20, 30, 100, 25, 85:

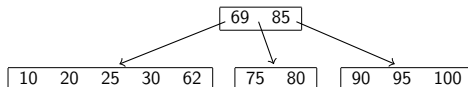


Exercícios

Inserção das chaves 20, 30, 100, 25, 85:



Inserção da chave 95:



Exercícios

2. Apresentar o estado final de uma árvore B de grau mínimo $t = 2$ após a inserção da seguinte sequência de chaves:

40, 50, 60, 30, 20, 35, 70, 80, 90, 65

Exercícios

2. Apresentar o estado final de uma árvore B de grau mínimo $t = 2$ após a inserção da seguinte sequência de chaves:

40, 50, 60, 30, 20, 35, 70, 80, 90, 65

Resposta:

Inserção das chaves 40, 50 e 60:

40	50	60
----	----	----

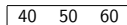
Exercícios

2. Apresentar o estado final de uma árvore B de grau mínimo $t = 2$ após a inserção da seguinte sequência de chaves:

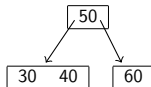
40, 50, 60, 30, 20, 35, 70, 80, 90, 65

Resposta:

Inserção das chaves 40, 50 e 60:



Inserção da chave 30:



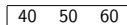
Exercícios

2. Apresentar o estado final de uma árvore B de grau mínimo $t = 2$ após a inserção da seguinte sequência de chaves:

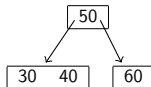
40, 50, 60, 30, 20, 35, 70, 80, 90, 65

Resposta:

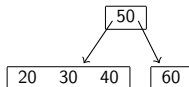
Inserção das chaves 40, 50 e 60:



Inserção da chave 30:

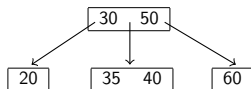


Inserção da chave 20:



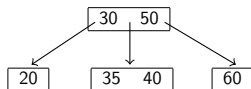
Exercícios

Inserção da chave 35:

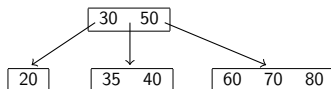


Exercícios

Inserção da chave 35:

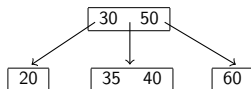


Inserção das chaves 70 e 80:

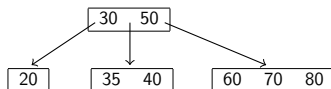


Exercícios

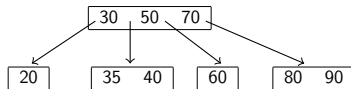
Inserção da chave 35:



Inserção das chaves 70 e 80:

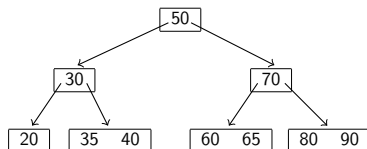


Inserção da chave 90:



Exercícios

Inserção do 65:



Remoção

Como remover chaves de uma árvore B mantendo-se suas propriedades?

Remoção

Como remover chaves de uma árvore B mantendo-se suas propriedades?

Deve-se evitar que, ao se descer na estrutura da árvore, a remoção de uma chave faça com que ajustes devam ser feitos nos nós nos níveis de volta à raiz.

Remoção

Como remover chaves de uma árvore B mantendo-se suas propriedades?

Deve-se evitar que, ao se descer na estrutura da árvore, a remoção de uma chave faça com que ajustes devam ser feitos nos nós nos níveis de volta à raiz.

- ▶ Ao se visitar um nó, com exceção da raiz, este nó deve ter pelo menos t chaves.

Remoção

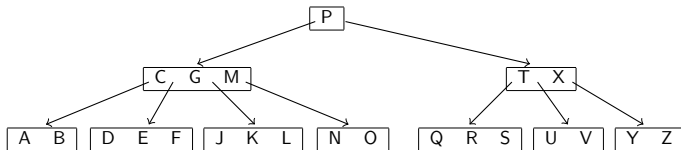
Como remover chaves de uma árvore B mantendo-se suas propriedades?

Deve-se evitar que, ao se descer na estrutura da árvore, a remoção de uma chave faça com que ajustes devam ser feitos nos nós nos níveis de volta à raiz.

- ▶ Ao se visitar um nó, com exceção da raiz, este nó deve ter pelo menos t chaves.

Há três casos genéricos de remoção. Será usada a árvore abaixo para ilustrá-los:

Grau mínimo $t = 3$



1. Se a chave k estiver em um nó x folha, remova k do nó

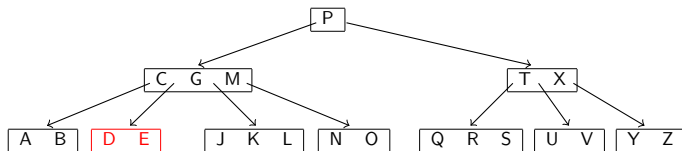
1. Se a chave k estiver em um nó x folha, remova k do nó

Remoção da chave: **F**

Remoção - Casos

1. Se a chave k estiver em um nó x folha, remova k do nó

Remoção da chave: **F**



Remoção - Casos

2. Se a chave k estiver em um nó x e x for um nó interno, faça o seguinte:
 - 2.1 se o nó filho y que precede k no nó x tiver pelo menos t chaves, então encontre o predecessor k' de k na subárvore com raiz em y .
Rekursivamente remova k' e substitua k por k' em x
 - 2.2 se y tiver menos do que t chaves, então, simetricamente, examine o nó filho z que segue k no nó x . Se z tiver pelo menos t chaves então encontre o sucessor k' de k na subárvore com raiz em z .
Rekursivamente remova k' e substitua k por k' em x
 - 2.3 se não, se ambos os filhos y e z tiverem $t - 1$ chaves, armazene z e todas as chaves e apontadores de z em y e remova k e os apontadores y e z de x . O nó y agora conterá $2t - 1$ chaves.
Apague z e recursivamente remova k de y

Remoção - Casos

2. Se a chave k estiver em um nó x e x for um nó interno, faça o seguinte:
 - 2.1 se o nó filho y que precede k no nó x tiver pelo menos t chaves, então encontre o predecessor k' de k na subárvore com raiz em y .
Rekursivamente remova k' e substitua k por k' em x
 - 2.2 se y tiver menos do que t chaves, então, simetricamente, examine o nó filho z que segue k no nó x . Se z tiver pelo menos t chaves então encontre o sucessor k' de k na subárvore com raiz em z .
Rekursivamente remova k' e substitua k por k' em x
 - 2.3 se não, se ambos os filhos y e z tiverem $t - 1$ chaves, armazene z e todas as chaves e apontadores de z em y e remova k e os apontadores y e z de x . O nó y agora conterá $2t - 1$ chaves.
Apague z e recursivamente remova k de y

Remoção da chave **M**

Remoção - Casos

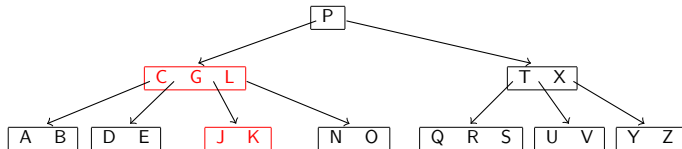
2. Se a chave k estiver em um nó x e x for um nó interno, faça o seguinte:
 - 2.1 se o nó filho y que precede k no nó x tiver pelo menos t chaves, então encontre o predecessor k' de k na subárvore com raiz em y .
Rekursivamente remova k' e substitua k por k' em x
 - 2.2 se y tiver menos do que t chaves, então, simetricamente, examine o nó filho z que segue k no nó x . Se z tiver pelo menos t chaves então encontre o sucessor k' de k na subárvore com raiz em z .
Rekursivamente remova k' e substitua k por k' em x
 - 2.3 se não, se ambos os filhos y e z tiverem $t - 1$ chaves, armazene z e todas as chaves e apontadores de z em y e remova k e os apontadores y e z de x . O nó y agora conterá $2t - 1$ chaves.
Apague z e recursivamente remova k de y

Remoção da chave **M** (Caso 2.1):

Remoção - Casos

2. Se a chave k estiver em um nó x e x for um nó interno, faça o seguinte:
 - 2.1 se o nó filho y que precede k no nó x tiver pelo menos t chaves, então encontre o predecessor k' de k na subárvore com raiz em y .
Rekursivamente remova k' e substitua k por k' em x
 - 2.2 se y tiver menos do que t chaves, então, simetricamente, examine o nó filho z que segue k no nó x . Se z tiver pelo menos t chaves então encontre o sucessor k' de k na subárvore com raiz em z .
Rekursivamente remova k' e substitua k por k' em x
 - 2.3 se não, se ambos os filhos y e z tiverem $t - 1$ chaves, armazene z e todas as chaves e apontadores de z em y e remova k e os apontadores y e z de x . O nó y agora conterá $2t - 1$ chaves.
Apague z e recursivamente remova k de y

Remoção da chave **M** (Caso 2.1):



Remoção - Casos

2. Se a chave k estiver em um nó x e x for um nó interno, faça o seguinte:
 - 2.1 se o nó filho y que precede k no nó x tiver pelo menos t chaves, então encontre o predecessor k' de k na subárvore com raiz em y .
Rekursivamente remova k' e substitua k por k' em x
 - 2.2 se y tiver menos do que t chaves, então, simetricamente, examine o nó filho z que segue k no nó x . Se z tiver pelo menos t chaves então encontre o sucessor k' de k na subárvore com raiz em z .
Rekursivamente remova k' e substitua k por k' em x
 - 2.3 se não, se ambos os filhos y e z tiverem $t - 1$ chaves, armazene z e todas as chaves e apontadores de z em y e remova k e os apontadores y e z de x . O nó y agora conterá $2t - 1$ chaves.
Apague z e recursivamente remova k de y

Remoção - Casos

2. Se a chave k estiver em um nó x e x for um nó interno, faça o seguinte:
 - 2.1 se o nó filho y que precede k no nó x tiver pelo menos t chaves, então encontre o predecessor k' de k na subárvore com raiz em y .
Rekursivamente remova k' e substitua k por k' em x
 - 2.2 se y tiver menos do que t chaves, então, simetricamente, examine o nó filho z que segue k no nó x . Se z tiver pelo menos t chaves então encontre o sucessor k' de k na subárvore com raiz em z .
Rekursivamente remova k' e substitua k por k' em x
 - 2.3 se não, se ambos os filhos y e z tiverem $t - 1$ chaves, armazene z e todas as chaves e apontadores de z em y e remova k e os apontadores y e z de x . O nó y agora conterá $2t - 1$ chaves.
Apague z e recursivamente remova k de y

Remoção da chave **G**

Remoção - Casos

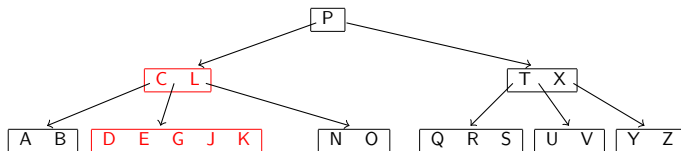
2. Se a chave k estiver em um nó x e x for um nó interno, faça o seguinte:
 - 2.1 se o nó filho y que precede k no nó x tiver pelo menos t chaves, então encontre o predecessor k' de k na subárvore com raiz em y .
Rekursivamente remova k' e substitua k por k' em x
 - 2.2 se y tiver menos do que t chaves, então, simetricamente, examine o nó filho z que segue k no nó x . Se z tiver pelo menos t chaves então encontre o sucessor k' de k na subárvore com raiz em z .
Rekursivamente remova k' e substitua k por k' em x
 - 2.3 se não, se ambos os filhos y e z tiverem $t - 1$ chaves, armazene z e todas as chaves e apontadores de z em y e remova k e os apontadores y e z de x . O nó y agora conterá $2t - 1$ chaves.
Apague z e recursivamente remova k de y

Remoção da chave **G** (Caso 2.3):

Remoção - Casos

2. Se a chave k estiver em um nó x e x for um nó interno, faça o seguinte:
 - 2.1 se o nó filho y que precede k no nó x tiver pelo menos t chaves, então encontre o predecessor k' de k na subárvore com raiz em y .
Rekursivamente remova k' e substitua k por k' em x
 - 2.2 se y tiver menos do que t chaves, então, simetricamente, examine o nó filho z que segue k no nó x . Se z tiver pelo menos t chaves então encontre o sucessor k' de k na subárvore com raiz em z .
Rekursivamente remova k' e substitua k por k' em x
 - 2.3 se não, se ambos os filhos y e z tiverem $t - 1$ chaves, armazene z e todas as chaves e apontadores de z em y e remova k e os apontadores y e z de x . O nó y agora conterá $2t - 1$ chaves.
Apague z e recursivamente remova k de y

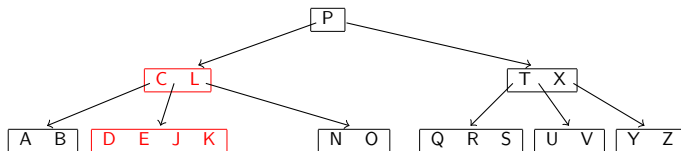
Remoção da chave **G** (Caso 2.3):



Remoção - Casos

2. Se a chave k estiver em um nó x e x for um nó interno, faça o seguinte:
 - 2.1 se o nó filho y que precede k no nó x tiver pelo menos t chaves, então encontre o predecessor k' de k na subárvore com raiz em y .
Rekursivamente remova k' e substitua k por k' em x
 - 2.2 se y tiver menos do que t chaves, então, simetricamente, examine o nó filho z que segue k no nó x . Se z tiver pelo menos t chaves então encontre o sucessor k' de k na subárvore com raiz em z .
Rekursivamente remova k' e substitua k por k' em x
 - 2.3 se não, se ambos os filhos y e z tiverem $t - 1$ chaves, armazene z e todas as chaves e apontadores de z em y e remova k e os apontadores y e z de x . O nó y agora conterá $2t - 1$ chaves.
Apague z e recursivamente remova k de y

Remoção da chave **G** (Caso 2.3):



Remoção - Casos

3. Se a chave k não estiver no nó interno x , determine a raiz $x.c_i$ da subárvore onde k deve estar (se k estiver na árvore). Se $x.c_i$ tiver somente $t - 1$ chaves, execute passos 3.1 e 3.2 tantas vezes quantas forem necessárias para que o processo continue em um nó com pelo menos t chaves. Então termine seguindo o processo recursivamente no filho apropriado de x .
 - 3.1 Se $x.c_i$ tem somente $t - 1$ chaves mas tiver um nó irmão imediato com pelo menos t chaves, mova uma chave de x a $x.c_i$. Em seguida, mova uma chave do irmão imediato de $x.c_i$ para x e mova o apontador apropriado do nó irmão para $x.c_i$
 - 3.2 Se $x.c_i$ e ambos os irmãos imediatos de $x.c_i$ tiverem $t - 1$ chaves, una $x.c_i$ com um irmão. Para isso, mova também uma chave de x para o novo nó, que se tornará a nova chave mediana desse nó

Remoção - Casos

3. Se a chave k não estiver no nó interno x , determine a raiz $x.c_i$ da subárvore onde k deve estar (se k estiver na árvore). Se $x.c_i$ tiver somente $t - 1$ chaves, execute passos 3.1 e 3.2 tantas vezes quantas forem necessárias para que o processo continue em um nó com pelo menos t chaves. Então termine seguindo o processo recursivamente no filho apropriado de x .
 - 3.1 Se $x.c_i$ tem somente $t - 1$ chaves mas tiver um nó irmão imediato com pelo menos t chaves, mova uma chave de x a $x.c_i$. Em seguida, mova uma chave do irmão imediato de $x.c_i$ para x e mova o apontador apropriado do nó irmão para $x.c_i$
 - 3.2 Se $x.c_i$ e ambos os irmãos imediatos de $x.c_i$ tiverem $t - 1$ chaves, una $x.c_i$ com um irmão. Para isso, mova também uma chave de x para o novo nó, que se tornará a nova chave mediana desse nó

Remoção da chave **D**

Remoção - Casos

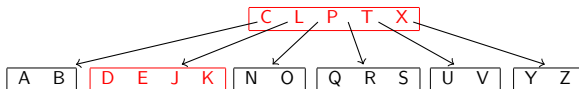
3. Se a chave k não estiver no nó interno x , determine a raiz $x.c_i$ da subárvore onde k deve estar (se k estiver na árvore). Se $x.c_i$ tiver somente $t - 1$ chaves, execute passos 3.1 e 3.2 tantas vezes quantas forem necessárias para que o processo continue em um nó com pelo menos t chaves. Então termine seguindo o processo recursivamente no filho apropriado de x .
 - 3.1 Se $x.c_i$ tem somente $t - 1$ chaves mas tiver um nó irmão imediato com pelo menos t chaves, mova uma chave de x a $x.c_i$. Em seguida, mova uma chave do irmão imediato de $x.c_i$ para x e mova o apontador apropriado do nó irmão para $x.c_i$
 - 3.2 Se $x.c_i$ e ambos os irmãos imediatos de $x.c_i$ tiverem $t - 1$ chaves, una $x.c_i$ com um irmão. Para isso, mova também uma chave de x para o novo nó, que se tornará a nova chave mediana desse nó

Remoção da chave **D** (Caso 3.2):

Remoção - Casos

3. Se a chave k não estiver no nó interno x , determine a raiz $x.c_i$ da subárvore onde k deve estar (se k estiver na árvore). Se $x.c_i$ tiver somente $t - 1$ chaves, execute passos 3.1 e 3.2 tantas vezes quantas forem necessárias para que o processo continue em um nó com pelo menos t chaves. Então termine seguindo o processo recursivamente no filho apropriado de x .
 - 3.1 Se $x.c_i$ tem somente $t - 1$ chaves mas tiver um nó irmão imediato com pelo menos t chaves, mova uma chave de x a $x.c_i$. Em seguida, mova uma chave do irmão imediato de $x.c_i$ para x e mova o apontador apropriado do nó irmão para $x.c_i$.
 - 3.2 Se $x.c_i$ e ambos os irmãos imediatos de $x.c_i$ tiverem $t - 1$ chaves, una $x.c_i$ com um irmão. Para isso, mova também uma chave de x para o novo nó, que se tornará a nova chave mediana desse nó

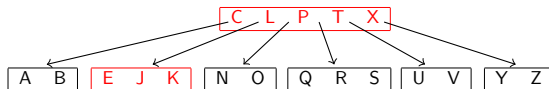
Remoção da chave **D** (Caso 3.2):



Remoção - Casos

3. Se a chave k não estiver no nó interno x , determine a raiz $x.c_i$ da subárvore onde k deve estar (se k estiver na árvore). Se $x.c_i$ tiver somente $t - 1$ chaves, execute passos 3.1 e 3.2 tantas vezes quantas forem necessárias para que o processo continue em um nó com pelo menos t chaves. Então termine seguindo o processo recursivamente no filho apropriado de x .
 - 3.1 Se $x.c_i$ tem somente $t - 1$ chaves mas tiver um nó irmão imediato com pelo menos t chaves, mova uma chave de x a $x.c_i$. Em seguida, mova uma chave do irmão imediato de $x.c_i$ para x e mova o apontador apropriado do nó irmão para $x.c_i$.
 - 3.2 Se $x.c_i$ e ambos os irmãos imediatos de $x.c_i$ tiverem $t - 1$ chaves, una $x.c_i$ com um irmão. Para isso, mova também uma chave de x para o novo nó, que se tornará a nova chave mediana desse nó

Remoção da chave **D** (Caso 3.2):



Remoção - Casos

3. Se a chave k não estiver no nó interno x , determine a raiz $x.c_i$ da subárvore onde k deve estar (se k estiver na árvore). Se $x.c_i$ tiver somente $t - 1$ chaves, execute passos 3.1 e 3.2 tantas vezes quantas forem necessárias para que o processo continue em um nó com pelo menos t chaves. Então termine seguindo o processo recursivamente no filho apropriado de x .
 - 3.1 Se $x.c_i$ tem somente $t - 1$ chaves mas tiver um nó irmão imediato com pelo menos t chaves, mova uma chave de x a $x.c_i$. Em seguida, mova uma chave do irmão imediato de $x.c_i$ para x e mova o apontador apropriado do nó irmão para $x.c_i$
 - 3.2 Se $x.c_i$ e ambos os irmãos imediatos de $x.c_i$ tiverem $t - 1$ chaves, una $x.c_i$ com um irmão. Para isso, mova também uma chave de x para o novo nó, que se tornará a nova chave mediana desse nó

Remoção - Casos

3. Se a chave k não estiver no nó interno x , determine a raiz $x.c_i$ da subárvore onde k deve estar (se k estiver na árvore). Se $x.c_i$ tiver somente $t - 1$ chaves, execute passos 3.1 e 3.2 tantas vezes quantas forem necessárias para que o processo continue em um nó com pelo menos t chaves. Então termine seguindo o processo recursivamente no filho apropriado de x .
 - 3.1 Se $x.c_i$ tem somente $t - 1$ chaves mas tiver um nó irmão imediato com pelo menos t chaves, mova uma chave de x a $x.c_i$. Em seguida, mova uma chave do irmão imediato de $x.c_i$ para x e mova o apontador apropriado do nó irmão para $x.c_i$
 - 3.2 Se $x.c_i$ e ambos os irmãos imediatos de $x.c_i$ tiverem $t - 1$ chaves, una $x.c_i$ com um irmão. Para isso, mova também uma chave de x para o novo nó, que se tornará a nova chave mediana desse nó

Remoção da chave **B**

Remoção - Casos

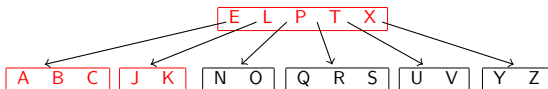
3. Se a chave k não estiver no nó interno x , determine a raiz $x.c_i$ da subárvore onde k deve estar (se k estiver na árvore). Se $x.c_i$ tiver somente $t - 1$ chaves, execute passos 3.1 e 3.2 tantas vezes quantas forem necessárias para que o processo continue em um nó com pelo menos t chaves. Então termine seguindo o processo recursivamente no filho apropriado de x .
 - 3.1 Se $x.c_i$ tem somente $t - 1$ chaves mas tiver um nó irmão imediato com pelo menos t chaves, mova uma chave de x a $x.c_i$. Em seguida, mova uma chave do irmão imediato de $x.c_i$ para x e mova o apontador apropriado do nó irmão para $x.c_i$
 - 3.2 Se $x.c_i$ e ambos os irmãos imediatos de $x.c_i$ tiverem $t - 1$ chaves, una $x.c_i$ com um irmão. Para isso, mova também uma chave de x para o novo nó, que se tornará a nova chave mediana desse nó

Remoção da chave **B** (Caso 3.1):

Remoção - Casos

3. Se a chave k não estiver no nó interno x , determine a raiz $x.c_i$ da subárvore onde k deve estar (se k estiver na árvore). Se $x.c_i$ tiver somente $t - 1$ chaves, execute passos 3.1 e 3.2 tantas vezes quantas forem necessárias para que o processo continue em um nó com pelo menos t chaves. Então termine seguindo o processo recursivamente no filho apropriado de x .
 - 3.1 Se $x.c_i$ tem somente $t - 1$ chaves mas tiver um nó irmão imediato com pelo menos t chaves, mova uma chave de x a $x.c_i$. Em seguida, mova uma chave do irmão imediato de $x.c_i$ para x e mova o apontador apropriado do nó irmão para $x.c_i$.
 - 3.2 Se $x.c_i$ e ambos os irmãos imediatos de $x.c_i$ tiverem $t - 1$ chaves, una $x.c_i$ com um irmão. Para isso, mova também uma chave de x para o novo nó, que se tornará a nova chave mediana desse nó

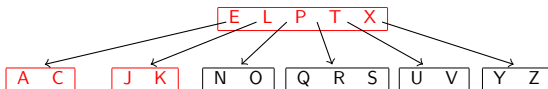
Remoção da chave **B** (Caso 3.1):



Remoção - Casos

3. Se a chave k não estiver no nó interno x , determine a raiz $x.c_i$ da subárvore onde k deve estar (se k estiver na árvore). Se $x.c_i$ tiver somente $t - 1$ chaves, execute passos 3.1 e 3.2 tantas vezes quantas forem necessárias para que o processo continue em um nó com pelo menos t chaves. Então termine seguindo o processo recursivamente no filho apropriado de x .
 - 3.1 Se $x.c_i$ tem somente $t - 1$ chaves mas tiver um nó irmão imediato com pelo menos t chaves, mova uma chave de x a $x.c_i$. Em seguida, mova uma chave do irmão imediato de $x.c_i$ para x e mova o apontador apropriado do nó irmão para $x.c_i$.
 - 3.2 Se $x.c_i$ e ambos os irmãos imediatos de $x.c_i$ tiverem $t - 1$ chaves, una $x.c_i$ com um irmão. Para isso, mova também uma chave de x para o novo nó, que se tornará a nova chave mediana desse nó

Remoção da chave **B** (Caso 3.1):



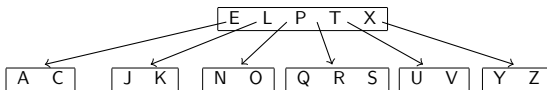
Remoção - Complexidade

Pelo Teorema apresentado anteriormente, a remoção também possui complexidade $\Theta(\log n)$ no pior caso.

Exercício

Qual seria o estado final da árvore abaixo, após a remoção da seguinte sequência de chaves: **N, R, P, U, S, T, J, A**

O grau mínimo da árvore, t , é igual a 3.

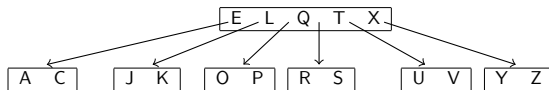


Respostas

Remoção do **N**:

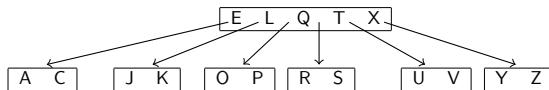
Respostas

Remoção do **N**:



Respostas

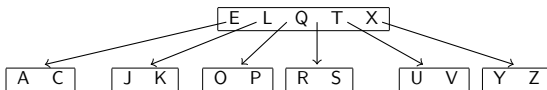
Remoção do **N**:



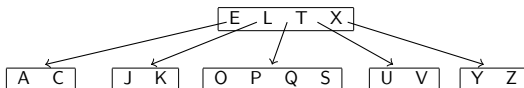
Remoção do **R**:

Respostas

Remoção do **N**:

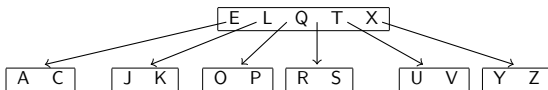


Remoção do **R**:

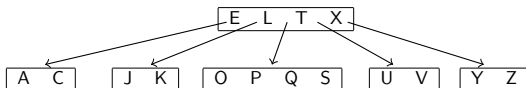


Respostas

Remoção do **N**:



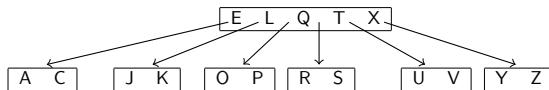
Remoção do **R**:



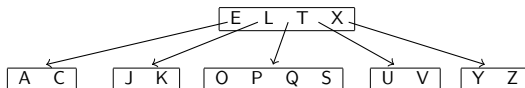
Remoção do **P**:

Respostas

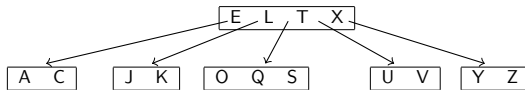
Remoção do **N**:



Remoção do **R**:



Remoção do **P**:

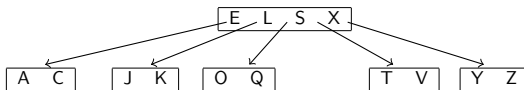


Respostas

Remoção do **U**:

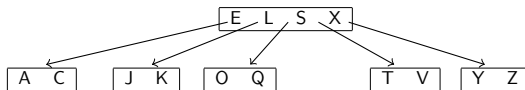
Respostas

Remoção do **U**:



Respostas

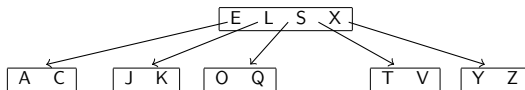
Remoção do **U**:



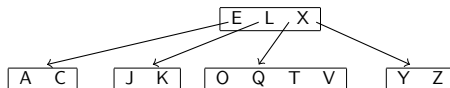
Remoção do **S**:

Respostas

Remoção do **U**:

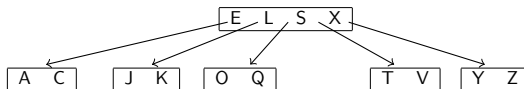


Remoção do **S**:

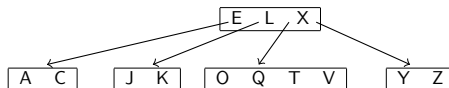


Respostas

Remoção do **U**:



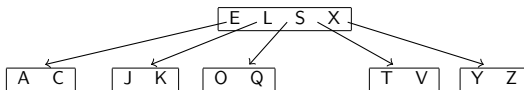
Remoção do **S**:



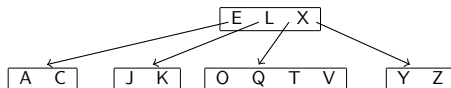
Remoção do **T**:

Respostas

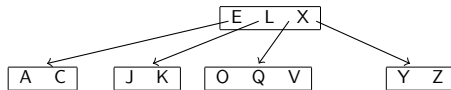
Remoção do **U**:



Remoção do **S**:

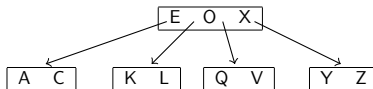


Remoção do **T**:

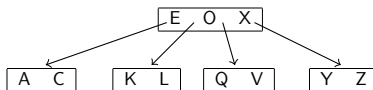


Remoção do **J**:

Remoção do **J**:



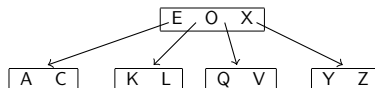
Remoção do **J**:



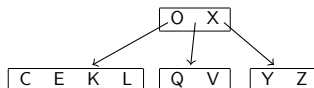
Remoção do **A**:

Respostas

Remoção do **J**:



Remoção do **A**:

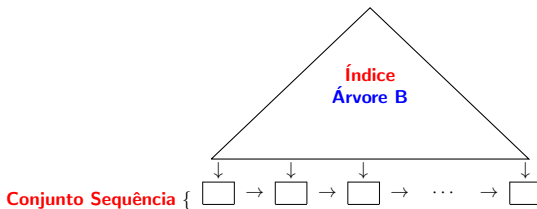


Árvores B+

Árvores B+

Variação de árvores B:

- ▶ estruturada em uma parte de **índice** e um **conjunto sequência**
- ▶ o índice é estruturado como uma árvore B - armazena **apenas chaves**
- ▶ o conjunto sequência é um encadeamento de nós, onde os registros efetivamente ficam armazenados
- ▶ o número de chaves armazenadas em cada nó do índice é em geral maior que o número de registros armazenados em cada nó do conjunto sequência



Árvore B+ - Inserção

- ▶ Árvores B+ podem ser apresentadas com pequenas variações

Árvore B+ - Inserção

- ▶ Árvores B+ podem ser apresentadas com pequenas variações
- ▶ Apresentaremos aqui árvores B+ considerando dois parâmetros:
 - ▶ : t_l : o **grau mínimo** para a parte de índice
 - ▶ : t_{cs} : um valor que indica o número mínimo e máximo de registros que podem ser armazenados nos nós do conjunto sequência

Árvore B+ - Inserção

- ▶ Árvores B+ podem ser apresentadas com pequenas variações
- ▶ Apresentaremos aqui árvores B+ considerando dois parâmetros:
 - ▶ : t_l : o **grau mínimo** para a parte de índice
 - ▶ : t_{cs} : um valor que indica o número mínimo e máximo de registros que podem ser armazenados nos nós do conjunto sequência
- ▶ Nós do índice:
 - ▶ número mínimo de chaves: $t_l - 1$
 - ▶ número máximo de chaves: $2t_l - 1$

Árvore B+ - Inserção

- ▶ Árvores B+ podem ser apresentadas com pequenas variações
- ▶ Apresentaremos aqui árvores B+ considerando dois parâmetros:
 - ▶ : t_l : o **grau mínimo** para a parte de índice
 - ▶ : t_{CS} : um valor que indica o número mínimo e máximo de registros que podem ser armazenados nos nós do conjunto sequência
- ▶ Nós do índice:
 - ▶ número mínimo de chaves: $t_l - 1$
 - ▶ número máximo de chaves: $2t_l - 1$
- ▶ Nós do conjunto sequência:
 - ▶ número mínimo de registros: $t_{CS} - 1$ (com exceção do primeiro nó na árvore: número mínimo = 1)
 - ▶ número máximo de registros: $2t_{CS} - 1$

Árvore B+ - Inserção

- ▶ Árvores B+ podem ser apresentadas com pequenas variações
- ▶ Apresentaremos aqui árvores B+ considerando dois parâmetros:
 - ▶ : t_l : o **grau mínimo** para a parte de índice
 - ▶ : t_{CS} : um valor que indica o número mínimo e máximo de registros que podem ser armazenados nos nós do conjunto sequência
- ▶ Nós do índice:
 - ▶ número mínimo de chaves: $t_l - 1$
 - ▶ número máximo de chaves: $2t_l - 1$
- ▶ Nós do conjunto sequência:
 - ▶ número mínimo de registros: $t_{CS} - 1$ (com exceção do primeiro nó na árvore: número mínimo = 1)
 - ▶ número máximo de registros: $2t_{CS} - 1$
- ▶ Os nós do conjunto sequência não armazenam apontadores

Árvore B+ - Inserção

- ▶ Árvores B+ podem ser apresentadas com pequenas variações
- ▶ Apresentaremos aqui árvores B+ considerando dois parâmetros:
 - ▶ : t_l : o **grau mínimo** para a parte de índice
 - ▶ : t_{CS} : um valor que indica o número mínimo e máximo de registros que podem ser armazenados nos nós do conjunto sequência
- ▶ Nós do índice:
 - ▶ número mínimo de chaves: $t_l - 1$
 - ▶ número máximo de chaves: $2t_l - 1$
- ▶ Nós do conjunto sequência:
 - ▶ número mínimo de registros: $t_{CS} - 1$ (com exceção do primeiro nó na árvore: número mínimo = 1)
 - ▶ número máximo de registros: $2t_{CS} - 1$
- ▶ Os nós do conjunto sequência não armazenam apontadores
- ▶ O algoritmo de inserção de árvores B se aplica de maneira análoga, considerando os valores t_l e t_{CS} .

Inserção - Ilustração

Consideremos que a árvore B do índice tenha grau mínimo $t_I = 3$ e que o conjunto sequência tenha grau mínimo $t_{CS} = 2$.

Nós do índice:

- ▶ número mínimo de chaves: $t_I - 1 = 2$
- ▶ número máximo de chaves: $2t_I - 1 = 5$

Nós do conjunto sequência:

- ▶ número mínimo de registros: $t_{CS} - 1 = 1$
- ▶ número máximo de registros: $2t_{CS} - 1 = 3$

Inserção - Ilustração

Consideremos que a árvore B do índice tenha grau mínimo $t_l = 3$ e que o conjunto sequência tenha grau mínimo $t_{cs} = 2$.

Nós do índice:

- ▶ número mínimo de chaves: $t_l - 1 = 2$
- ▶ número máximo de chaves: $2t_l - 1 = 5$

Nós do conjunto sequência:

- ▶ número mínimo de registros: $t_{cs} - 1 = 1$
- ▶ número máximo de registros: $2t_{cs} - 1 = 3$

Nos exemplos a seguir, chaves em azul na árvore indicam o armazenamento de **todo o registro**. Chaves em marrom indicam o armazenamento da **chave apenas**.

Inserção - Ilustração

Inserção de um registro com chave **50**:

CS: 50

Inserção - Ilustração

Inserção de um registro com chave **50**:

CS: 50

Inserção de um registro com chave **30**:

Inserção - Ilustração

Inserção de um registro com chave **50**:

CS:

50

Inserção de um registro com chave **30**:

CS:

30	50
----	----

Inserção - Ilustração

Inserção de um registro com chave **50**:

CS:

50

Inserção de um registro com chave **30**:

CS:

30	50
----	----

Inserção de um registro com chave **70**:

Inserção - Ilustração

Inserção de um registro com chave **50**:

CS:

50

Inserção de um registro com chave **30**:

CS:

30	50
----	----

Inserção de um registro com chave **70**:

CS:

30	50	70
----	----	----

Inserção - Ilustração

Inserção de um registro com chave **80**:

Inserção - Ilustração

Inserção de um registro com chave **80**:

Quebra do nó!

Inserção - Ilustração

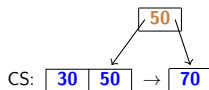
Inserção de um registro com chave **80**:

Quebra do nó! Apenas a chave sobe

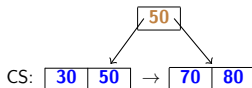
Inserção - Ilustração

Inserção de um registro com chave **80**:

Quebra do nó! Apenas a chave sobe



Inserção do registro:

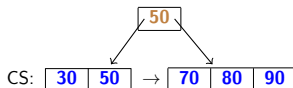


Inserção - Ilustração

Inserção de um registro com chave **90**:

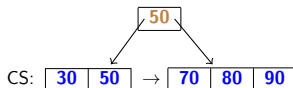
Inserção - Ilustração

Inserção de um registro com chave **90**:



Inserção - Ilustração

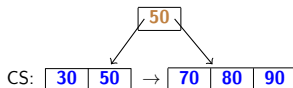
Inserção de um registro com chave **90**:



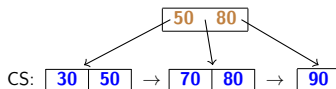
Inserção de um registro com chave **85**:

Inserção - Ilustração

Inserção de um registro com chave **90**:

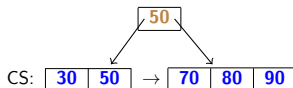


Inserção de um registro com chave **85**:

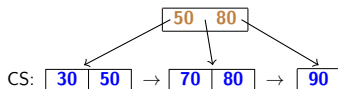


Inserção - Ilustração

Inserção de um registro com chave **90**:



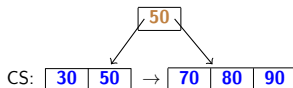
Inserção de um registro com chave **85**:



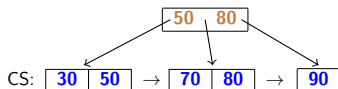
Inserção do registro:

Inserção - Ilustração

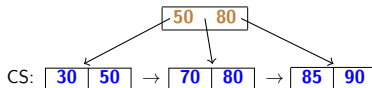
Inserção de um registro com chave **90**:



Inserção de um registro com chave **85**:



Inserção do registro:

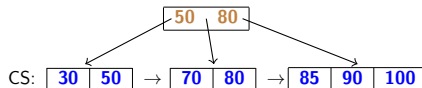


Inserção - Ilustração

Inserção de um registro com chave **100**:

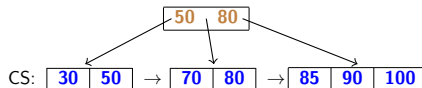
Inserção - Ilustração

Inserção de um registro com chave **100**:



Inserção - Ilustração

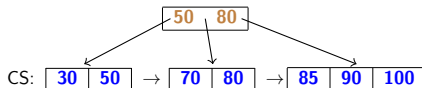
Inserção de um registro com chave **100**:



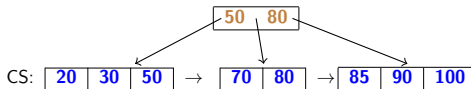
Inserção de um registro com chave **20**:

Inserção - Ilustração

Inserção de um registro com chave **100**:

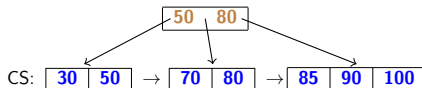


Inserção de um registro com chave **20**:

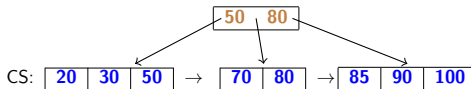


Inserção - Ilustração

Inserção de um registro com chave **100**:



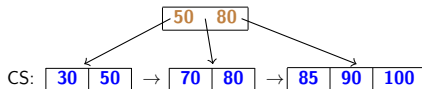
Inserção de um registro com chave **20**:



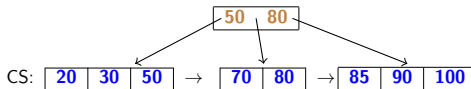
Inserção de um registro com chave **10**:

Inserção - Ilustração

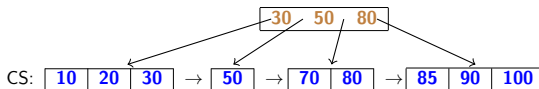
Inserção de um registro com chave **100**:



Inserção de um registro com chave **20**:

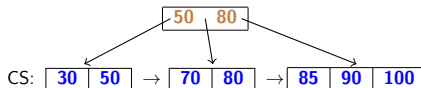


Inserção de um registro com chave **10**:

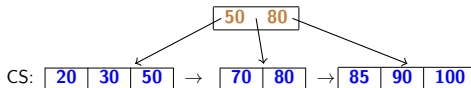


Inserção - Ilustração

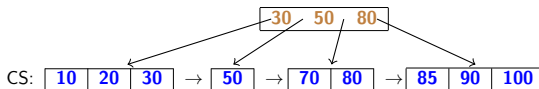
Inserção de um registro com chave **100**:



Inserção de um registro com chave **20**:



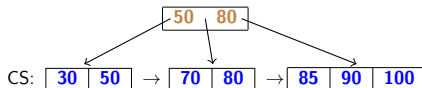
Inserção de um registro com chave **10**:



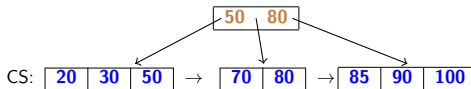
Inserção de um registro com chave **75**:

Inserção - Ilustração

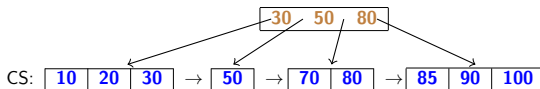
Inserção de um registro com chave **100**:



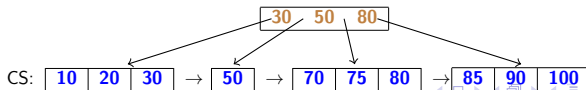
Inserção de um registro com chave **20**:



Inserção de um registro com chave **10**:



Inserção de um registro com chave **75**:

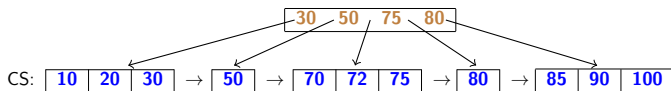


Inserção - Ilustração

Inserção de um registro com chave **72**:

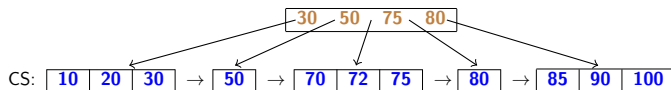
Inserção - Ilustração

Inserção de um registro com chave **72**:



Inserção - Ilustração

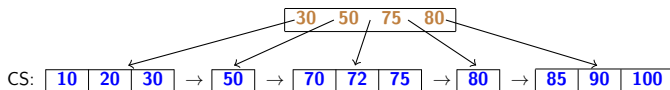
Inserção de um registro com chave **72**:



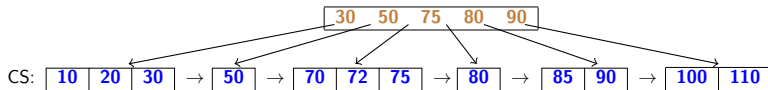
Inserção de um registro com chave **110**:

Inserção - Ilustração

Inserção de um registro com chave **72**:

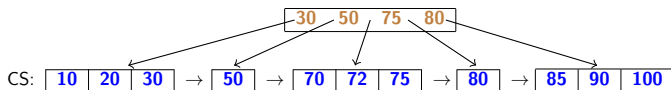


Inserção de um registro com chave **110**:

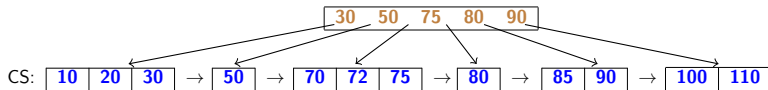


Inserção - Ilustração

Inserção de um registro com chave **72**:



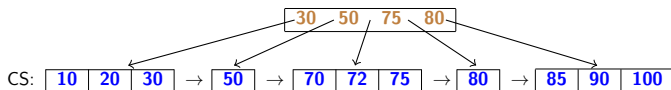
Inserção de um registro com chave **110**:



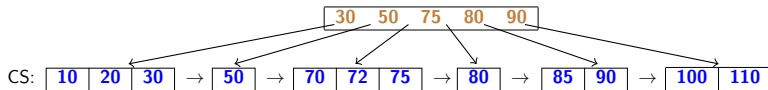
Inserção de um registro com chave **40**:

Inserção - Ilustração

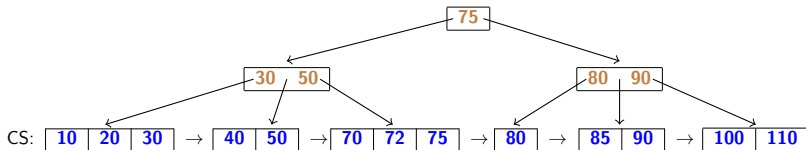
Inserção de um registro com chave **72**:



Inserção de um registro com chave **110**:



Inserção de um registro com chave **40**:



Árvores B+

- ▶ O índice tende a ter altura menor do que uma árvore B com mesmo conjunto de chaves armazenado na árvore - **apenas algumas chaves aparecem no índice!**

Árvores B+

- ▶ O índice tende a ter altura menor do que uma árvore B com mesmo conjunto de chaves armazenado na árvore - **apenas algumas chaves aparecem no índice!**
- ▶ Árvores B+ oferecem ganho de desempenho em relação a árvores B tanto no acesso direto quanto em acesso sequencial

Árvores B+

- ▶ O índice tende a ter altura menor do que uma árvore B com mesmo conjunto de chaves armazenado na árvore - **apenas algumas chaves aparecem no índice!**
- ▶ Árvores B+ oferecem ganho de desempenho em relação a árvores B tanto no acesso direto quanto em acesso sequencial
- ▶ O algoritmo de remoção é análogo ao de árvore B
 - ▶ A remoção de um registro com uma chave **c** não necessariamente causa a remoção da chave **c** do índice!
 - ▶ Ex.: remoção do registro com chave 75 na árvore final do *slide* anterior

Árvores B+

- ▶ O índice tende a ter altura menor do que uma árvore B com mesmo conjunto de chaves armazenado na árvore - **apenas algumas chaves aparecem no índice!**
- ▶ Árvores B+ oferecem ganho de desempenho em relação a árvores B tanto no acesso direto quanto em acesso sequencial
- ▶ O algoritmo de remoção é análogo ao de árvore B
 - ▶ A remoção de um registro com uma chave **c** não necessariamente causa a remoção da chave **c** do índice!
 - ▶ Ex.: remoção do registro com chave 75 na árvore final do *slide* anterior
- ▶ A busca sempre segue até um nó do conjunto sequência, pois a presença de uma chave no índice **não** indica a existência de um registro com aquela chave na árvore

Exercício

Apresente o estado de uma árvore B+ após a inserção de registros com a seguinte sequência de chaves:

6, 7, 9, 11, 14, 16, 10, 20, 21, 8, 30, 40, 50

Exercício

Apresente o estado de uma árvore B+ após a inserção de registros com a seguinte sequência de chaves:

6, 7, 9, 11, 14, 16, 10, 20, 21, 8, 30, 40, 50

Resposta:

