



Fonte: luisdev.com.br

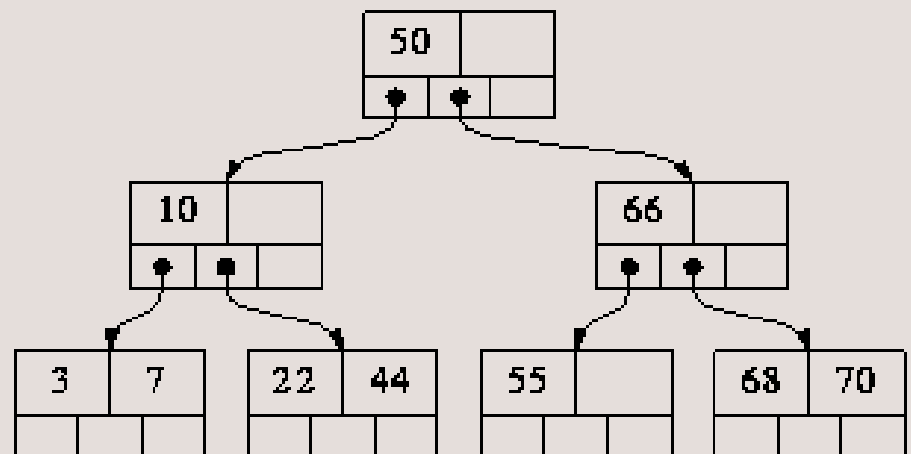
Estruturas de Dados

ÁRVORES B e B+ - Unidade 14

Prof. Kenia Kodol

Pontos Trabalhados na Disciplina

- **(Plano da Disciplina)** Ementa, Metodologia, Avaliação
- Introdução a Estruturas de Dados
- Linguagem C
- Variáveis Dinâmicas e Apontadores
- Arquivos Binários
- Eficiência de Algoritmos (Complexidade, Notação O)
- Listas Lineares
- Pilhas e Filas
- Hashing
- Árvores Binárias de Busca
- Árvores AVL
- Filas de Prioridades e Heaps
- Conjuntos Union Find



Árvore B

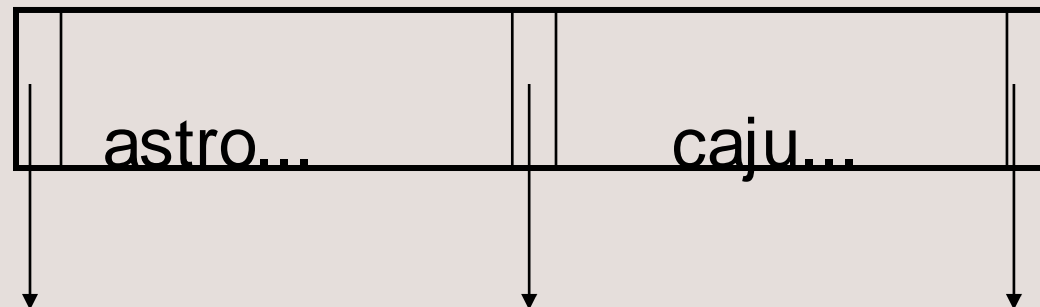
As árvores B, também denominadas Btree, são árvores cujos critérios de organização (crescimento /armazenamento de dados) as conservam balanceadas.

Apresentam crescimento *bottom up* – das folhas para a raiz, diferente, por exemplo, do crescimento de árvores binárias que é *top down* – da raiz para as folhas.

Árvore B

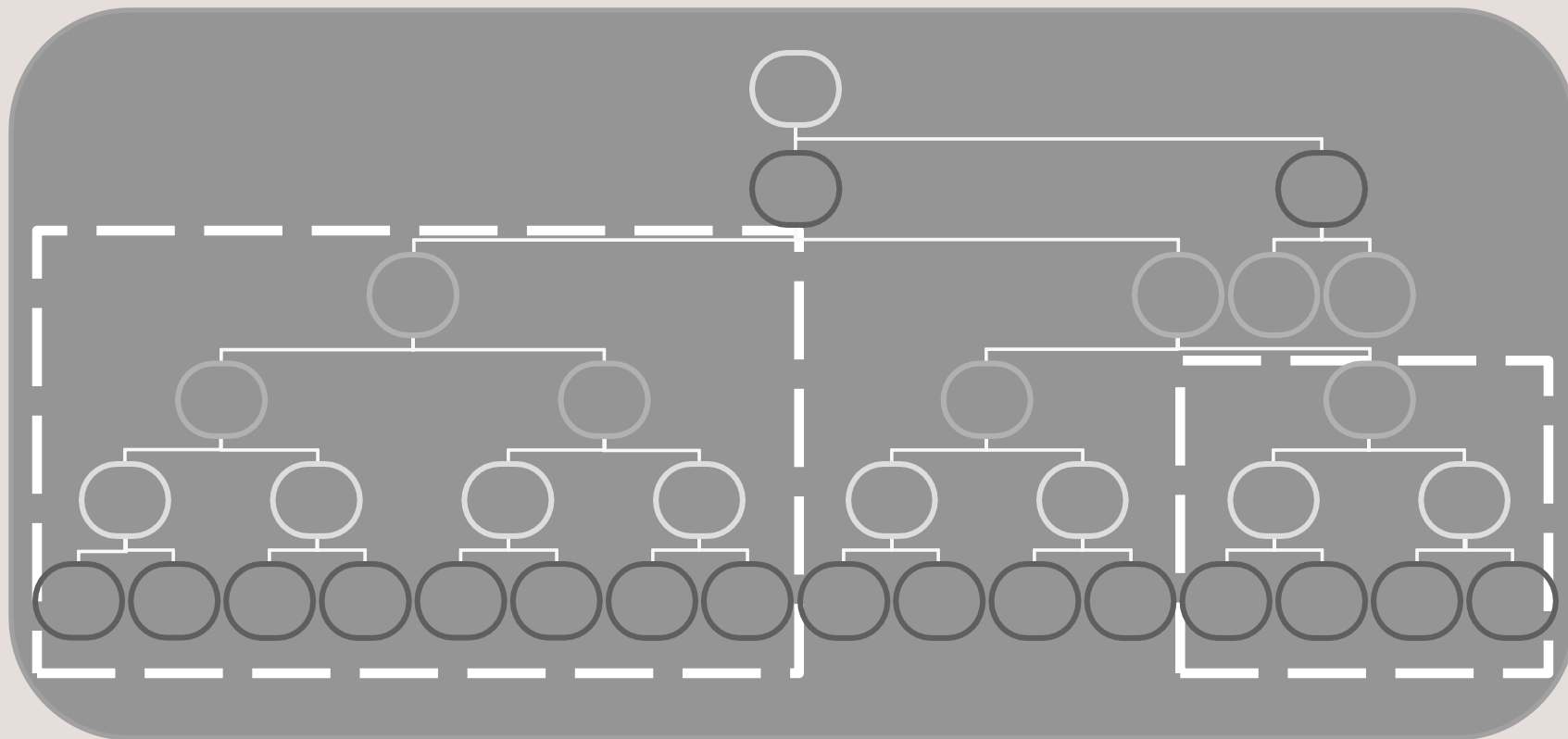
Cada nó de uma árvore B tem mais de uma entrada e múltiplas descendências.

Em cada entrada pode ser mantido registro de dados (vários campos); apesar de nestes slides serem apresentadas apenas as chaves de busca.



Árvore B

As árvores B nasceram da paginação (blocos de nós) das árvores binárias de busca. Assim sendo os nós de árvores B são denominados também páginas.



Árvore B

APLICAÇÃO

Árvores B são a estrutura subjacente a muitos sistemas de arquivos e bancos de dados. Por exemplo:

- o sistema de arquivos NTFS do SO Windows,
- os sistemas de arquivos ReiserFS, XFS, Ext3FS, JFS do SO Linux
- os bancos de dados (BD) ORACLE, DB2, INGRES, SQL e PostgreSQL



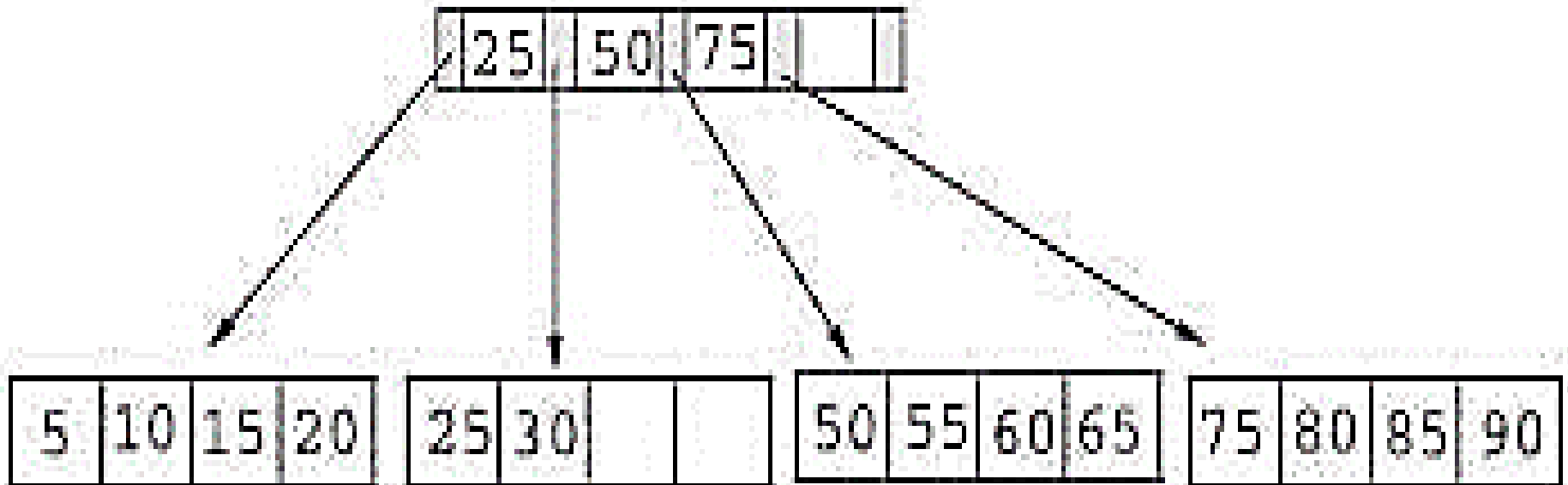
Árvore B: APLICAÇÃO

Dado um **arquivo A** - residente em memória com grande capacidade de armazenamento e baixa velocidade; considerando que A tem um número de chaves tão grande que **não cabe na memória principal do computador**, para aproveitar a velocidade desta. Para manipular A, é **preciso dividi-lo em segmentos A_1, A_2, \dots, A_n** , que **caibam na memória principal** – com alta velocidade, mas pouca capacidade de armazenamento. Os segmentos são definidos de modo que **todas as chaves em um segmento sejam menores que todas as chaves no segmento seguinte** para otimizar a busca. Esta é a origem do **conceito de árvore B**. (SEDGEWICK; WAYNE 2018)

Árvore B

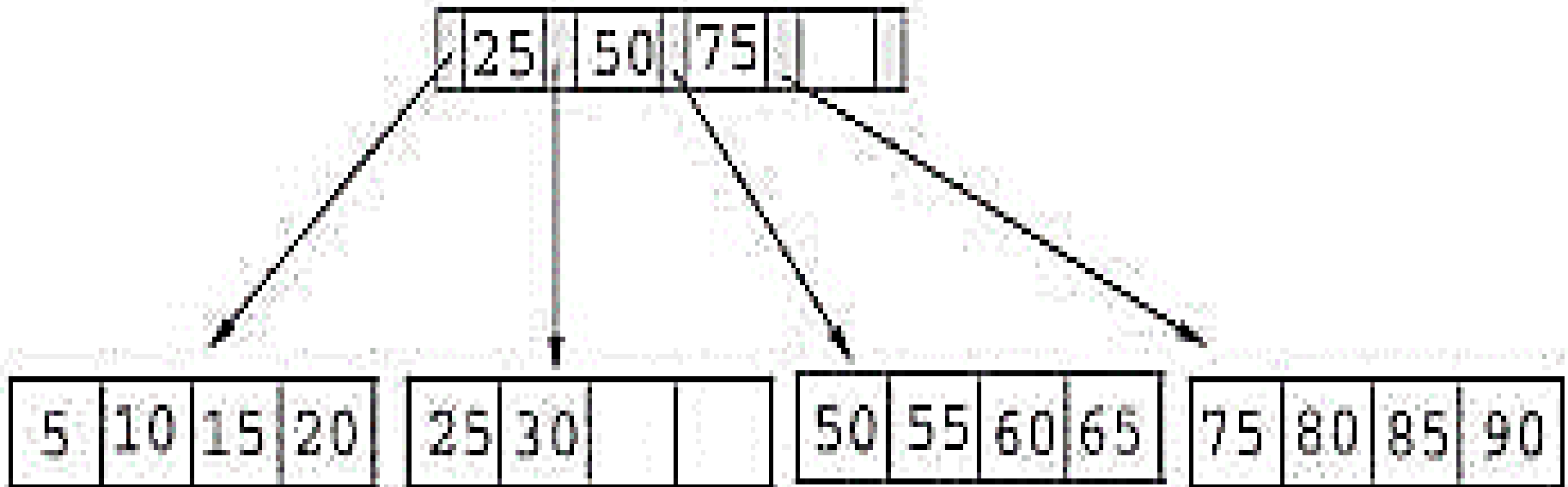
APLICAÇÃO

O tamanho dos nós das árvores B é determinado pela capacidade de armazenamento da memória principal.



Árvore B

APLICAÇÃO



Os nós (páginas) das árvores B são segmentos da base de dados A (arquivo) a ser acessada, os quais são carregados na memória principal para realização de operações sobre A, sejam de inserção remoção ou busca.

- Na organização de livros acerca de estruturas de dados, as árvores B encaixam-se em unidades sobre **árvores balanceadas**, bem como em **pesquisa externa**.
- As árvores B são balanceadas e ordenadas.
- Numa árvore B de ordem x :
 1. Todos os nós, exceto a raiz, devem armazenar de x a $2x$ dados.
 2. A raiz deve armazenar de 1 a $2x$ dados.
 3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
 4. Todas as folhas devem apresentar-se num mesmo nível.

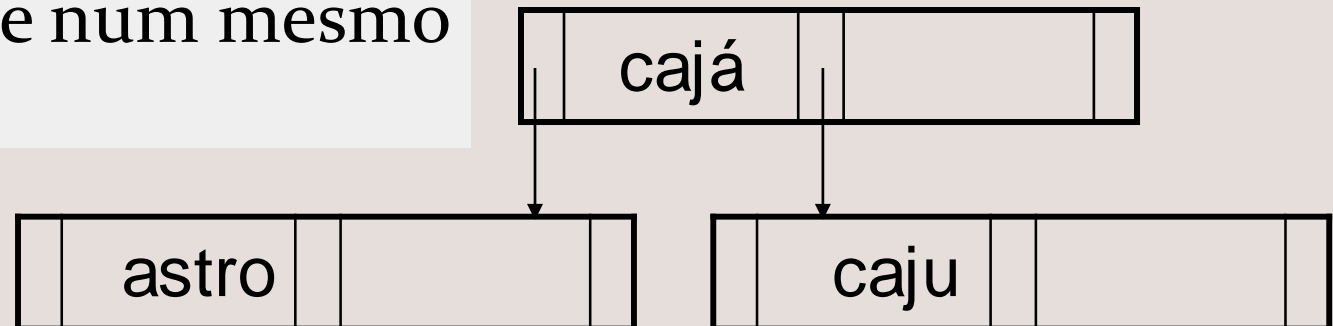
Numa árvore B de ordem x :

1. Todos os nós, exceto a raiz, devem armazenar de x a $2x$ dados.
2. A raiz deve armazenar de 1 a $2x$ dados.
3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
4. Todas as folhas devem apresentar-se num mesmo nível.

Árvore B



A árvore dada é uma árvore B?



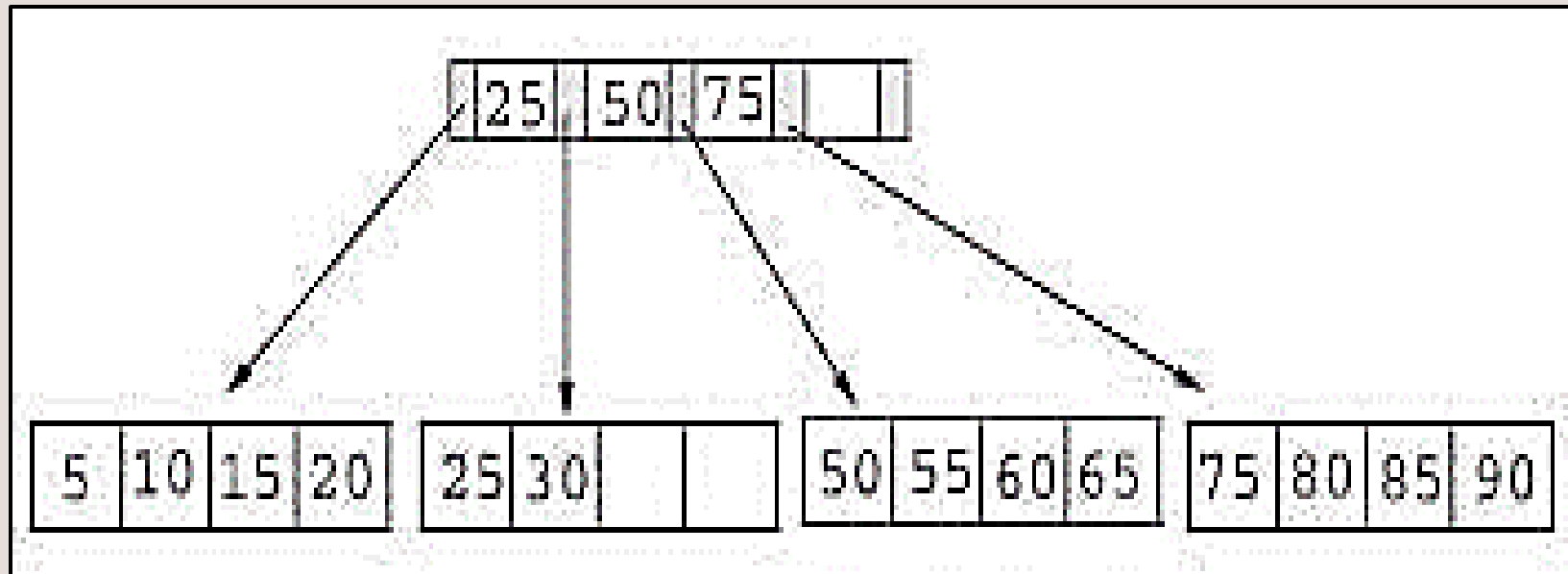
Numa árvore B de ordem x :

1. Todos os nós, exceto a raiz, devem armazenar de 1 a $2x$ dados.
2. A raiz deve armazenar de 1 a $2x$ dados.
3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
4. Todas as folhas devem apresentar-se num mesmo nível.

Árvore B



A árvore dada é uma árvore B?



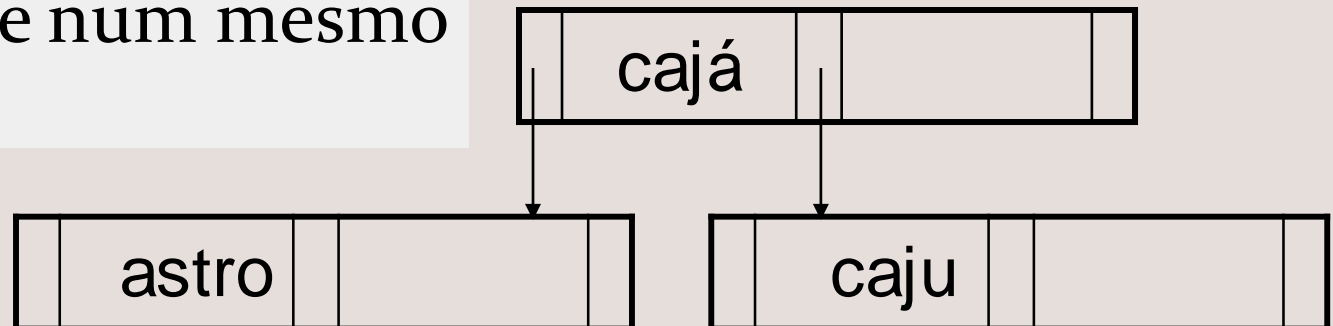
Numa árvore B de ordem x :

1. Todos os nós, exceto a raiz, devem armazenar de x a $2x$ dados.
2. A raiz deve armazenar de 1 a $2x$ dados.
3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
4. Todas as folhas devem apresentar-se num mesmo nível.

Árvore B



Qual a ordem da árvore B dada?



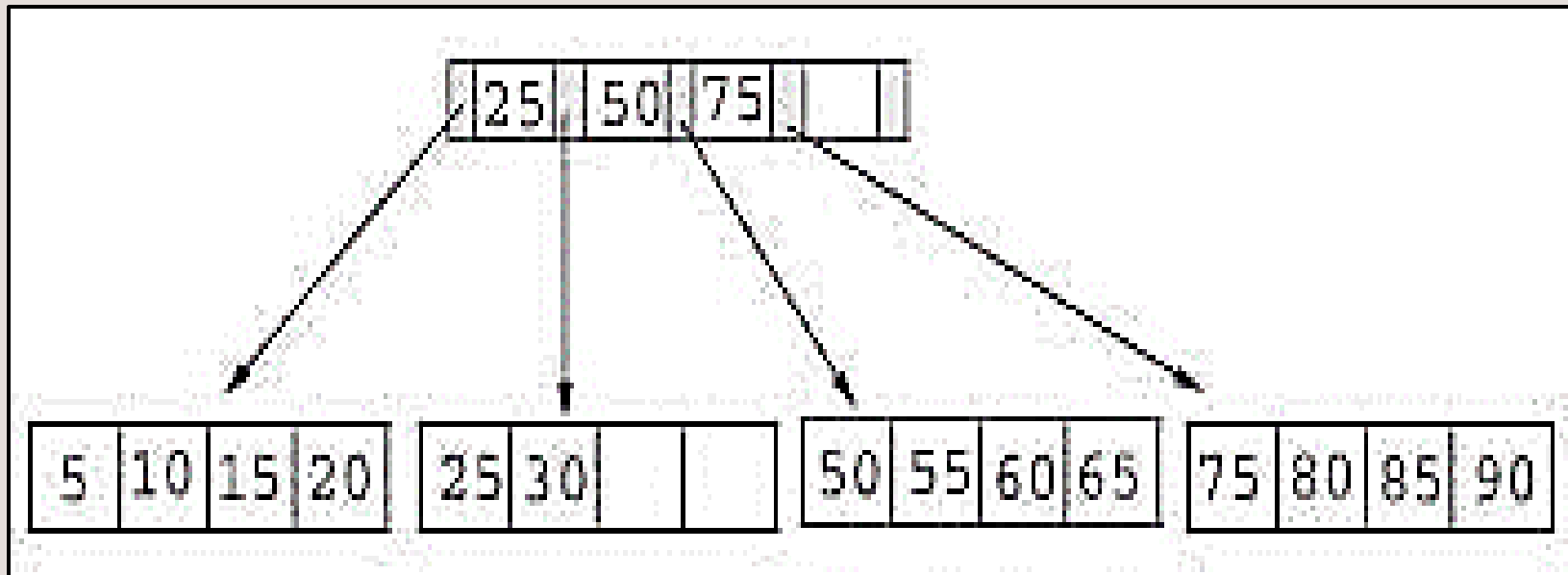
Numa árvore B de ordem x :

1. Todos os nós, exceto a raiz, devem armazenar de x a $2x$ dados.
2. A raiz deve armazenar de 1 a $2x$ dados.
3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
4. Todas as folhas devem apresentar-se num mesmo nível.

Árvore B



Qual a ordem da árvore B dada?

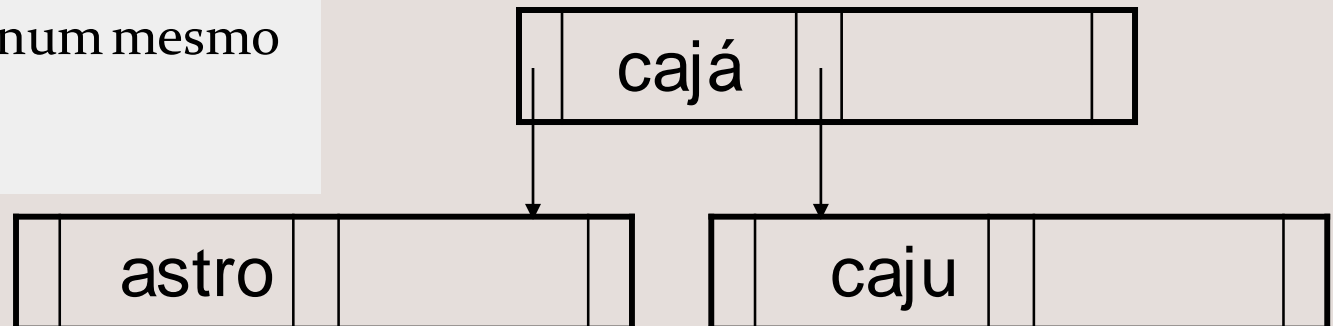


Árvore B

Numa árvore B de ordem x :

1. Todos os nós, exceto a raiz, devem armazenar de x a $2x$ dados.
2. A raiz deve armazenar de 1 a $2x$ dados.
3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
4. Todas as folhas devem apresentar-se num mesmo nível.

Segue uma árvore B de ordem 1, entretanto, vale esclarecer que as árvores B, na prática, são usadas para armazenar grande quantidade de dados; apresentando, em geral, ordens elevadas.

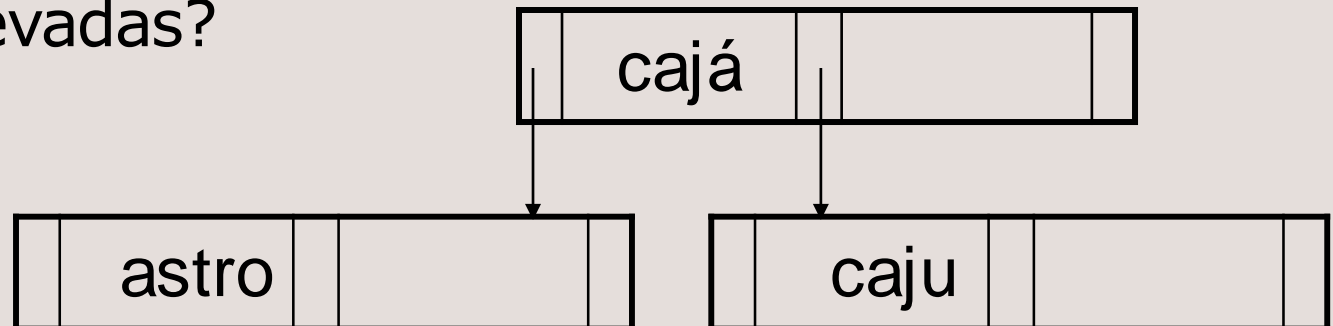


Árvore B



Por que as
árvores B têm,
em geral,
ordens
elevadas?

Segue uma árvore B de ordem
1, entretanto, vale
esclarecer que as árvores
B, na prática, são usadas
para armazenar grande
quantidade de dados;
apresentando, em geral,
ordens elevadas.



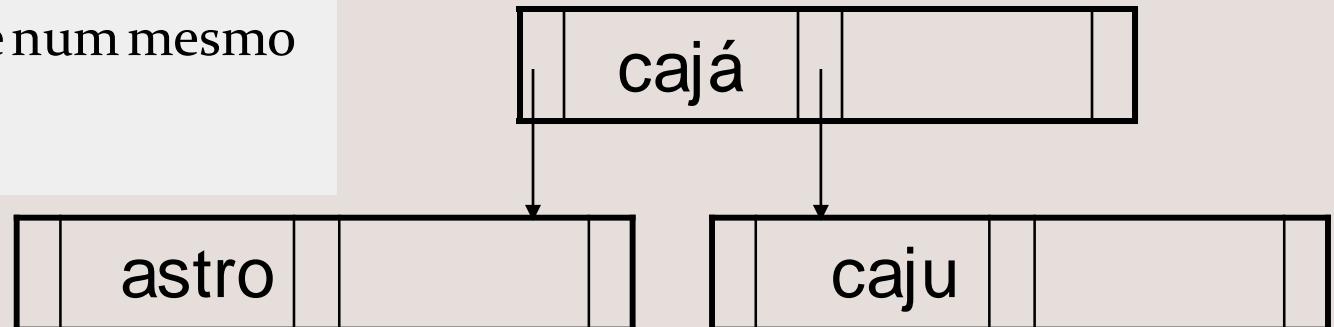
Árvore B

Numa árvore B de ordem x :

1. Todos os nós, exceto a raiz, devem armazenar de x a $2x$ dados.
2. A raiz deve armazenar de 1 a $2x$ dados.
3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
4. Todas as folhas devem apresentar-se num mesmo nível.



Sendo uma árvore B de ordem x , quantas entradas de dados deve ter cada nó desta, em relação a x ?



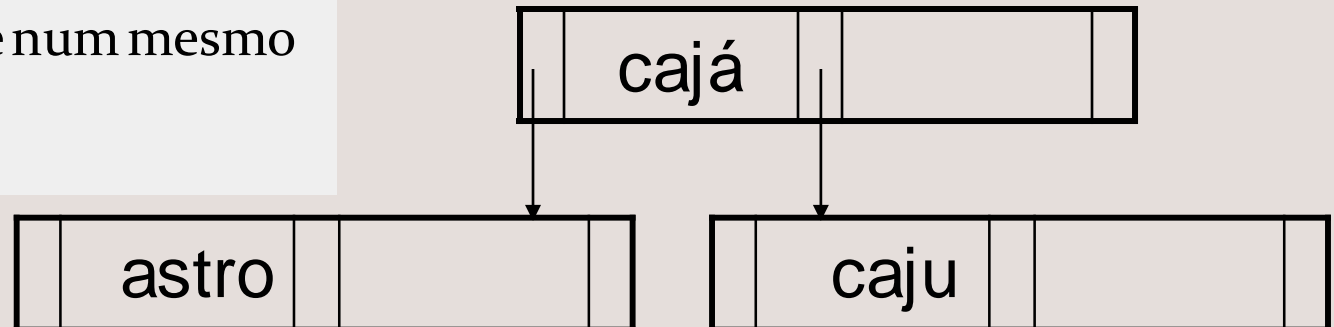
Árvore B

Numa árvore B de ordem x :

1. Todos os nós, exceto a raiz, devem armazenar de x a $2x$ dados.
2. A raiz deve armazenar de 1 a $2x$ dados.
3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
4. Todas as folhas devem apresentar-se num mesmo nível.



Sendo uma árvore B de ordem x , quantos apontadores deve ter cada nó desta, em relação a x ?

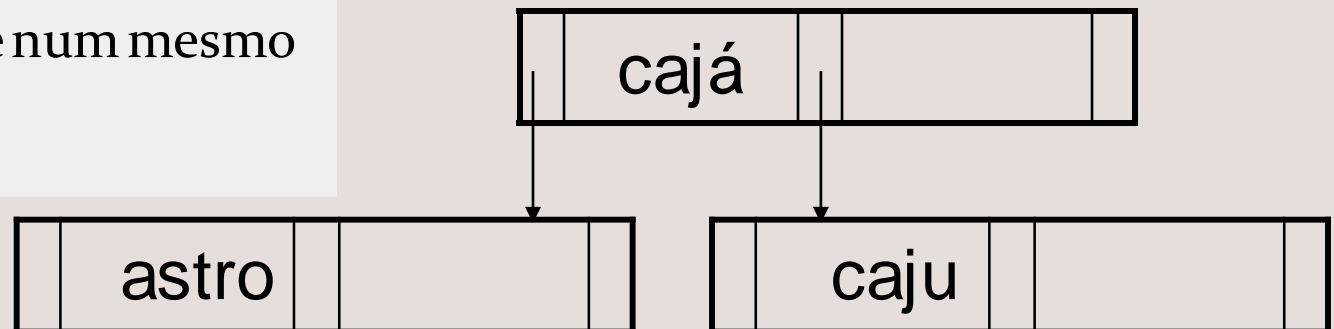


Árvore B

Numa árvore B de ordem x :

1. Todos os nós, exceto a raiz, devem armazenar de x a $2x$ dados.
2. A raiz deve armazenar de 1 a $2x$ dados.
3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
4. Todas as folhas devem apresentar-se num mesmo nível.

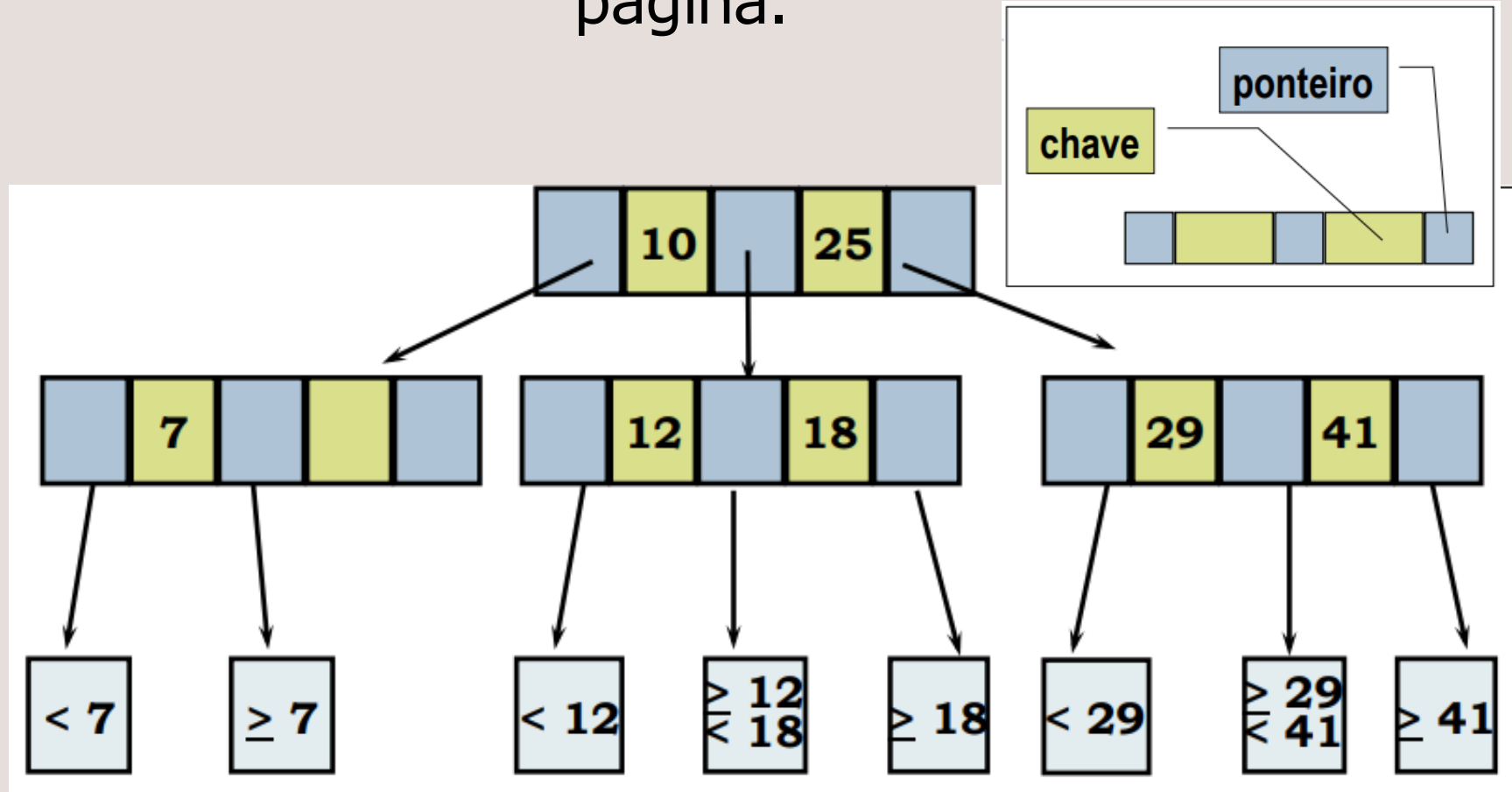
Importante observar que, sendo uma árvore B de ordem x , seus nós devem ser compostos por $2x$ entradas para dados e $2x+1$ apontadores.



Árvore B

CONSULTA

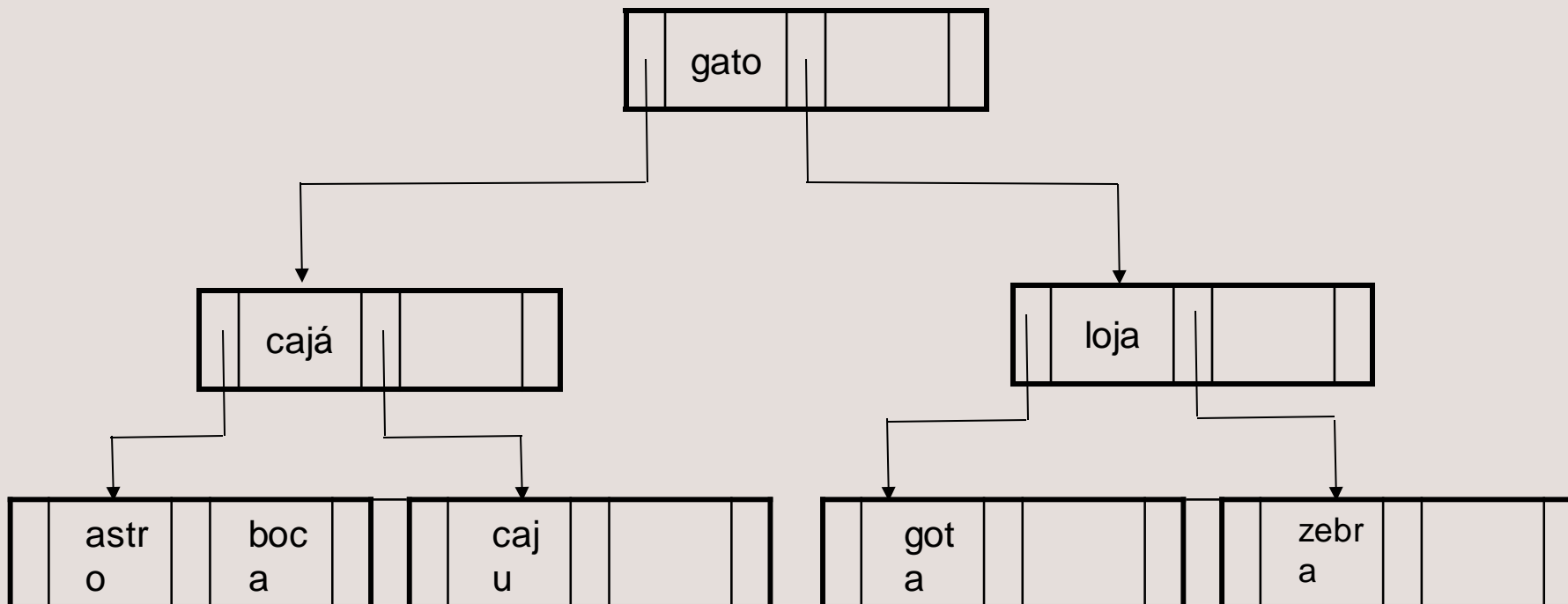
A **busca** em árvore B é semelhante à utilizada em árvore de pesquisa, acrescentando testes relativos às chaves existentes em cada página.



Árvore B

CONSULTA

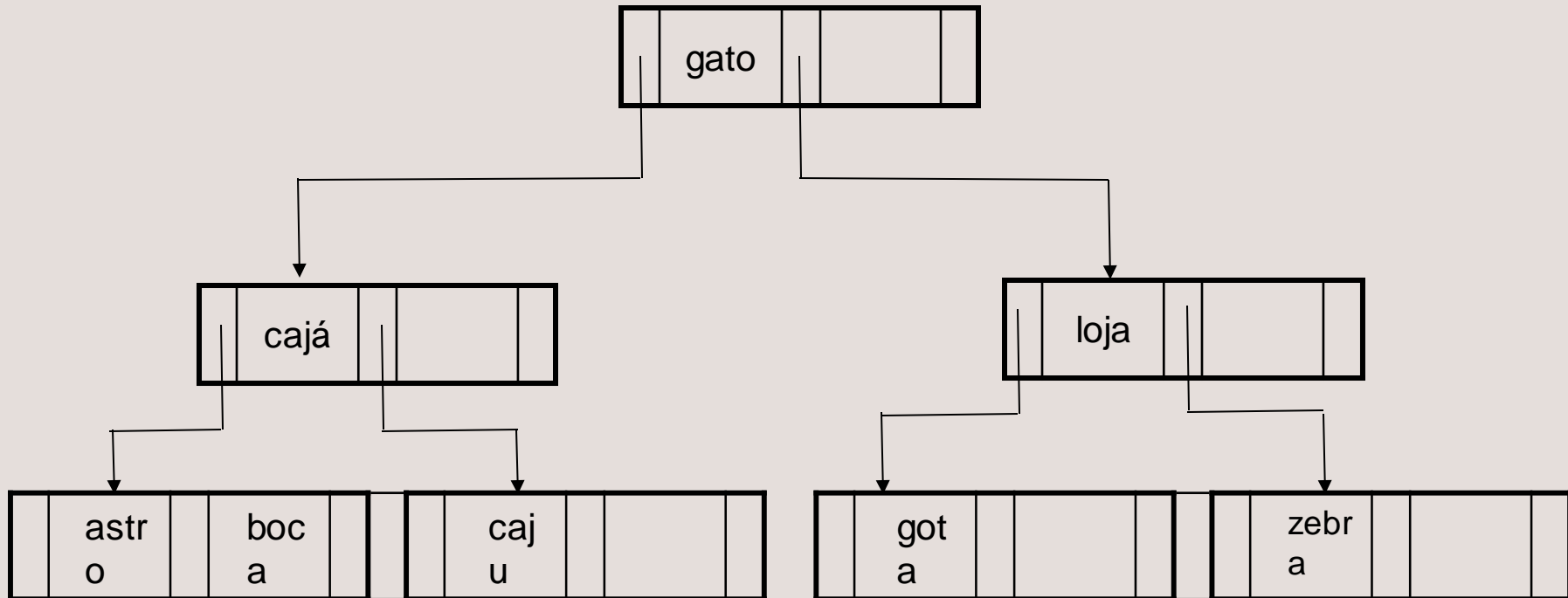
A **busca** em árvore B é semelhante à utilizada em árvore de pesquisa, acrescentando testes relativos às chaves existentes em cada página.



Árvore B

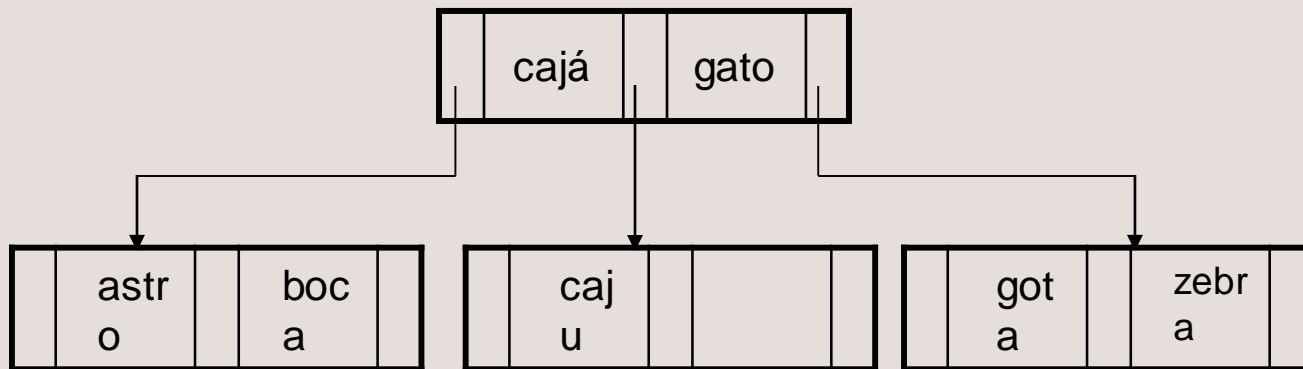
CONSULTA

Efetuar consulta às chaves zebra e bule na árvore B dada abaixo.



Árvore B

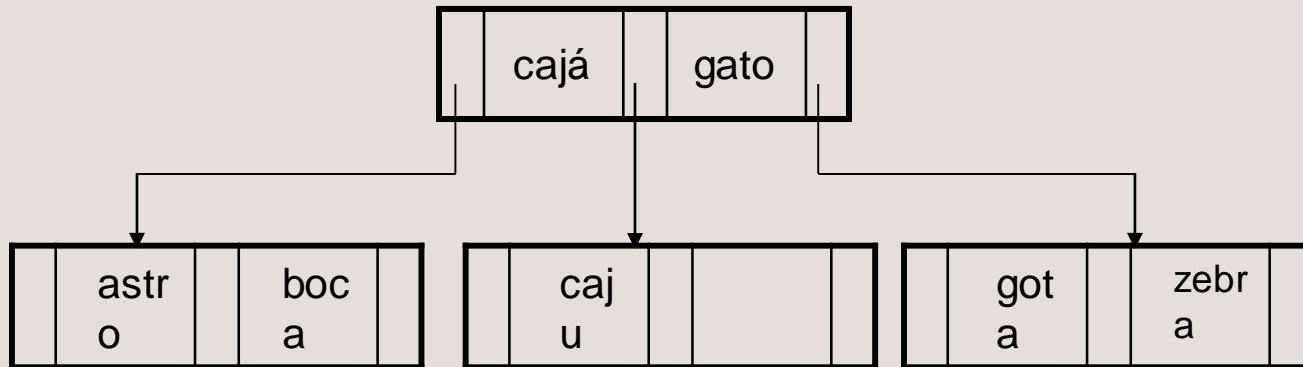
CONSULTA



Efetuar consulta
do dado zica na
árvore B dada:

Árvore B

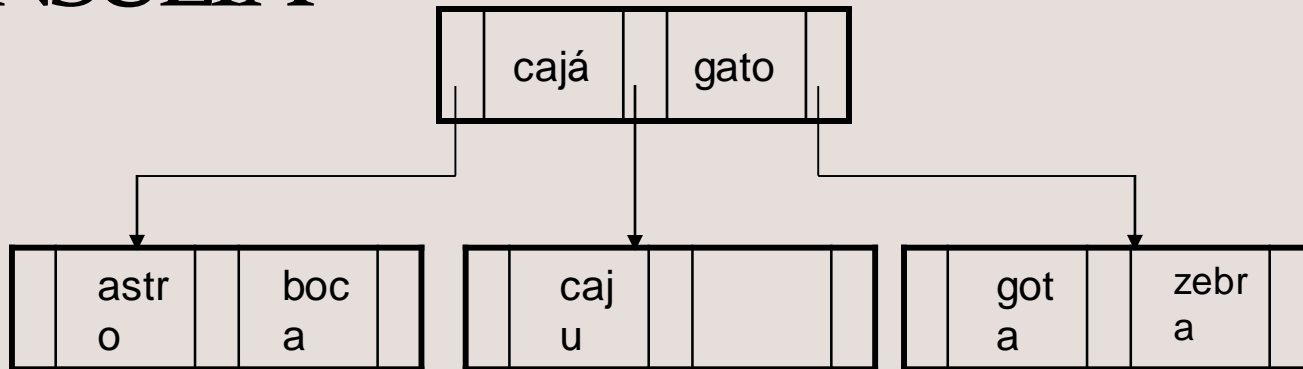
CONSULTA



Efetuar consulta
da chave asas na
árvore B dada:

Árvore B

CONSULTA



Segundo Szwarcfiter e Markeson (2015), o passo de maior custo na execução da operação de consulta em Árvore B é o acesso à memória secundária. Assim sendo, o melhor caso ocorre quando a chave é encontrada na página raiz – $O(k)$ e o pior caso, quando chave encontrada em folha $O(h)$, sendo h a altura da árvore.

A **altura h** de uma árvore T é o maior comprimento contido em T , ou seja, o maior número de aresta do caminho mais longo entre a raiz e uma folha.

Árvore B

INSERÇÃO



É possível compreender o processo de inserção de dados em árvores B através da construção, por exemplo, de uma árvore de ordem 1 composta por: *caju*¹, *cajá*², *astro*³, *boca*⁴, *loja*⁵, *gato*⁶, *gota*⁷, *zebra*⁸.

Árvore B

INSERÇÃO

1

	caju			
--	------	--	--	--

Inserindo *cajá*:

2

	cajá		caju	
--	------	--	------	--

Há o deslocamento do primeiro valor para manter a estrutura ordenada.

Árvore B

INSERÇÃO

2

	cajá		caju	
--	------	--	------	--

Inserindo *astro* :

3

	cajá		caju

astro

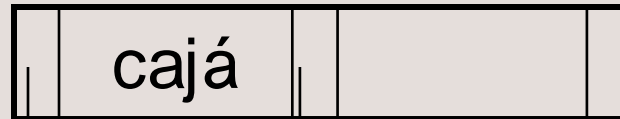
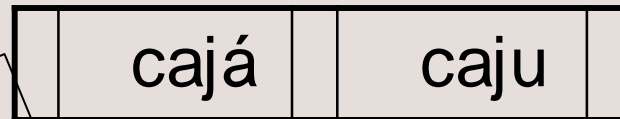
Observa-se o estouro da capacidade de armazenamento do nó (overflow acompanhado de cisão) implicando na necessidade de crescimento da estrutura.

Árvore B

INSERÇÃO

3

astro



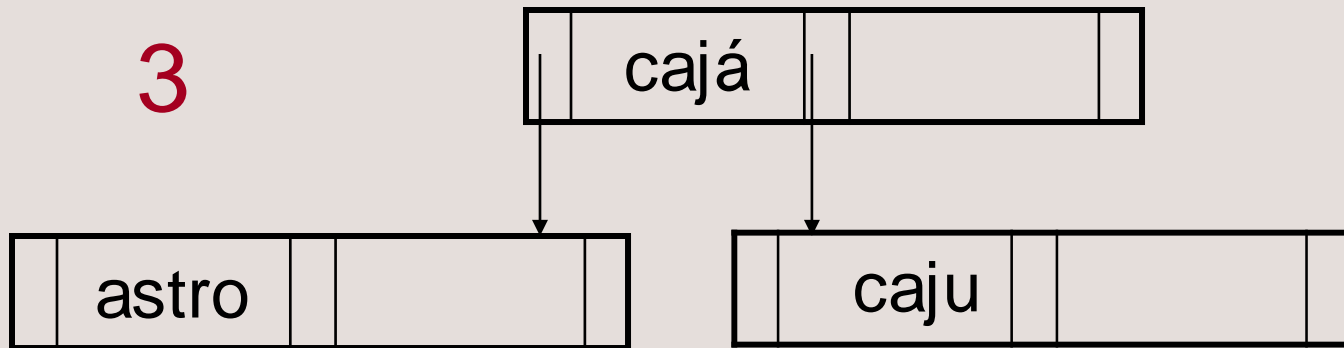
Houve estouro da capacidade de armazenamento do nó e consequente crescimento *bottom up* da estrutura.

Árvore B

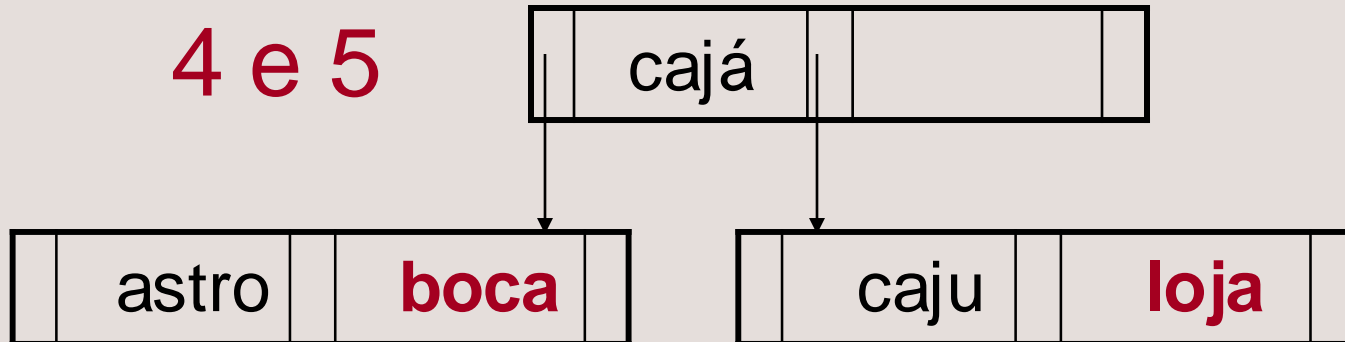
INSERÇÃO

Inserindo *boca* e *loja*:

3



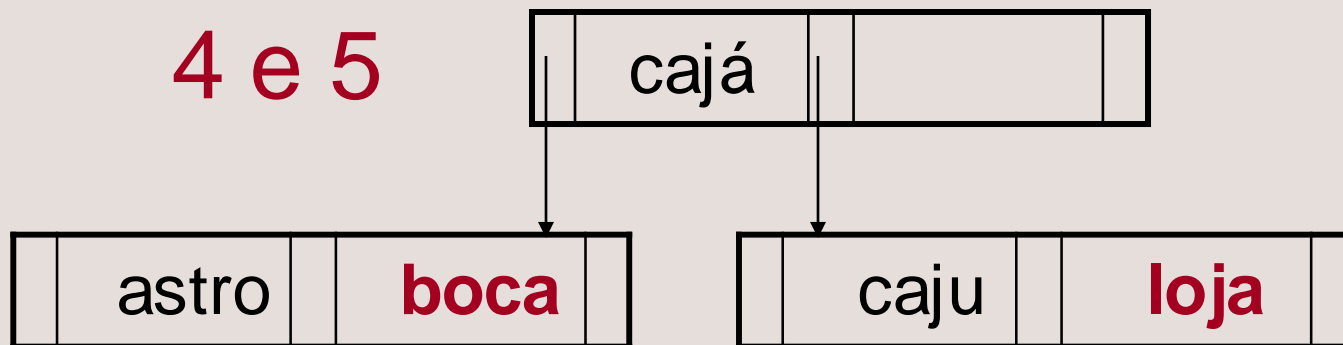
4 e 5



Árvore B

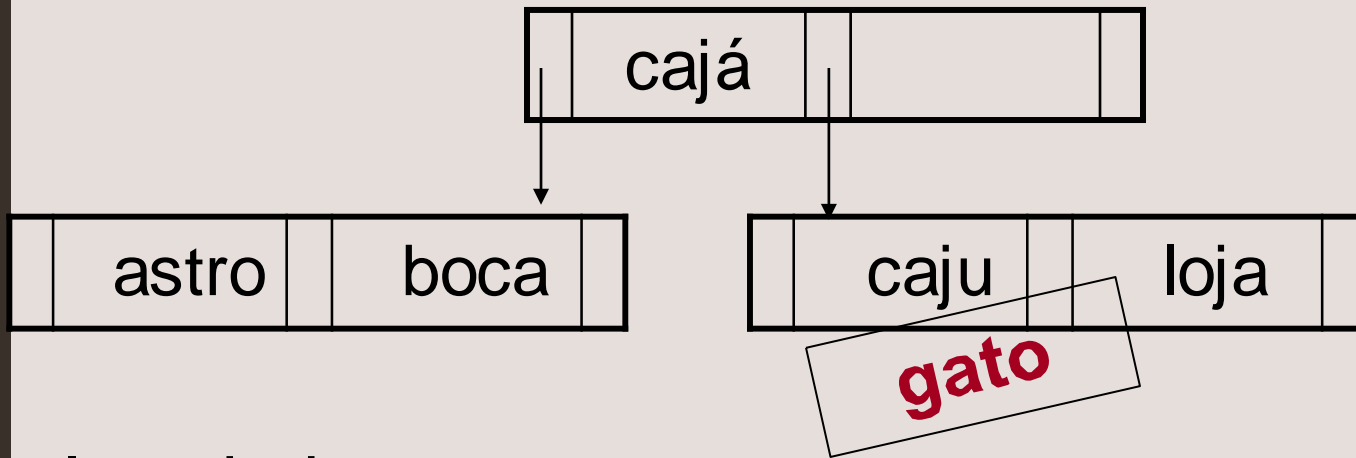
INSERÇÃO

Na inclusão de *boca*, compara-se esta com a raiz, sendo menor, segue-se pela esquerda. Compara-se a nova palavra com o valor encontrado: *astro*, sendo maior, segue para direita, havendo espaço, faz-se a inclusão.



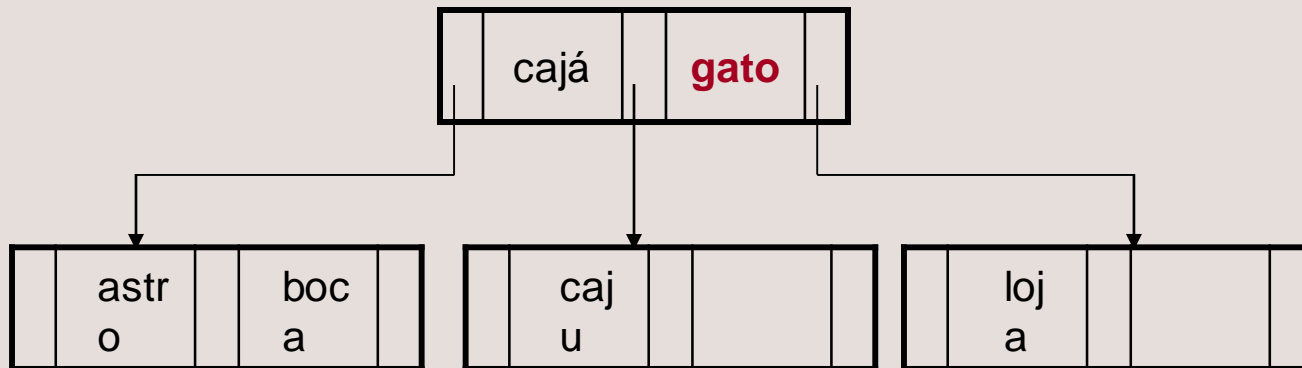
Árvore B

INSERÇÃO



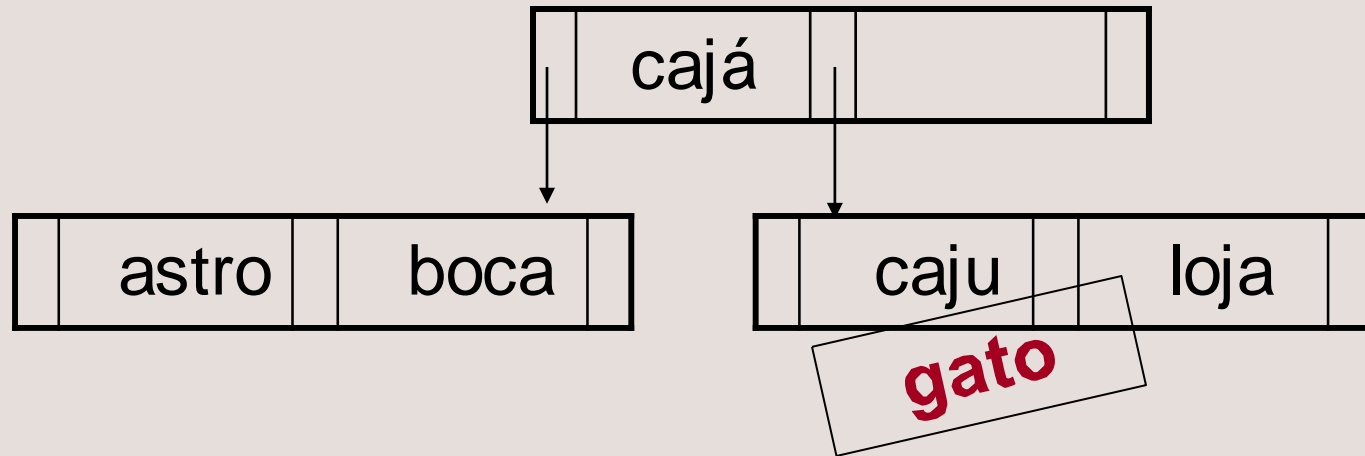
Inserindo *gato*:

6



Árvore B

INSERÇÃO

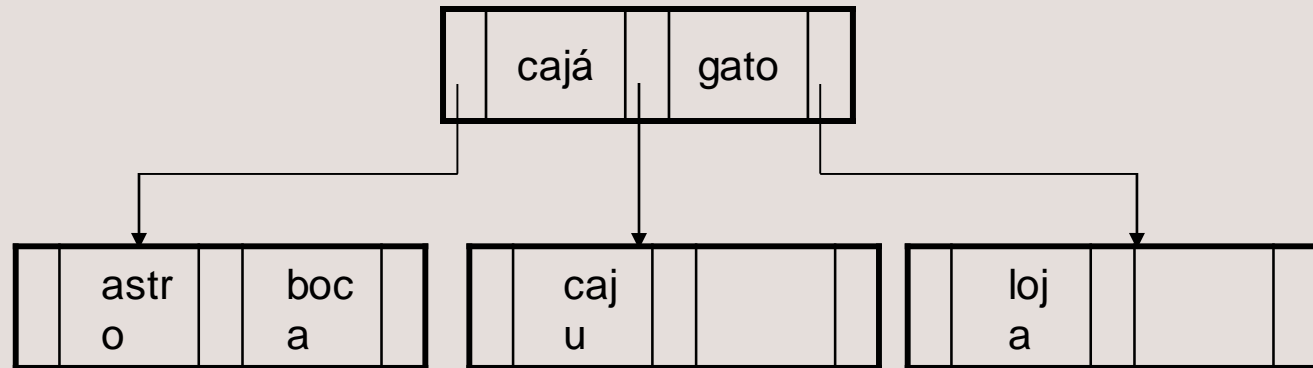


Há o estouro (overflow seguido de cisão) da capacidade de armazenamento do nó composto por *caju* || *loja*, então o elemento mediano *gato* é remetido ao nível ascendente – crescimento bottom up.

Árvore B

INSERÇÃO

6

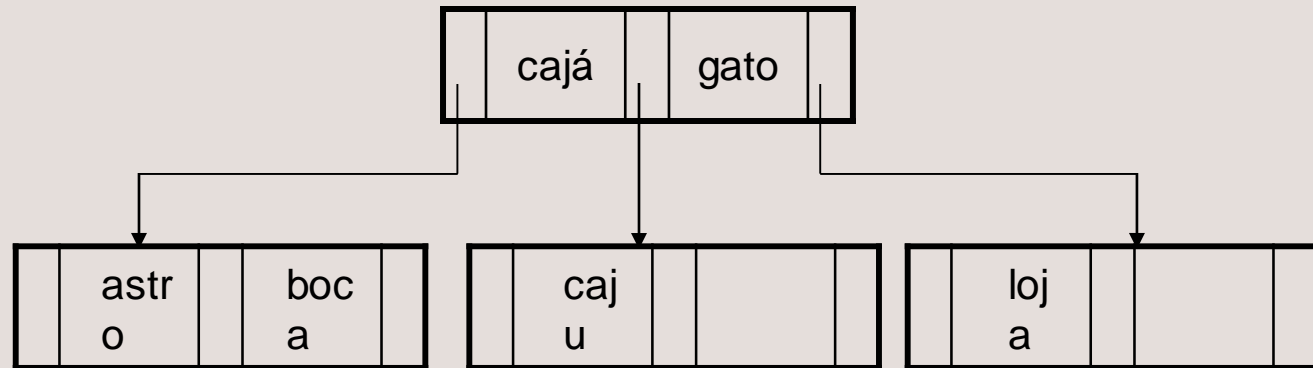


Como efetuar a
inclusão de
gota?

Árvore B

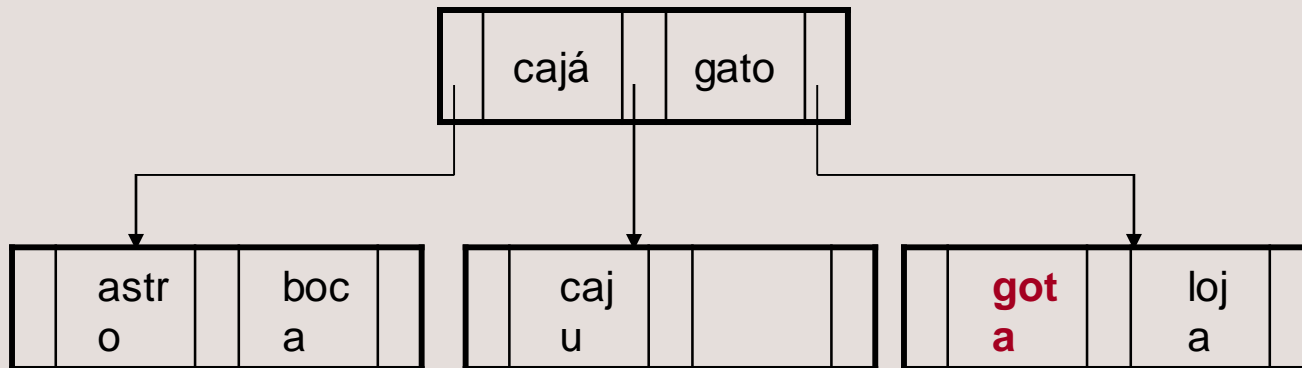
INSERÇÃO

6



Inserindo *gota*:

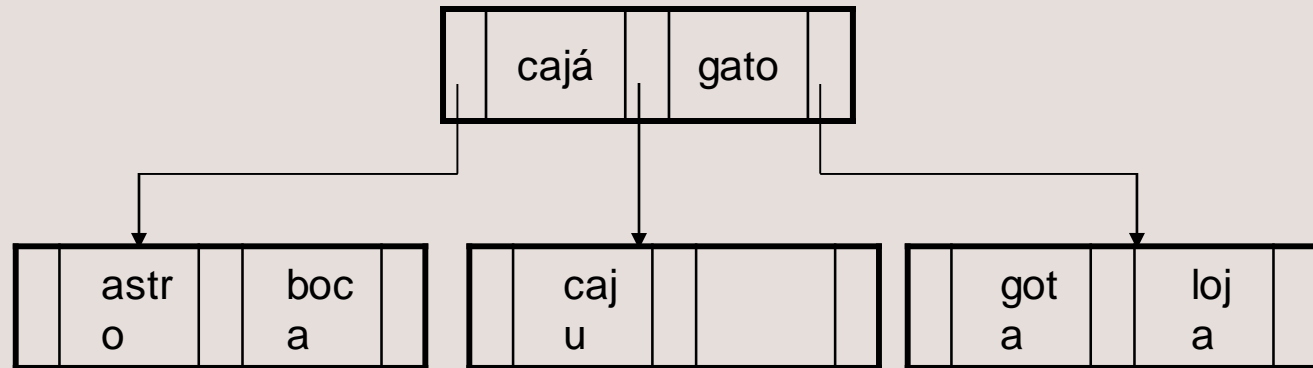
7



Árvore B

INSERÇÃO

6

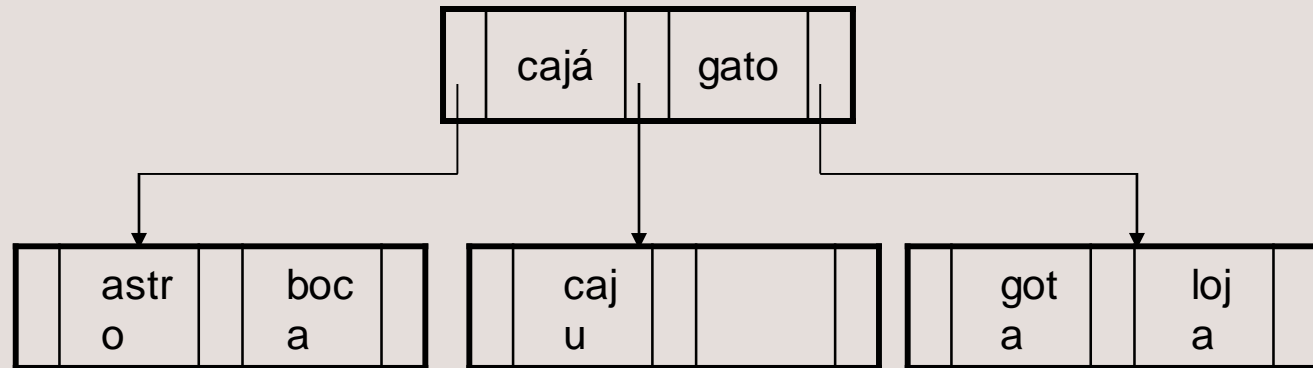


Como efetuar a
inclusão de
zebra?

Árvore B

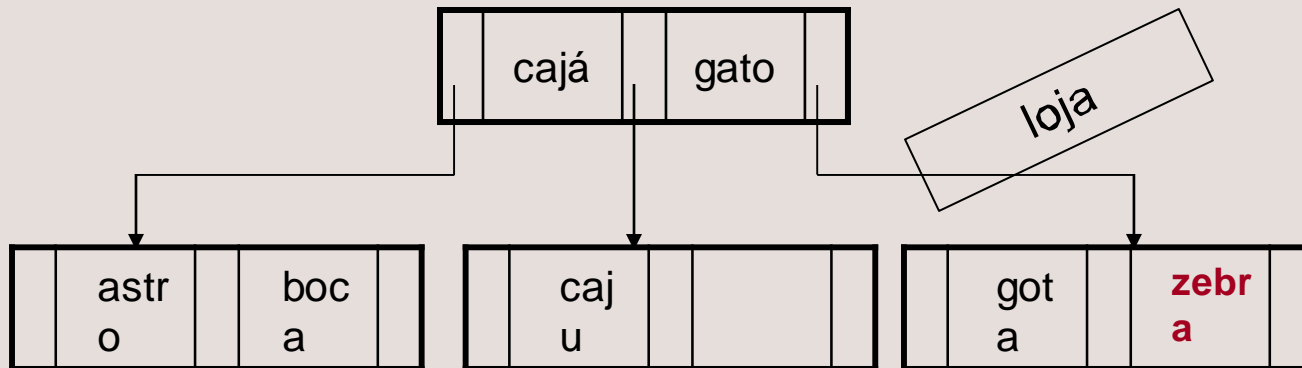
INSERÇÃO

6



Inserindo *zebra*:

7

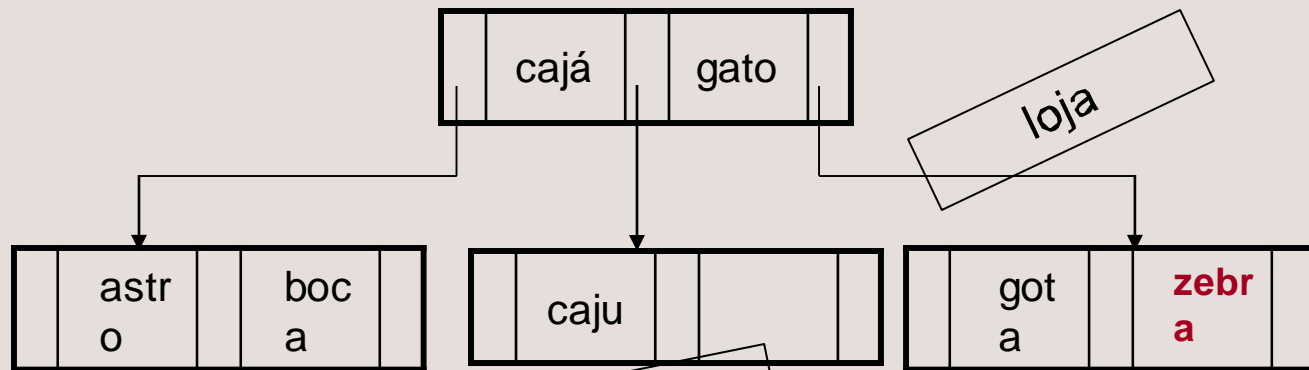


Árvore B

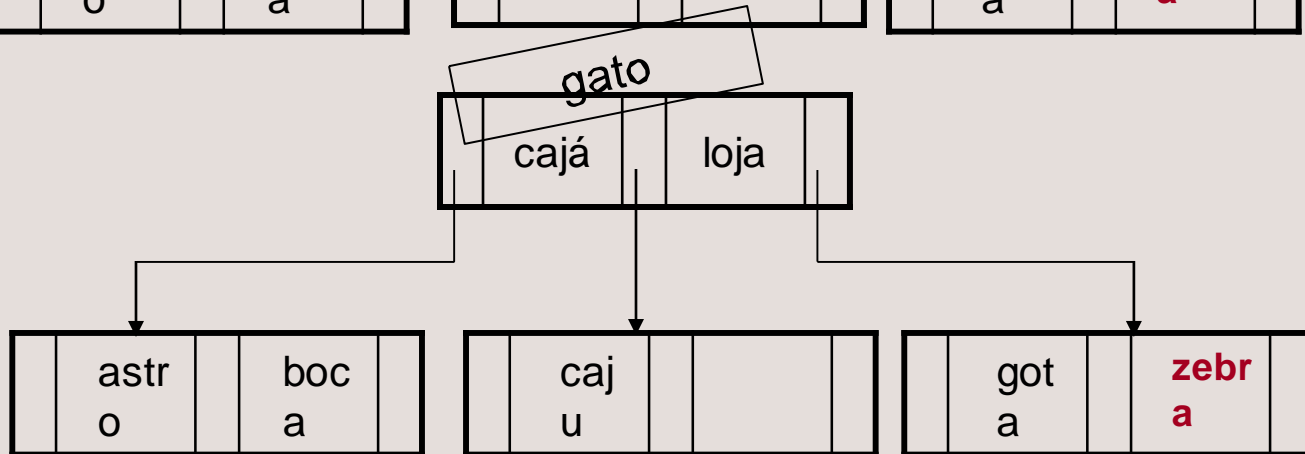
INSERÇÃO

Inserindo *zebra*:

7



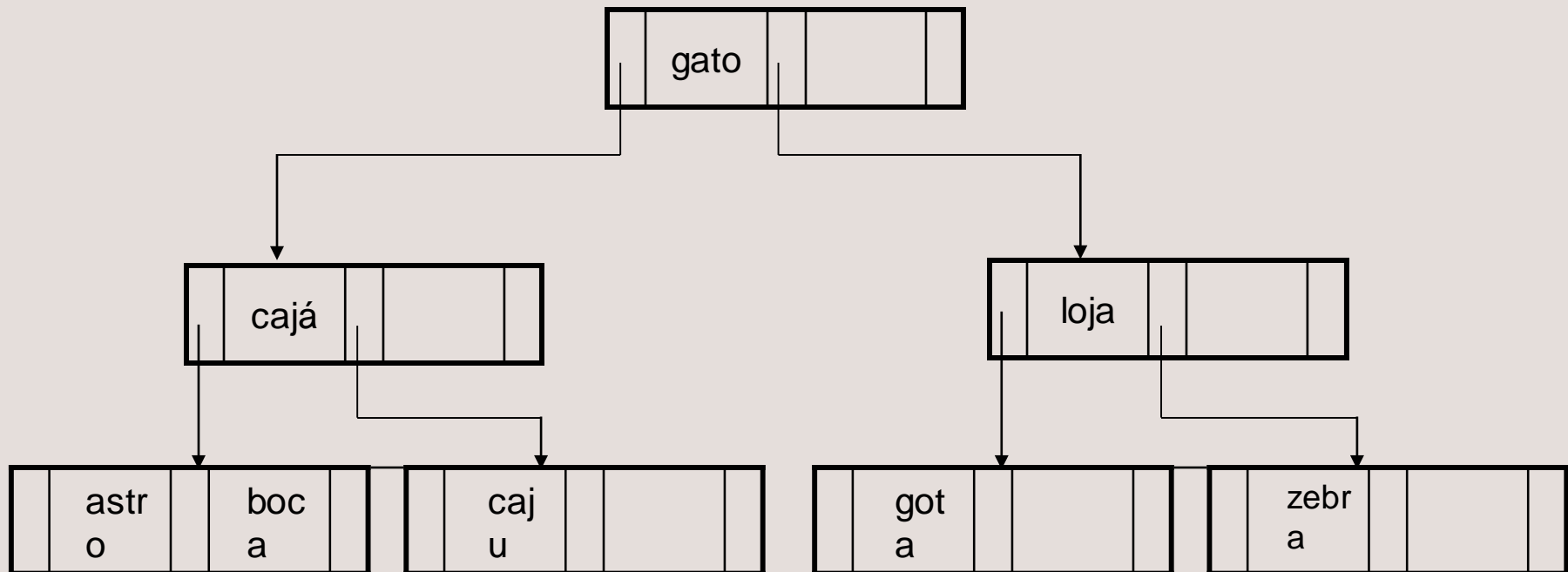
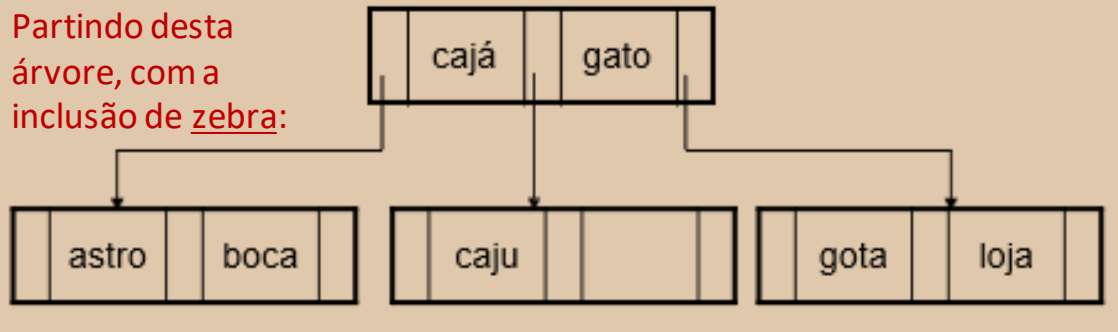
7



Árvore B

INSERÇÃO

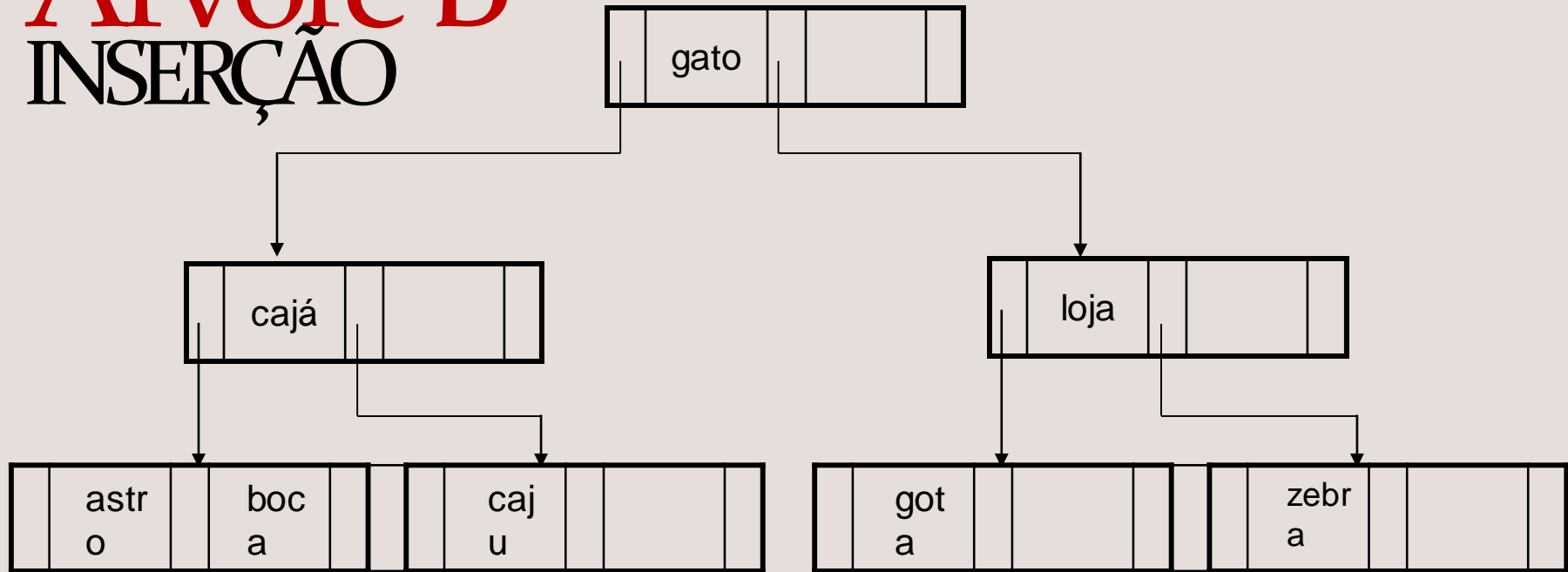
Partindo desta
árvore, com a
inclusão de zebra:



O estouro do nó *gota||loja* provoca o estouro do
nó ascendente *cajá||gato*.

Árvore B

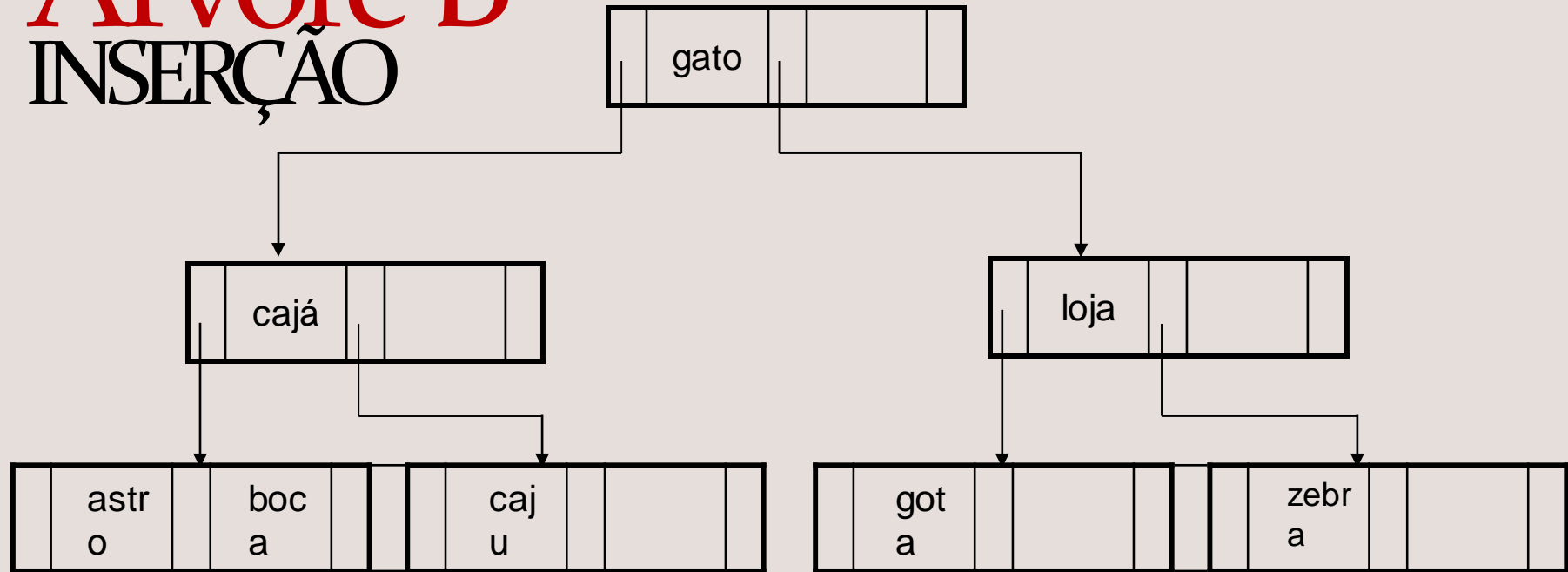
INSERÇÃO



Qual o custo da
operação de
inserção em
árvore B?

Árvore B

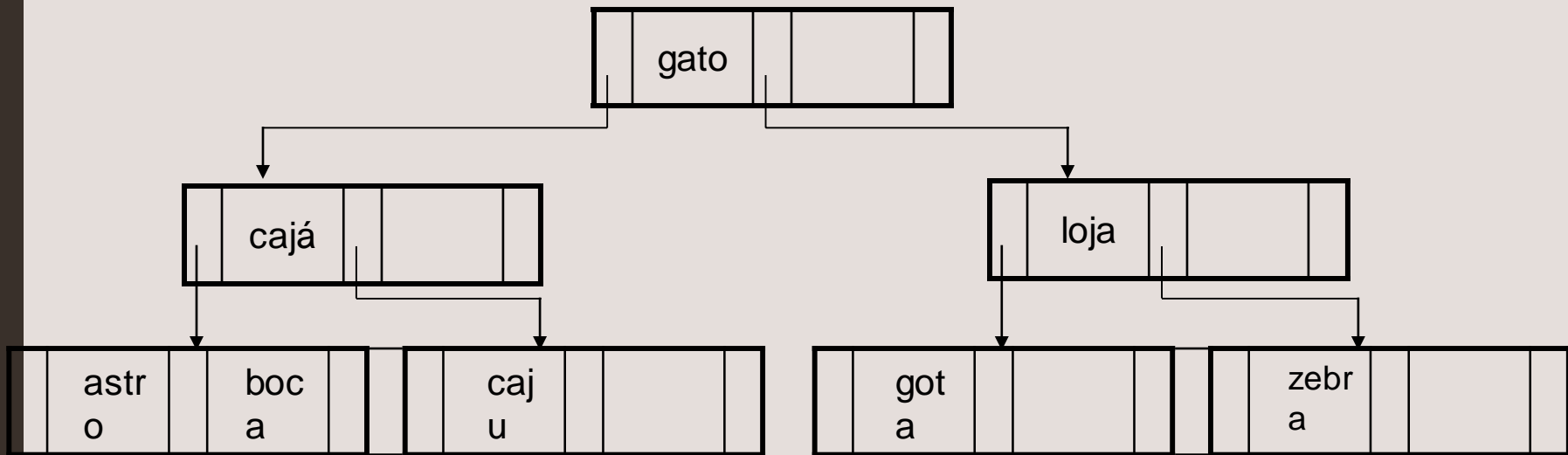
INSERÇÃO



A inserção sem overflow requer $h+1$ acessos à memória secundária e a inserção em que há cisão em todas as páginas do caminho da folha (ponto de inserção) até a raiz requer $2h + 1$ acessos à memória secundária, logo o custo da operação de inserção é $O(h)$.

Árvore B

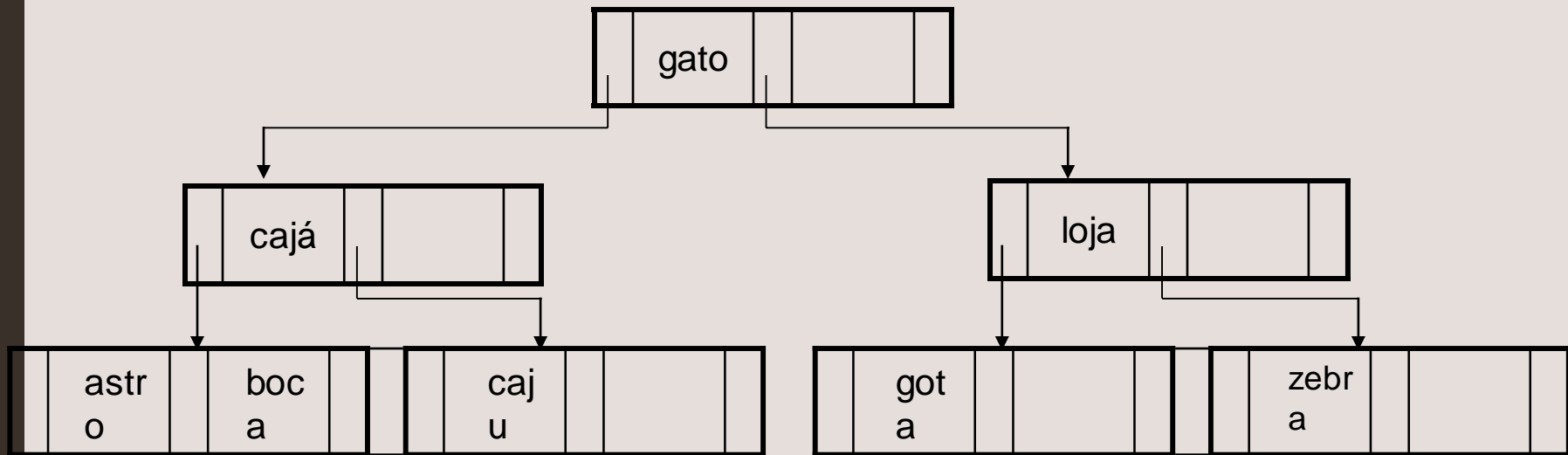
EXCLUSÃO



Na exclusão deve-se preservar as propriedades das árvores B e para tanto pode ser necessário concatenar nós (páginas) ou redistribuir os dados nestes.

Árvore B

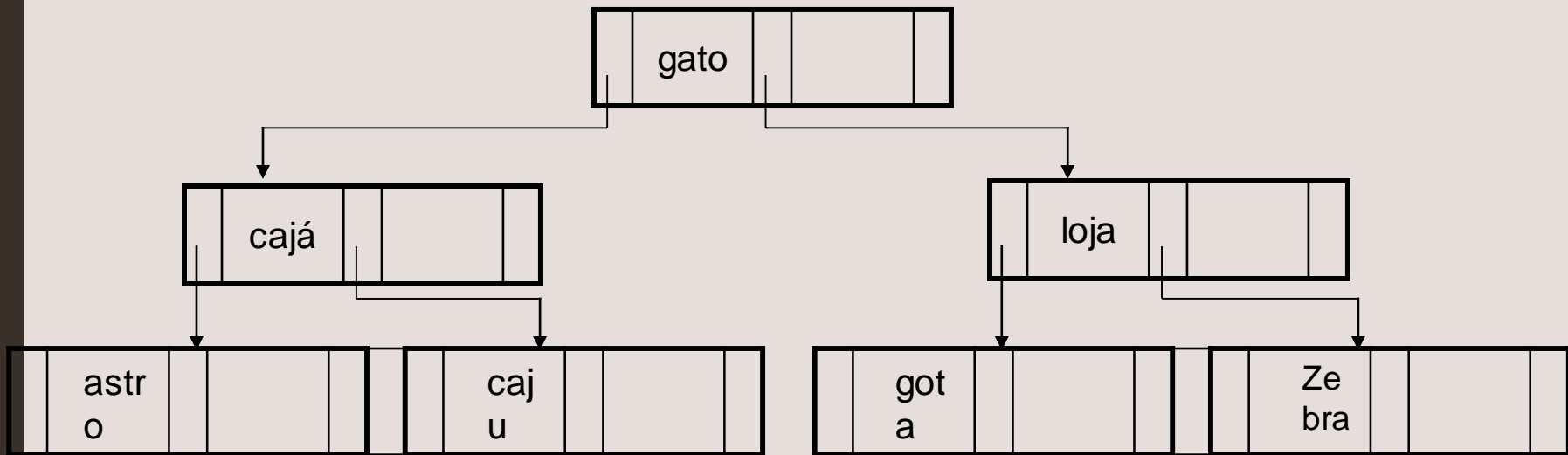
EXCLUSÃO



Como efetuar a
exclusão do dado
boca na árvore dada?

Árvore B

EXCLUSÃO

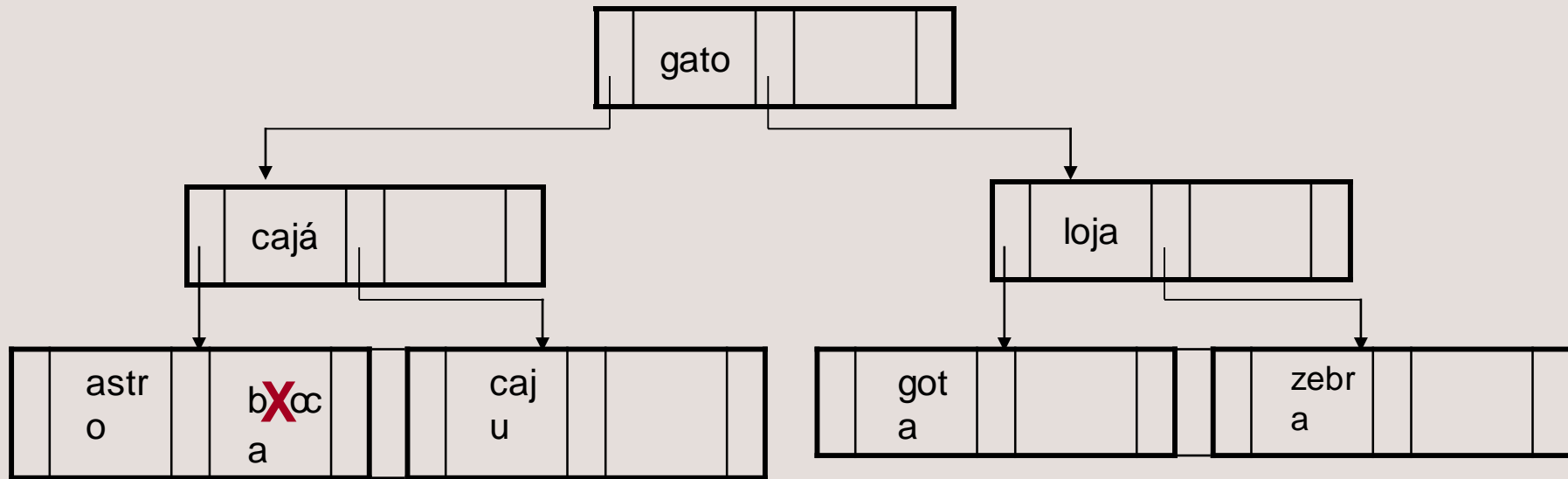


Como efetuar a exclusão do dado ***gato*** na árvore dada?

Árvore B

EXCLUSÃO

Excluindo *boca*:

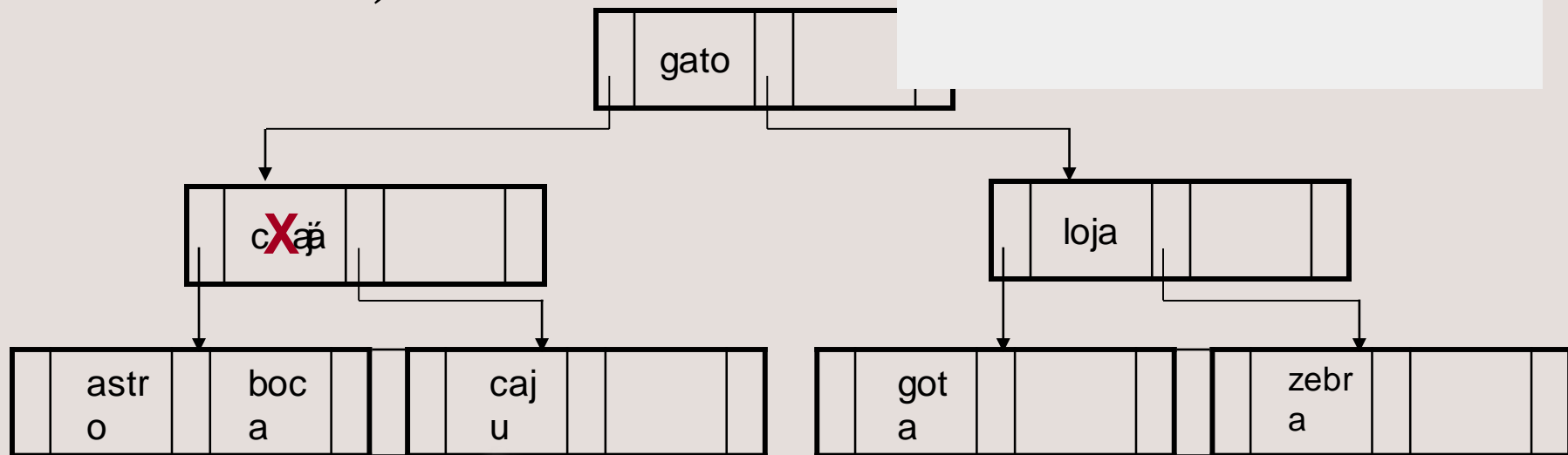


Este constitui um caso trivial, não provoca nenhuma alteração na estrutura da árvore.

Árvore B

EXCLUSÃO

Excluindo *cajá*:



Numa árvore B de ordem x :

1. Todos os nós, exceto a raiz, devem armazenar de x a $2x$ dados.
2. A raiz deve armazenar de 1 a $2x$ dados.
3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
4. Todas as folhas devem apresentar-se num mesmo nível.

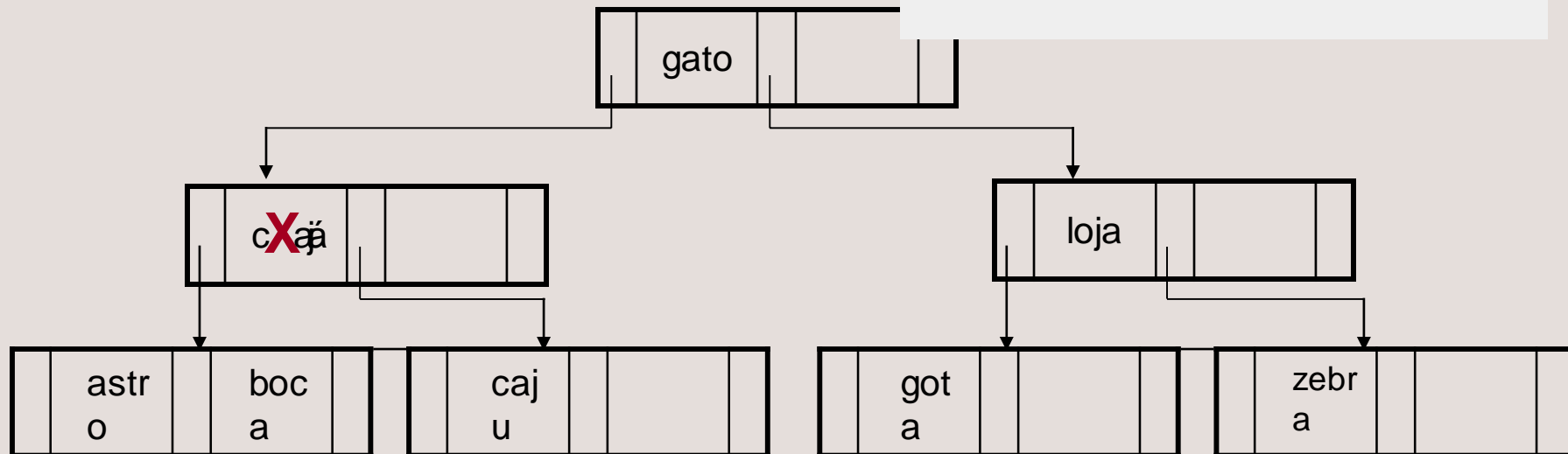


Como efetuar a exclusão do dado ***cajá*** na árvore dada?

Árvore B

EXCLUSÃO

Excluindo *cajá*:



Numa árvore B de ordem x :

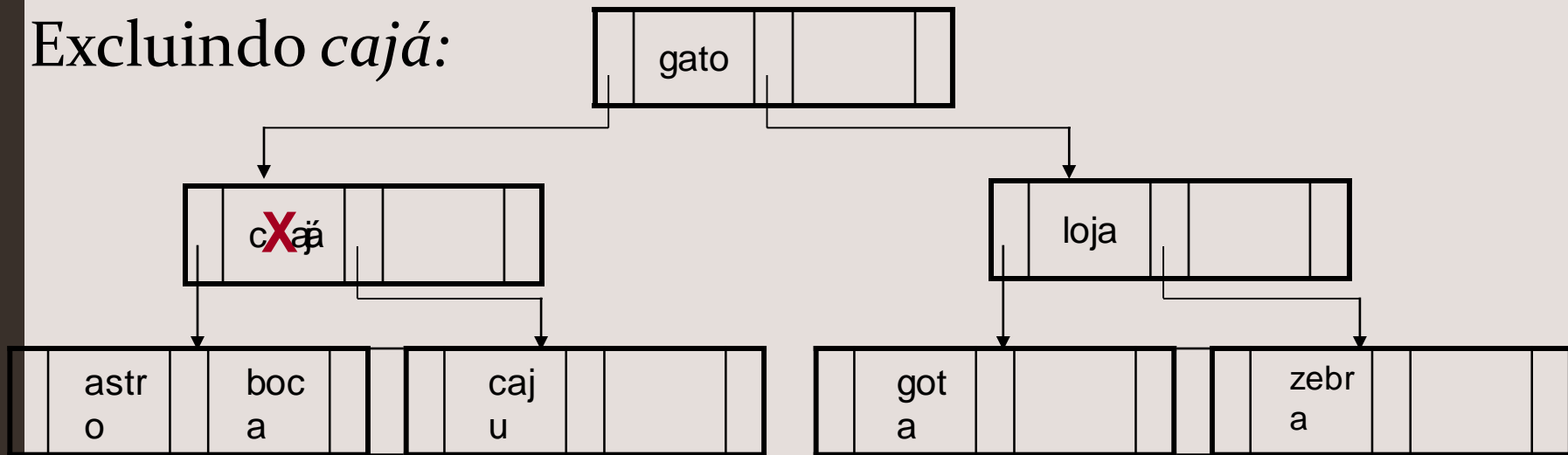
1. Todos os nós, exceto a raiz, devem armazenar de x a $2x$ dados.
2. A raiz deve armazenar de 1 a $2x$ dados.
3. Todo nó ou é folha ou tem $k+1$ descendentes, onde k é o número de dados mantidos no nó.
4. Todas as folhas devem apresentar-se num mesmo nível.

A simples remoção da palavra violaria a definição de árvore B. Então este é substituído pelo seu antecessor imediato – o nó mais à direita do ramo da esquerda.

Árvore B

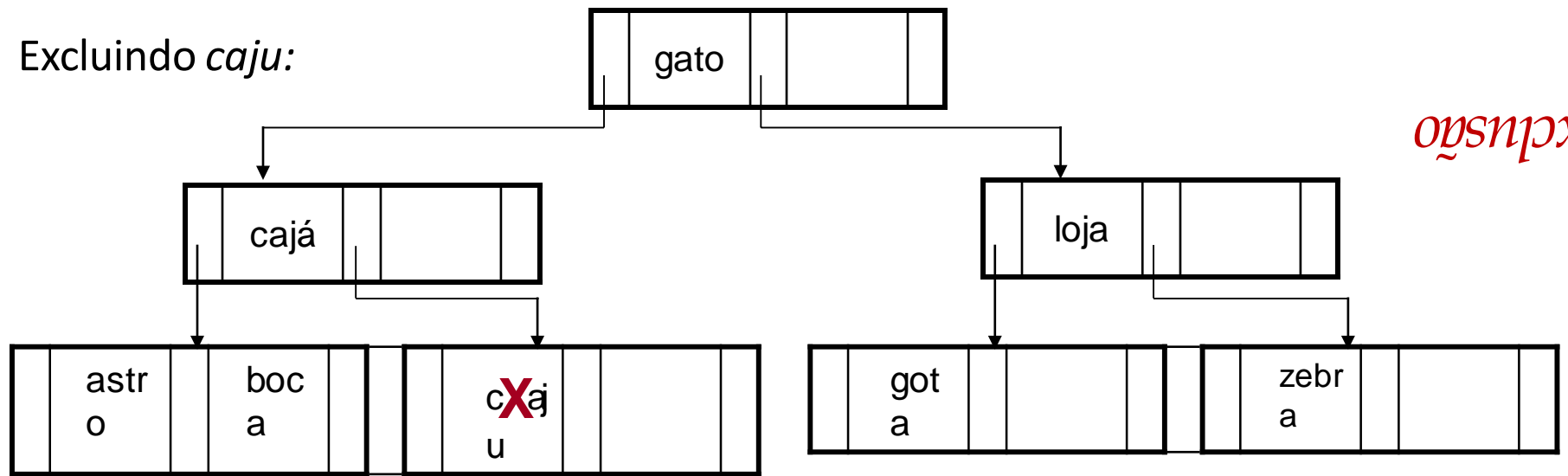
EXCLUSÃO

Excluindo *cajá*:

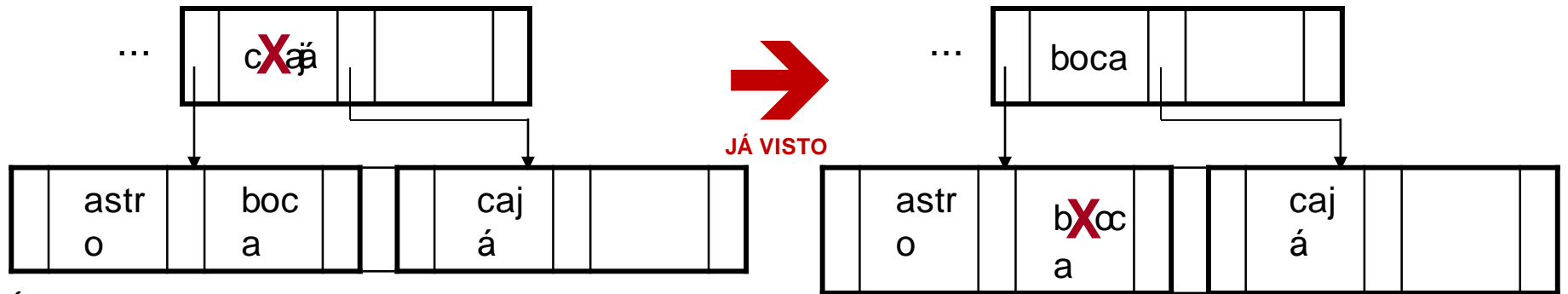


Somente o antecessor imediato pode substituir um nó excluído?

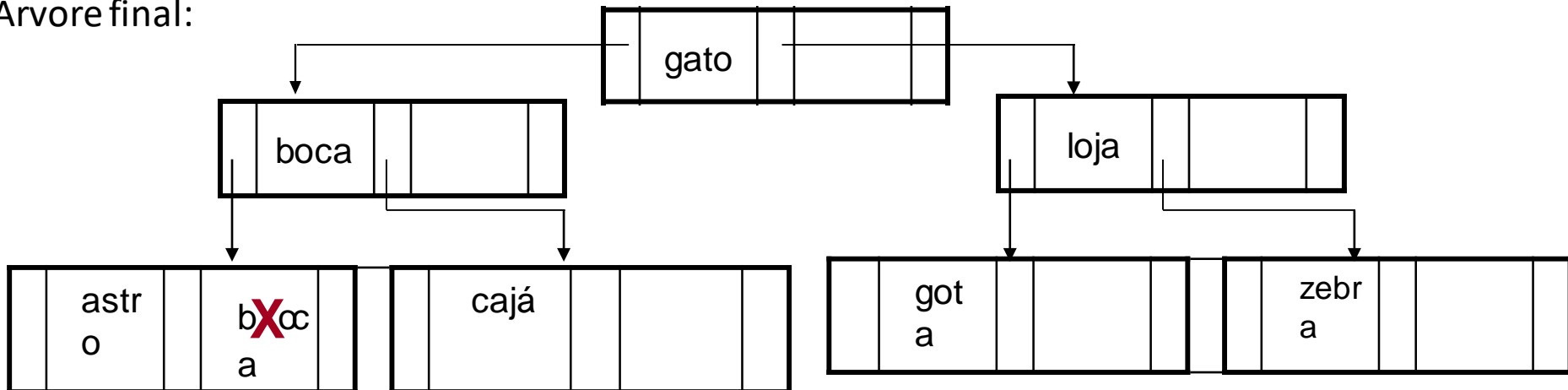
Excluindo *caju*:



Neste, o elemento não tem antecessor imediato, então faz-se uso do nó ascendente: *cajá*.



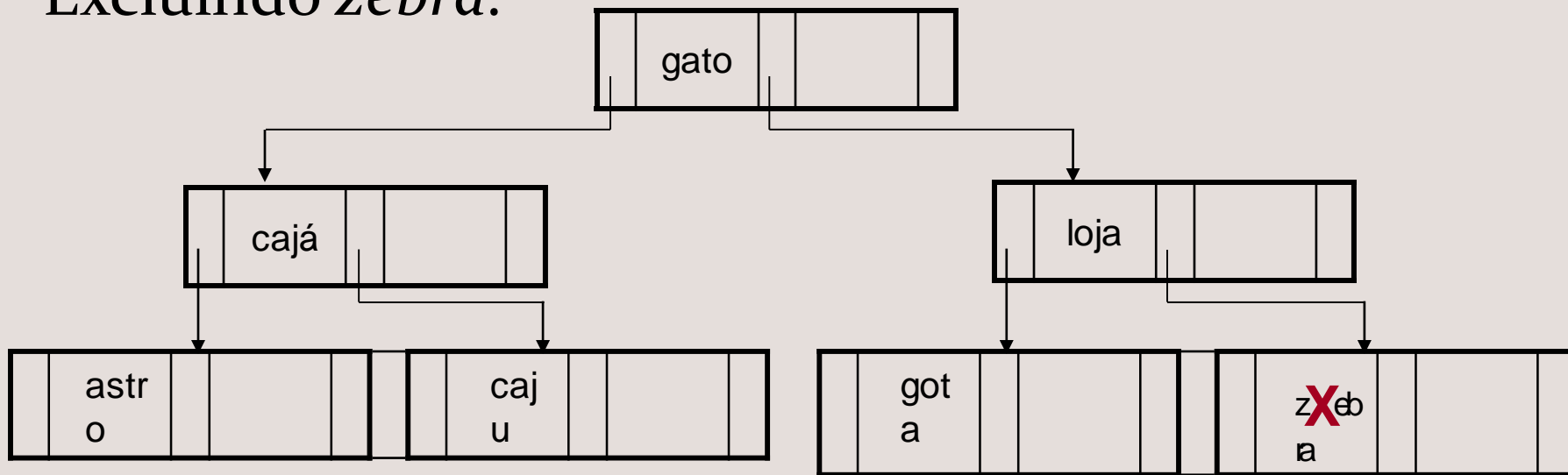
Árvore final:



Árvore B

EXCLUSÃO

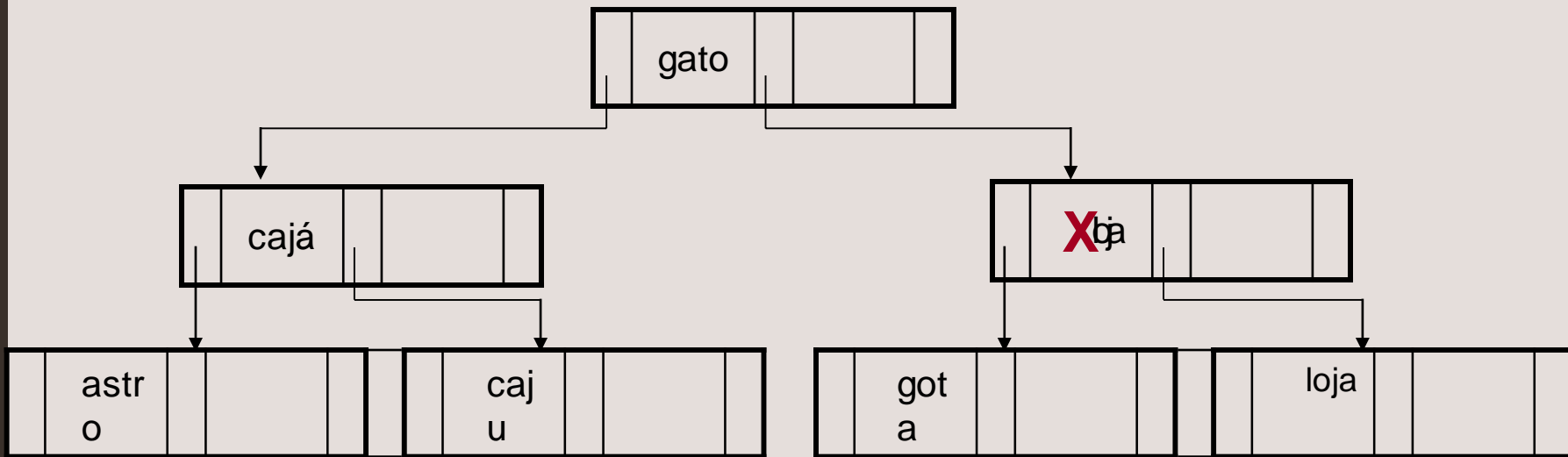
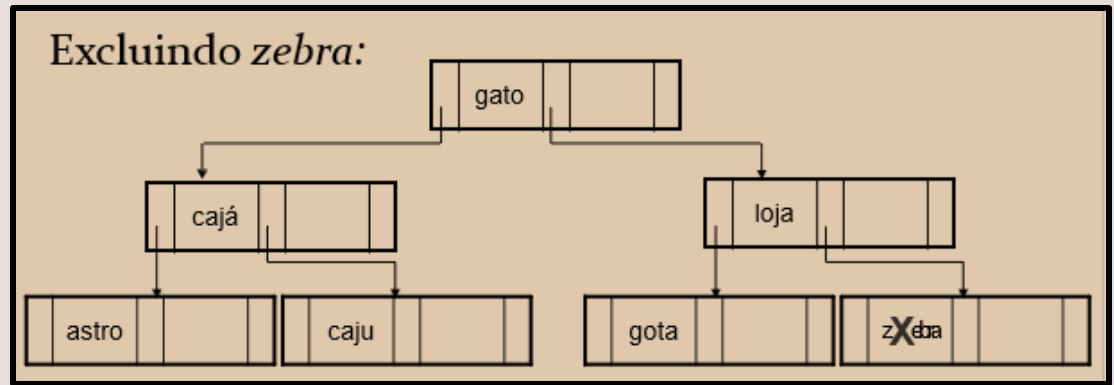
Excluindo *zebra*:



Neste caso o elemento não tem antecessor imediato, então faz-se uso do nó ascendente, no caso: *loja*.

Árvore B EXCLUSÃO

Excluindo zebra:

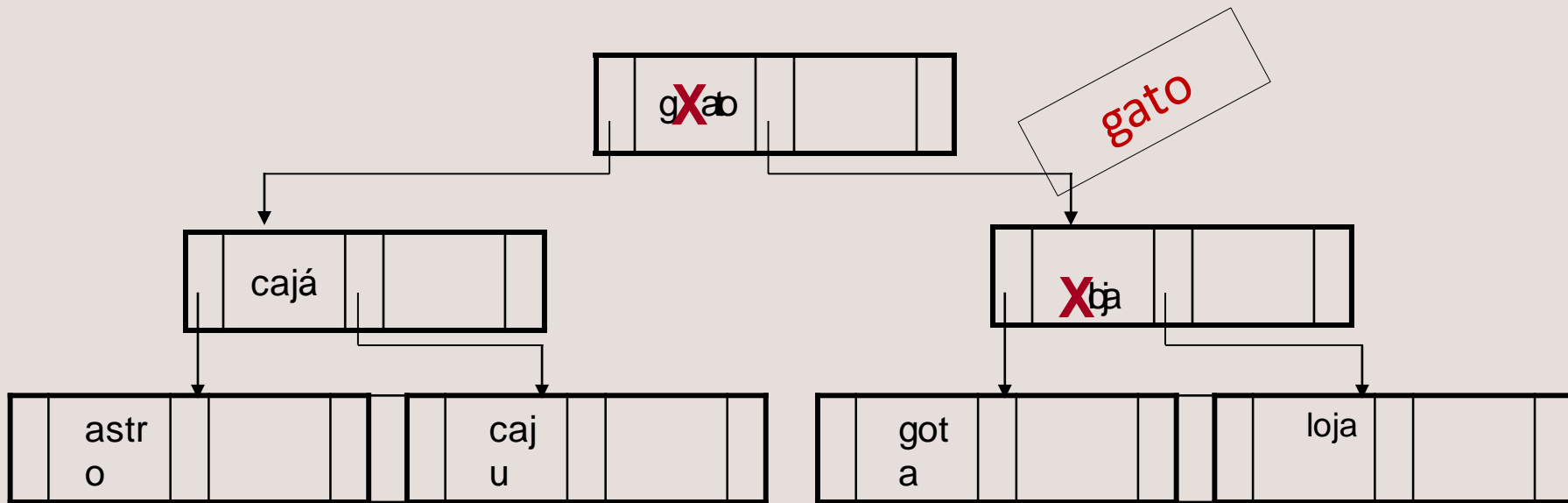


Este elemento – *loja* – também não tem antecessor imediato, então faz-se uso do nó ascendente, no caso: *gato*.

Árvore B

EXCLUSÃO

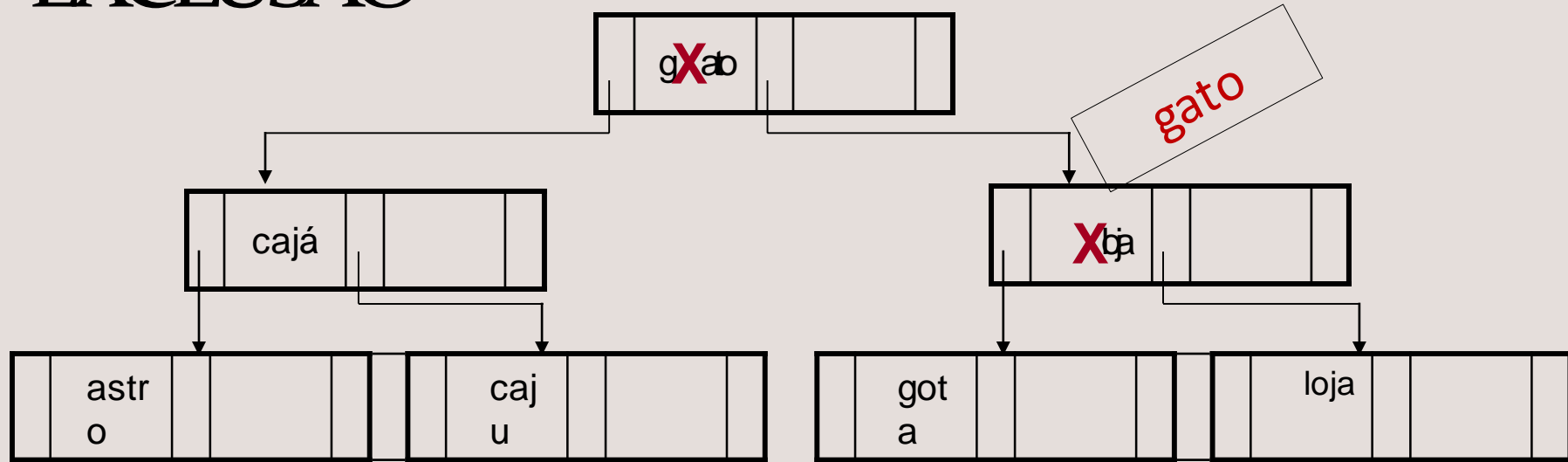
Excluindo *zebra*:



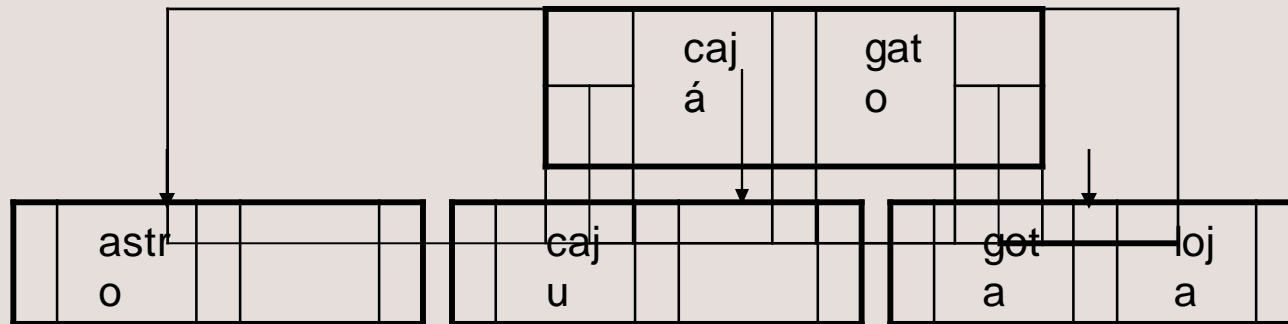
Este elemento – *gato* – não tem antecessor imediato nem nó ascendente, então há a combinação/concatenação de nós.

Árvore B

EXCLUSÃO

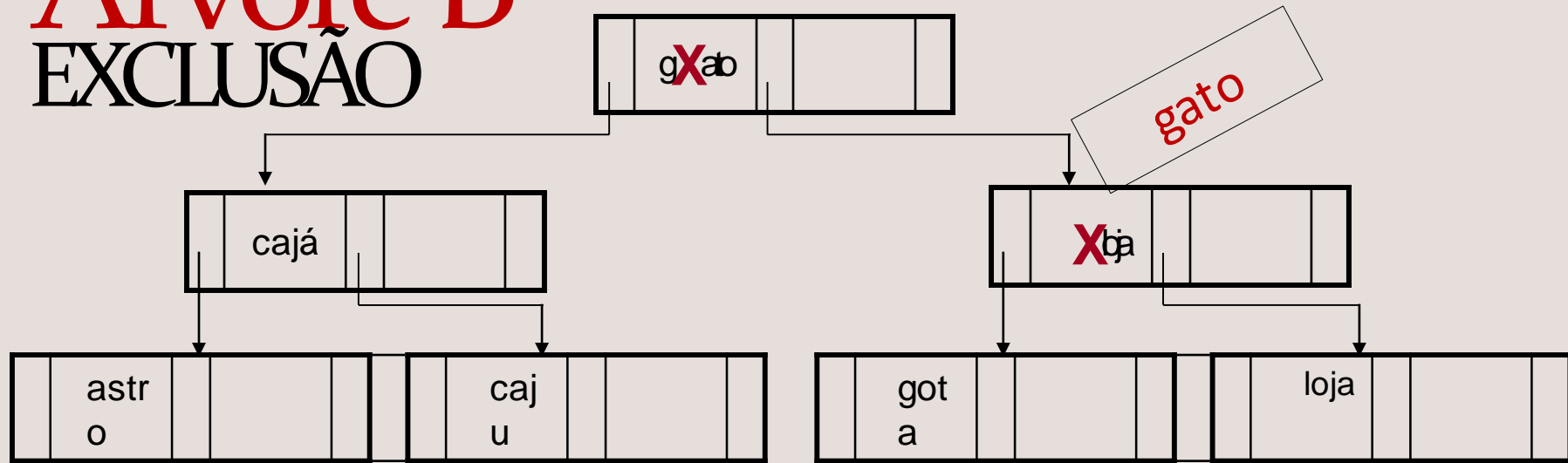


Ocorre a concatenação de nós:

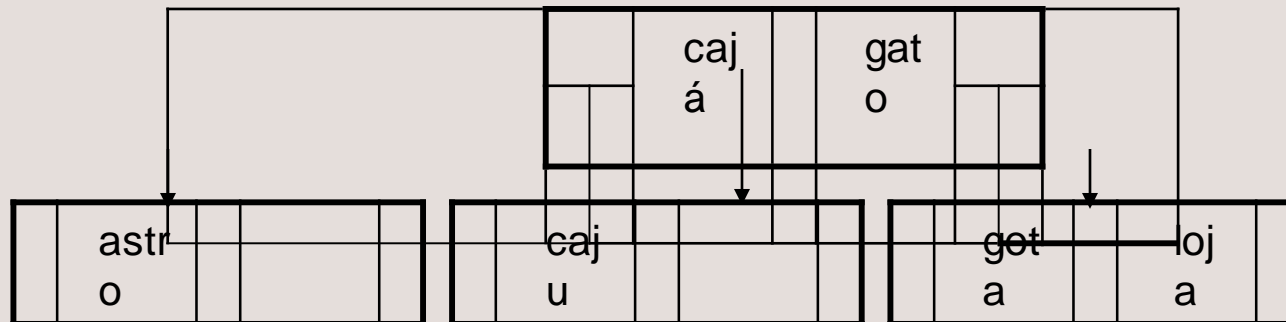


Árvore B

EXCLUSÃO



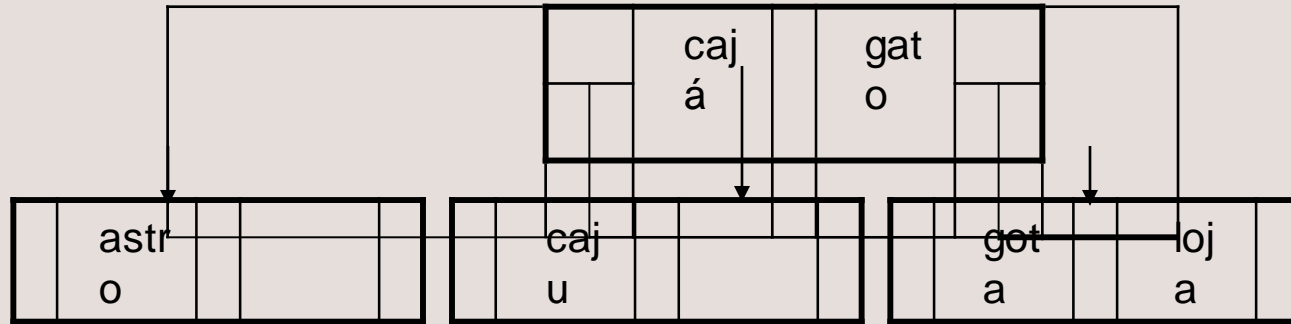
Ocorre a concatenação de nós:



Qual o custo da operação de exclusão em árvore B?

Árvore B

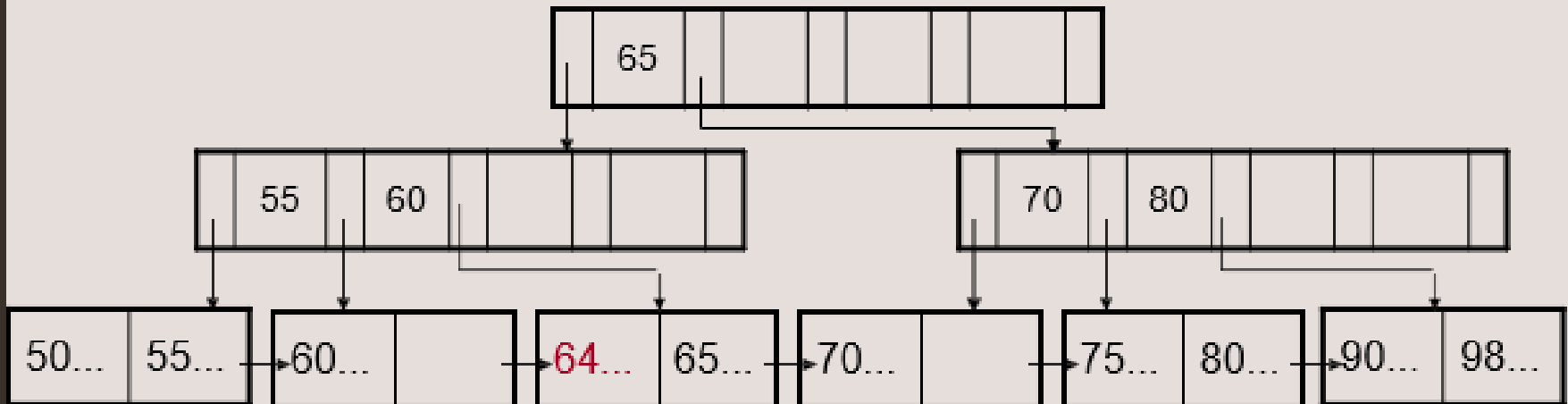
EXCLUSÃO



A exclusão sem concatenações requer $h+1$ acessos à memória secundária e a exclusão em que há concatenação em todas as páginas do caminho da folha (ponto de remoção ou de redistribuição) até a raiz requer $2h + 1$ acessos à memória secundária, logo o custo da operação de remoção é $O(h)$, sendo h a altura da árvore.

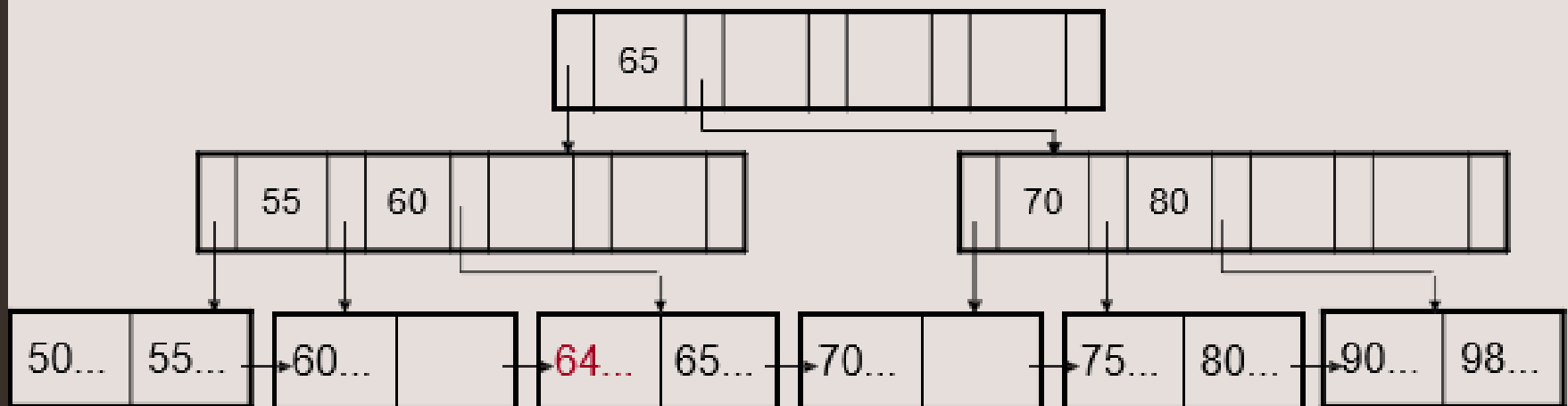
Árvore B+

Consiste numa variante da árvore B que, como sugere o nome, apresenta vantagens em relação à estrutura original.

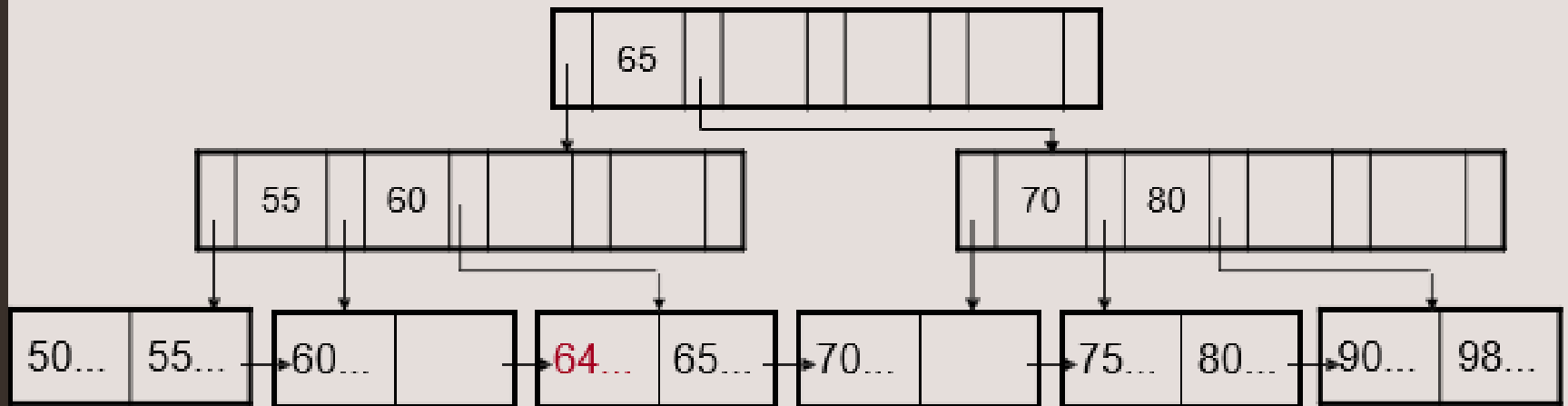


Árvore B+

Geralmente as árvores B+ apresentam menor profundidade que as árvores B com mesmos dados, pois nas B+ há aumento do número de entradas mantidas por nó.

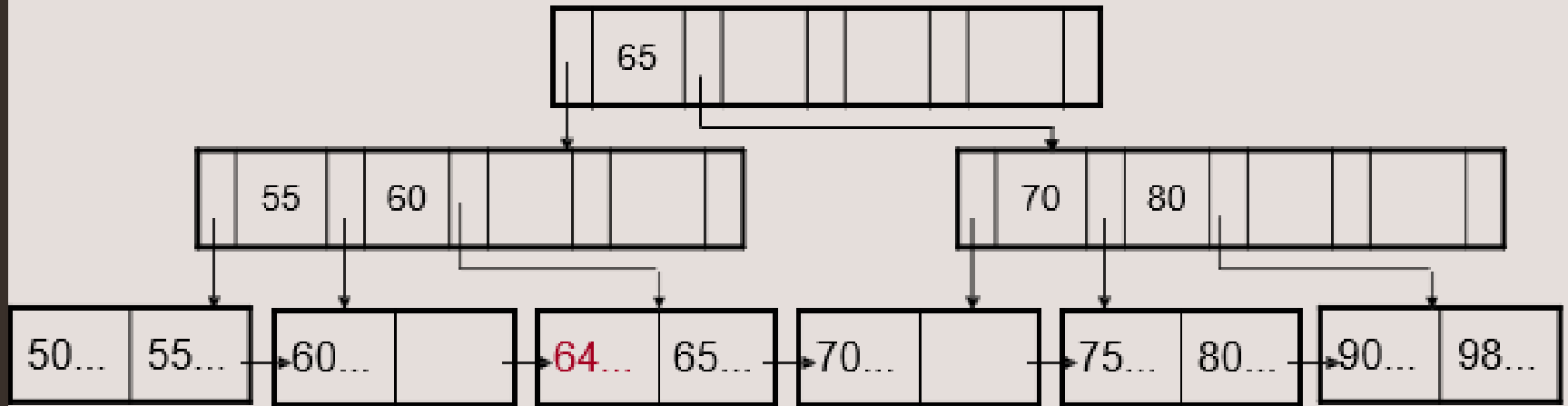


Árvore B+



O tamanho dos nós é determinado pela capacidade da memória principal. Nas B+ só as chaves são mantidas nos nós não terminais, assim tem-se árvores com maior número de entradas por nó (página) e menos profundidade (com impacto positivo sobre os custos), pois as chaves constituem apenas um subconjunto dos registros completos.

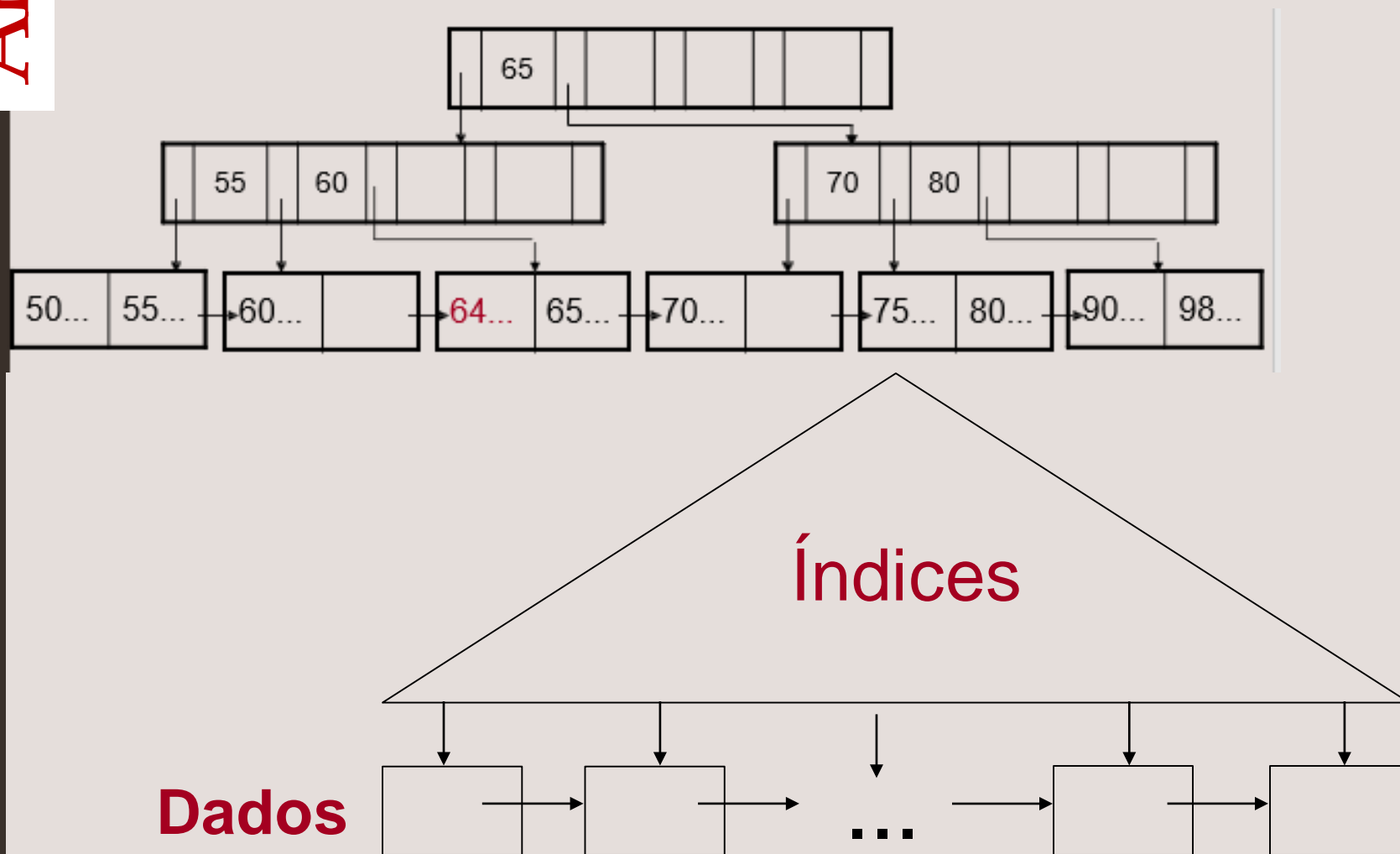
Árvore B+



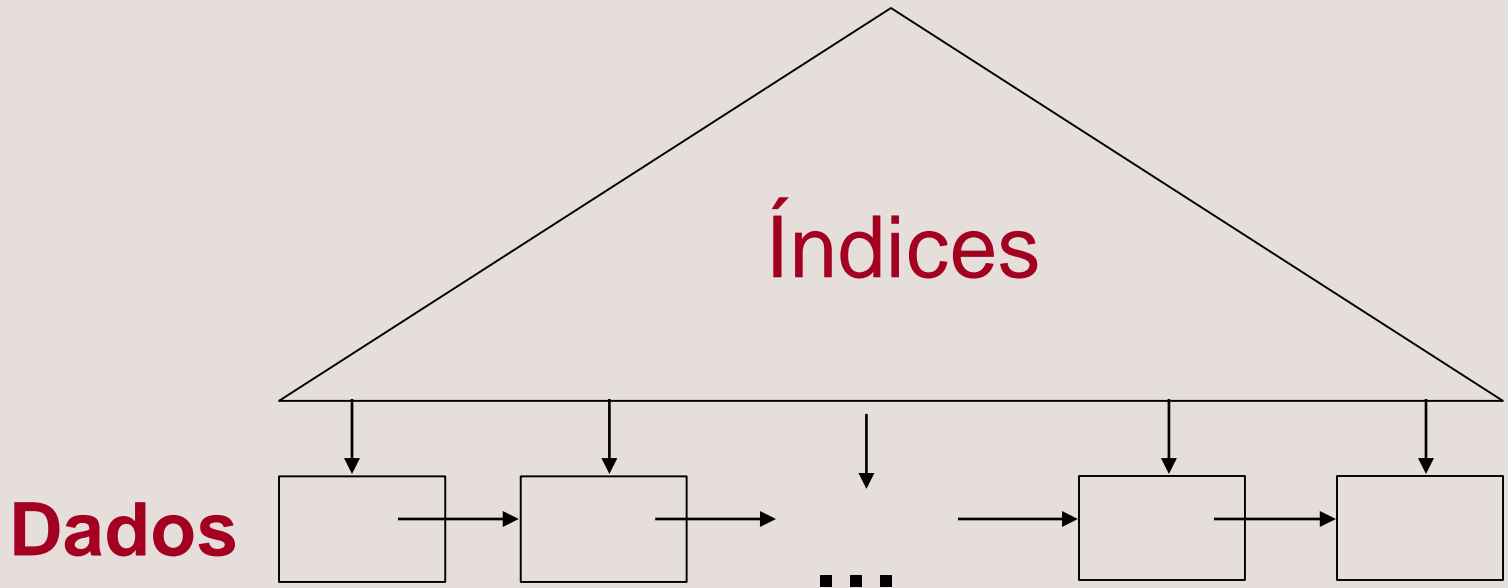
Nas árvores B+ os registros completos são armazenados exclusivamente nas folhas. Os nós não terminais atuam como índices de acesso aos dados contidos nos nós terminais.

Árvore B+

Uma árvore B+ é uma árvore B em que os registros de dados são mantidos nos nós folhas e os nós não terminais constituem índices de acesso aos dados.

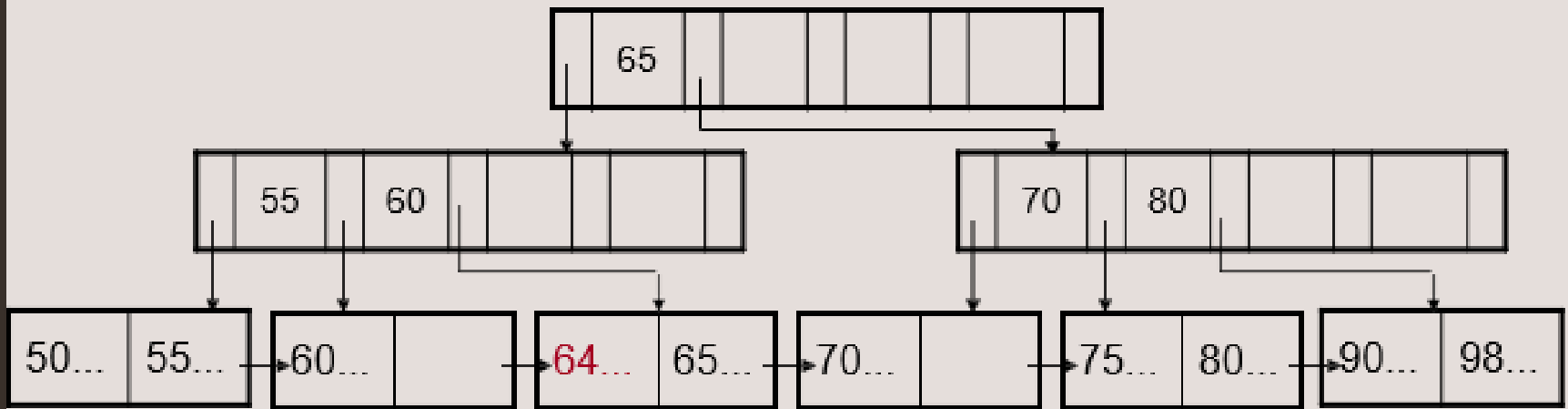


Árvore B+



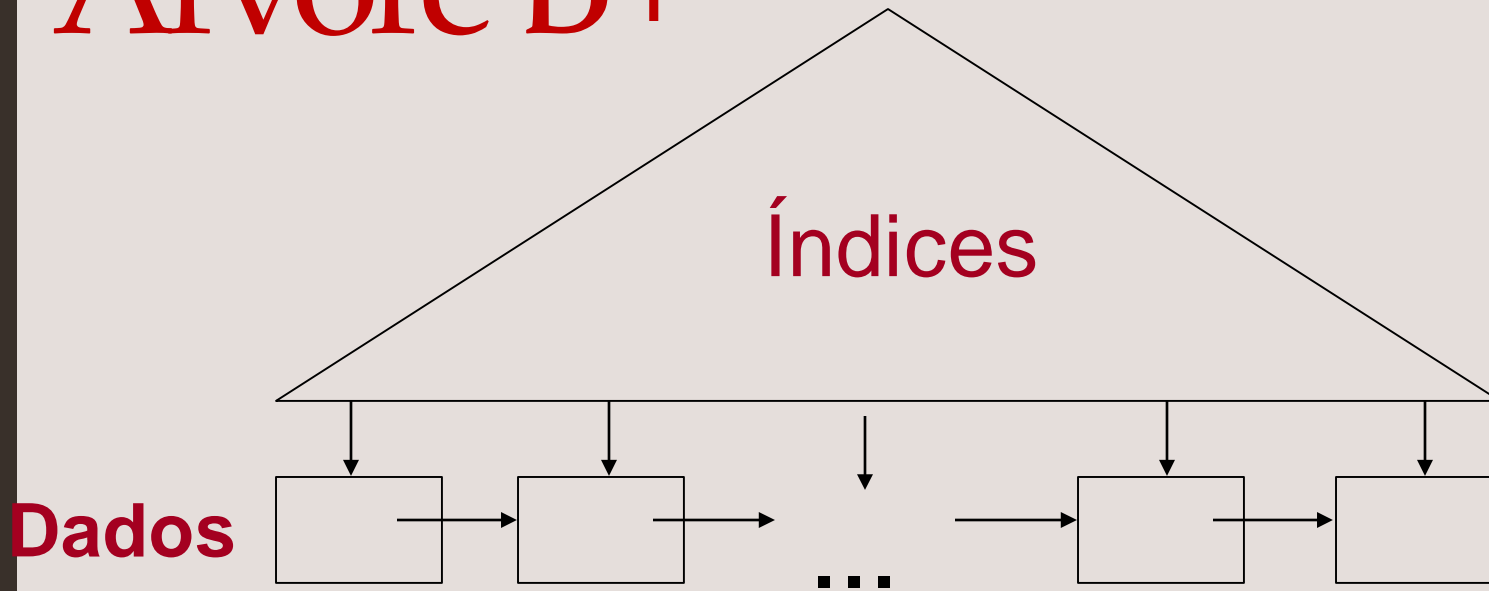
É possível um processamento linear dos dados, garantido pelos links que mantêm os nós folhas dispostos em sequência, bem como o acesso direto às páginas dos nós folhas por meio dos índices.

Árvore B+



- É possível efetuar processamento direto ou sequencial sobre os dados mantidos em árvores B+. *Ver.*

Árvore B+



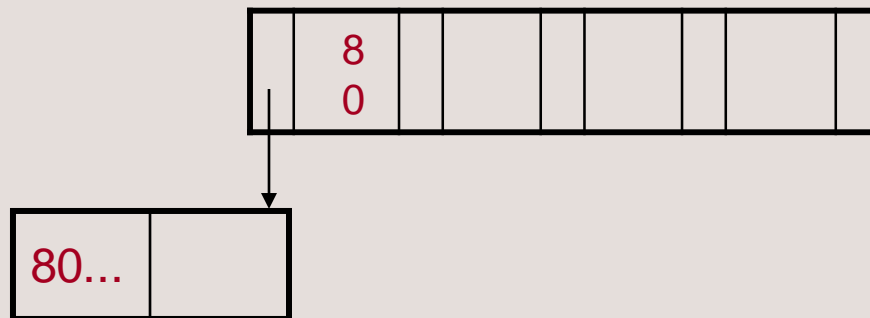
- Na inserção de dados em árvores B+ os registros são armazenados nos nós folhas e as chaves são incluídas nos nós índices com um ponteiro, ou caminho, para o respectivo nó terminal.

Árvore B+

INCLUSÃO

Considera-se uma árvore B+ de ordem 2.

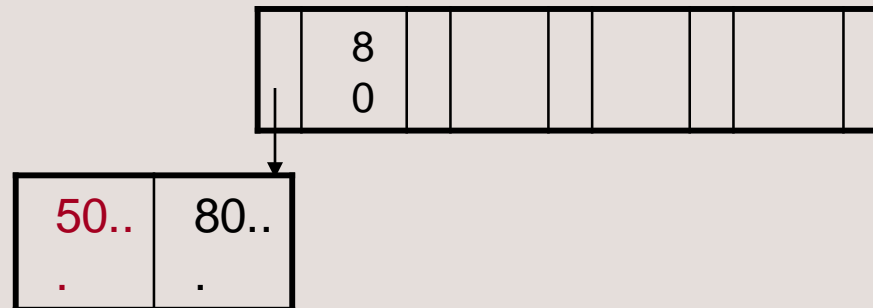
1 Incluindo $80^1 - 50^2 - 98^3 - 90^4 - 60^5 - 65^6 - 70^7 - 55^8 - 64^9$.



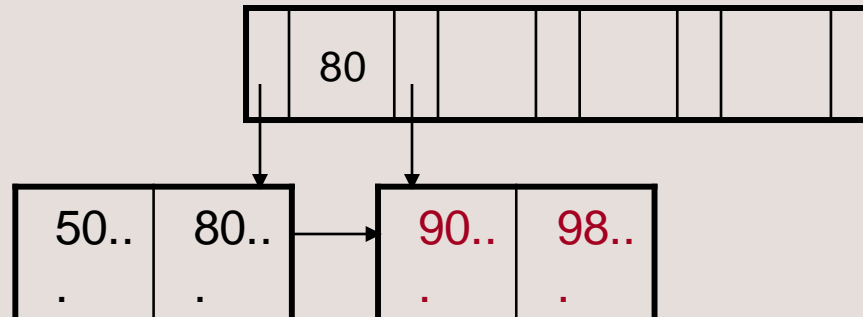
Árvore B+

INCLUSÃO

2 Incluindo 50:

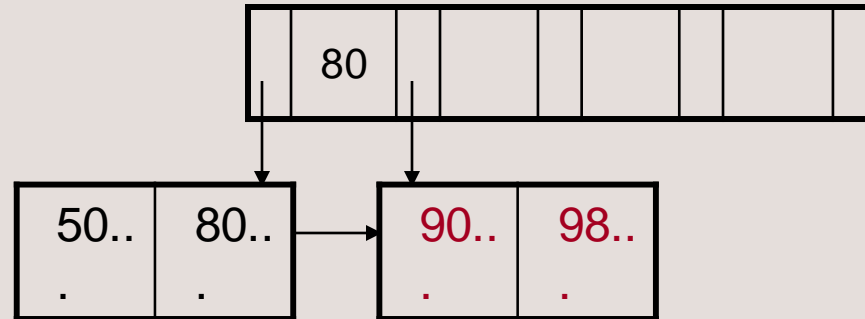


3 e 4 Incluindo 98 e 90:



Árvore B+

INCLUSÃO

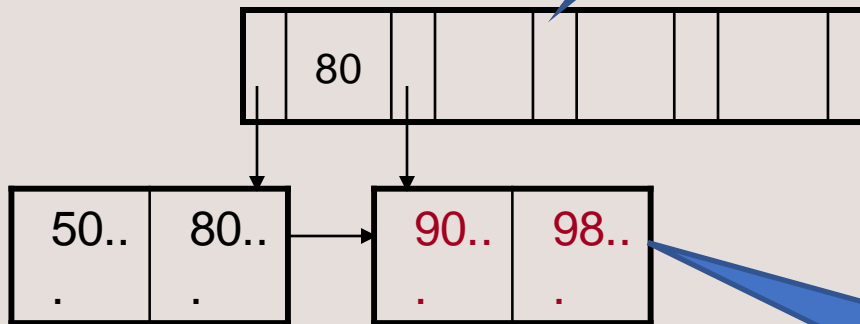


Por que a
distinção entre
os nós?

Árvore B+

INCLUSÃO

Nós não terminais, de índices, compostos por chaves e organizados hierarquicamente.

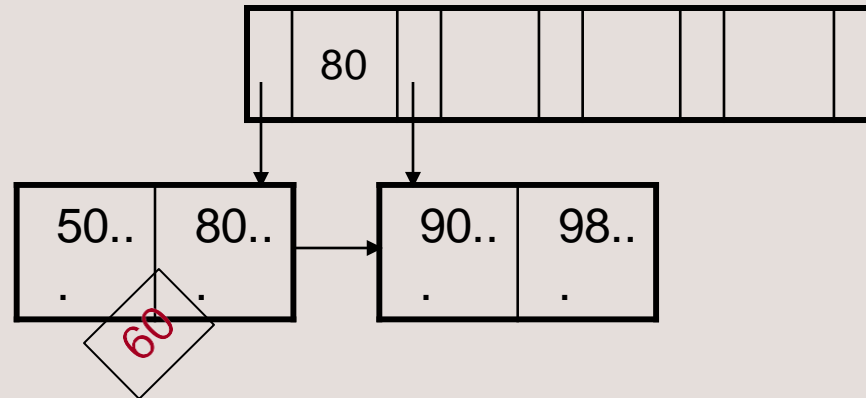


Nós folhas, de dados, compostos por registros “completos” e organizados de forma linear.

Árvore B+

INCLUSÃO

5 Incluindo 60:

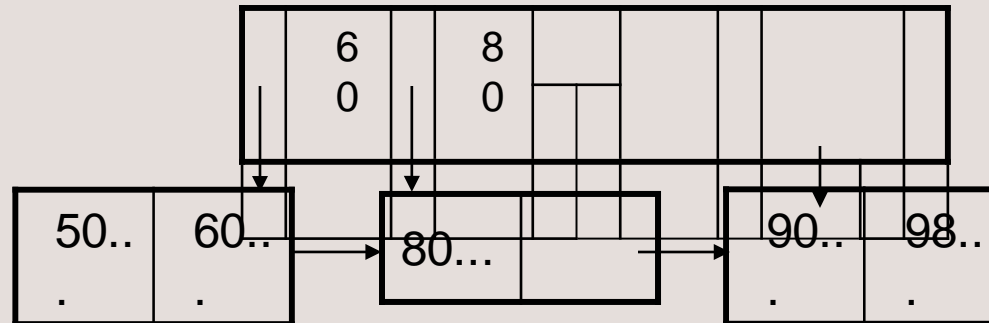
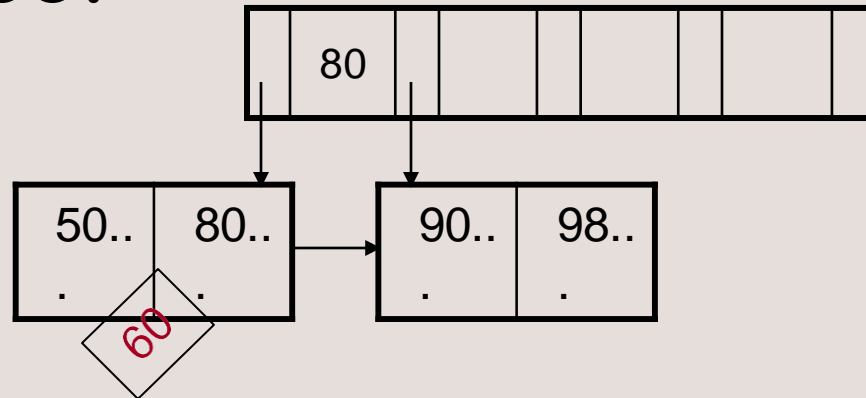


Esta implica em estouro do nó 50|80 e provoca o encaminhamento da chave mediana para o nó ascendente e do 80 para próximo espaço.

Árvore B+

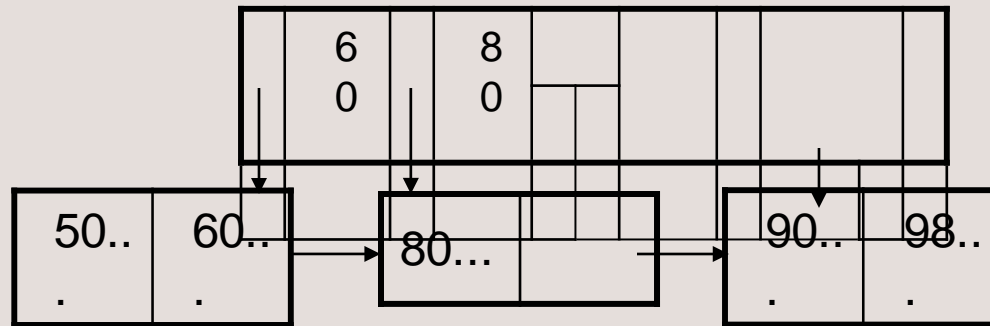
INCLUSÃO

5 Incluindo 60:



Árvore B+

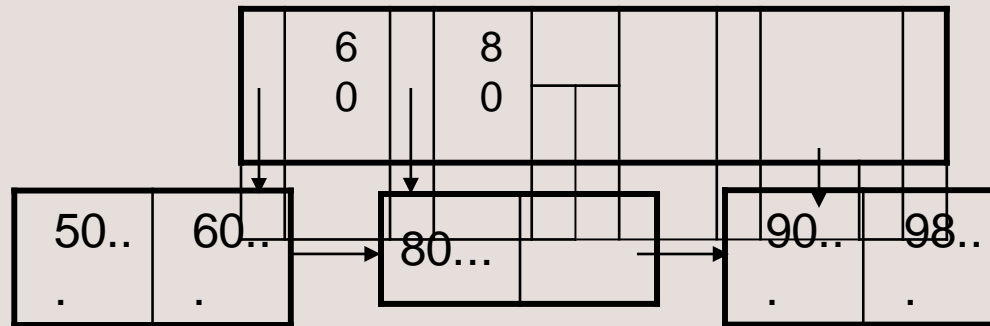
INCLUSÃO



Por que os dados
60 e 80 estão
registrados em
dois lugares da
árvore?

Árvore B+

INCLUSÃO

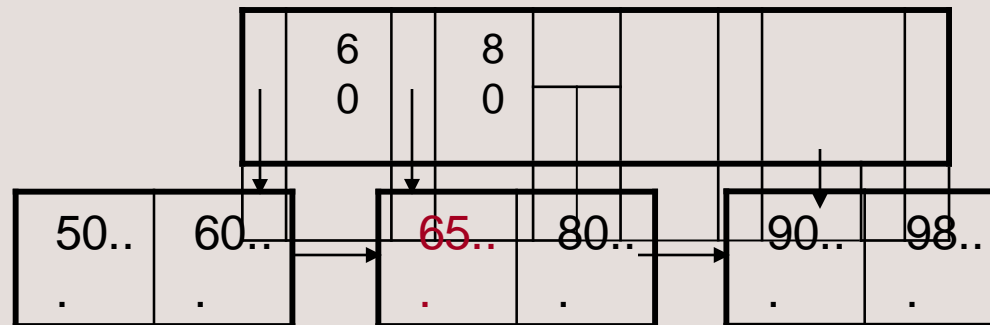
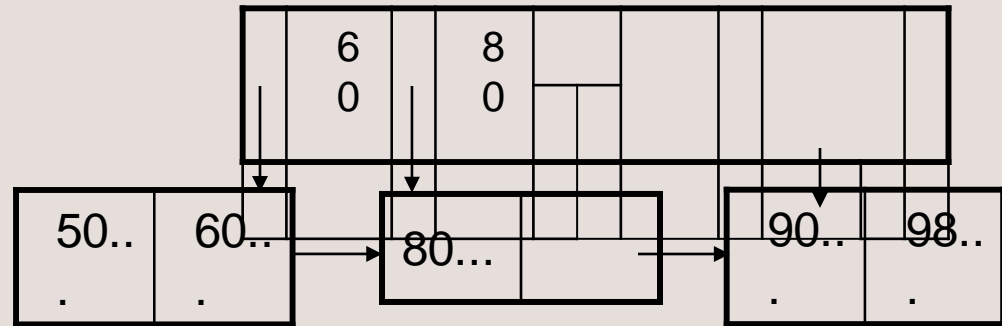


Como efetuar a
inclusão do nó
65?

Árvore B+

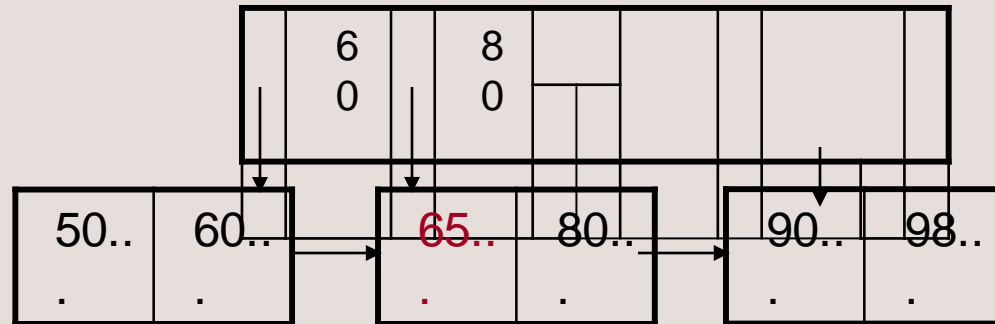
INCLUSÃO

6 Incluindo 65:



Árvore B+

INCLUSÃO

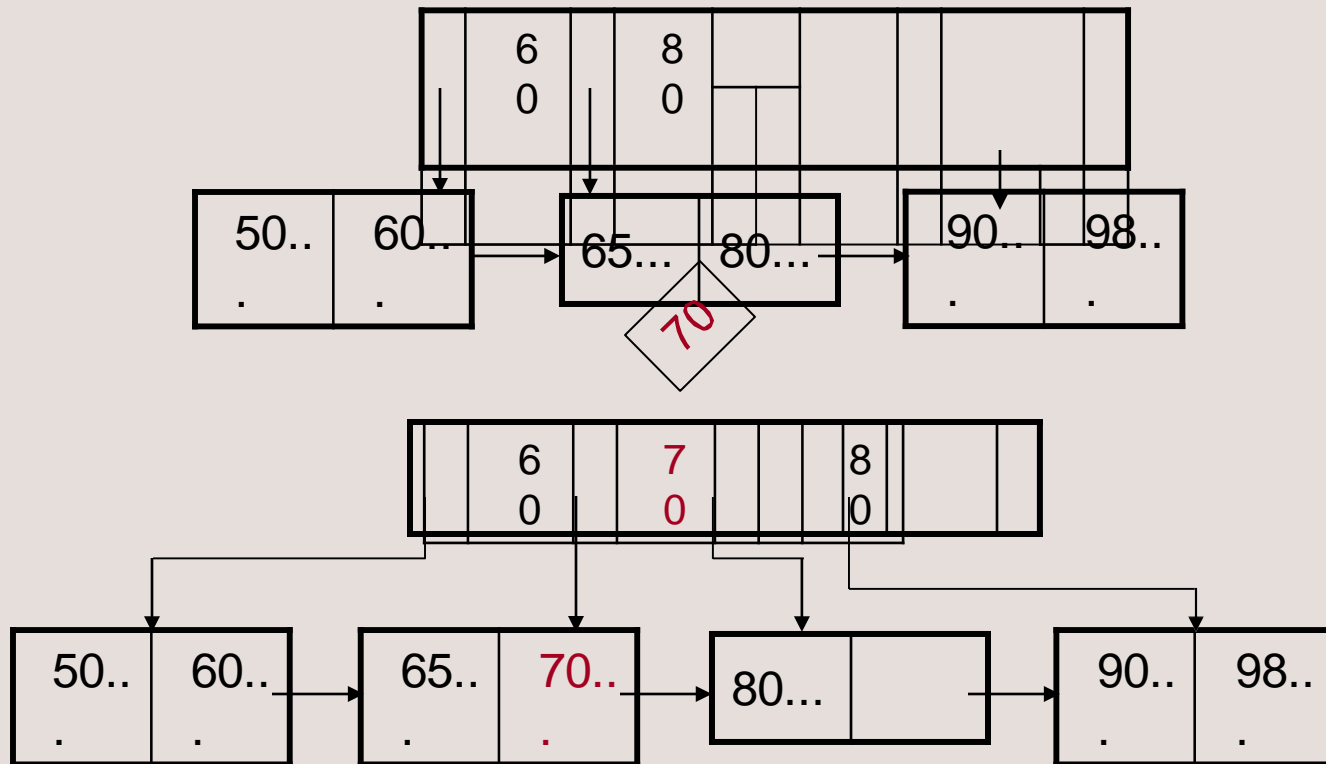


Como efetuar a
inclusão do nó
70?

Árvore B+

INCLUSÃO

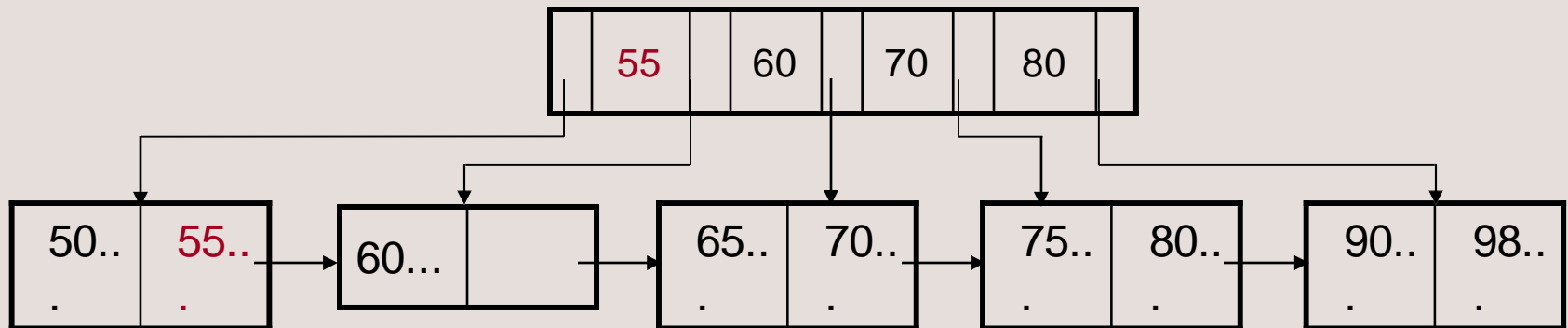
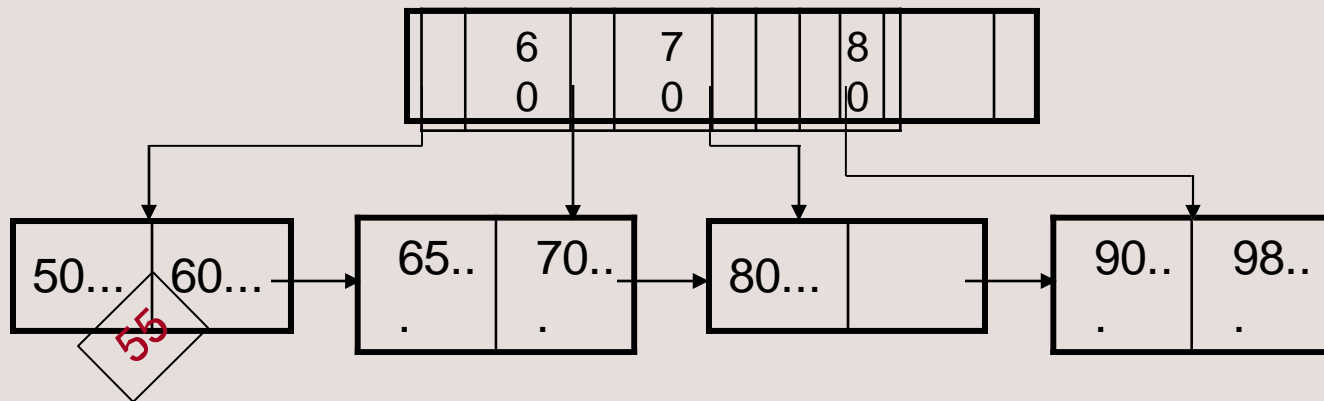
7 Incluindo 70:



Árvore B+

INCLUSÃO

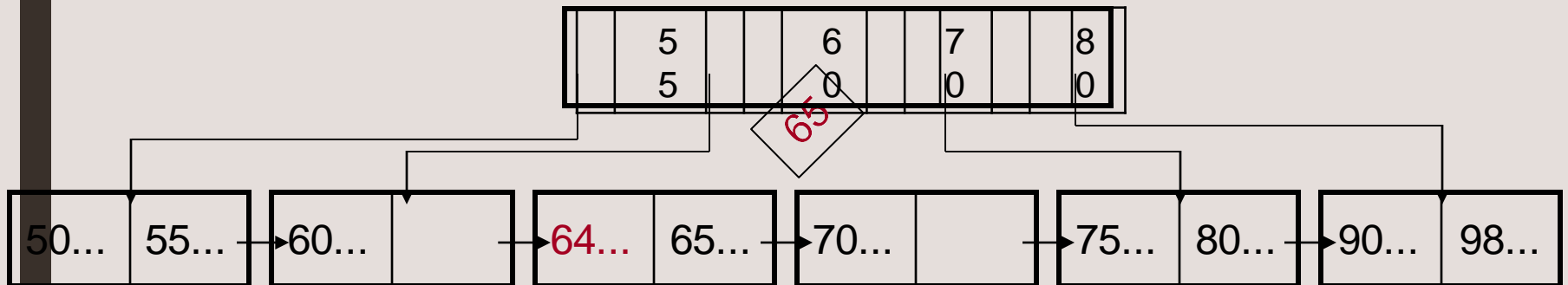
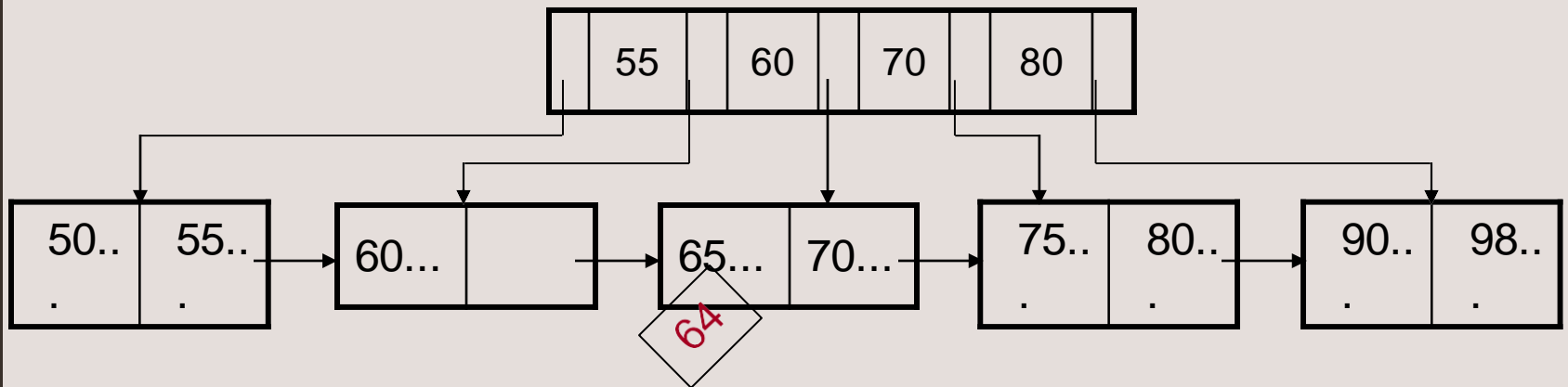
8 Incluindo 55:



Árvore B+

INCLUSÃO

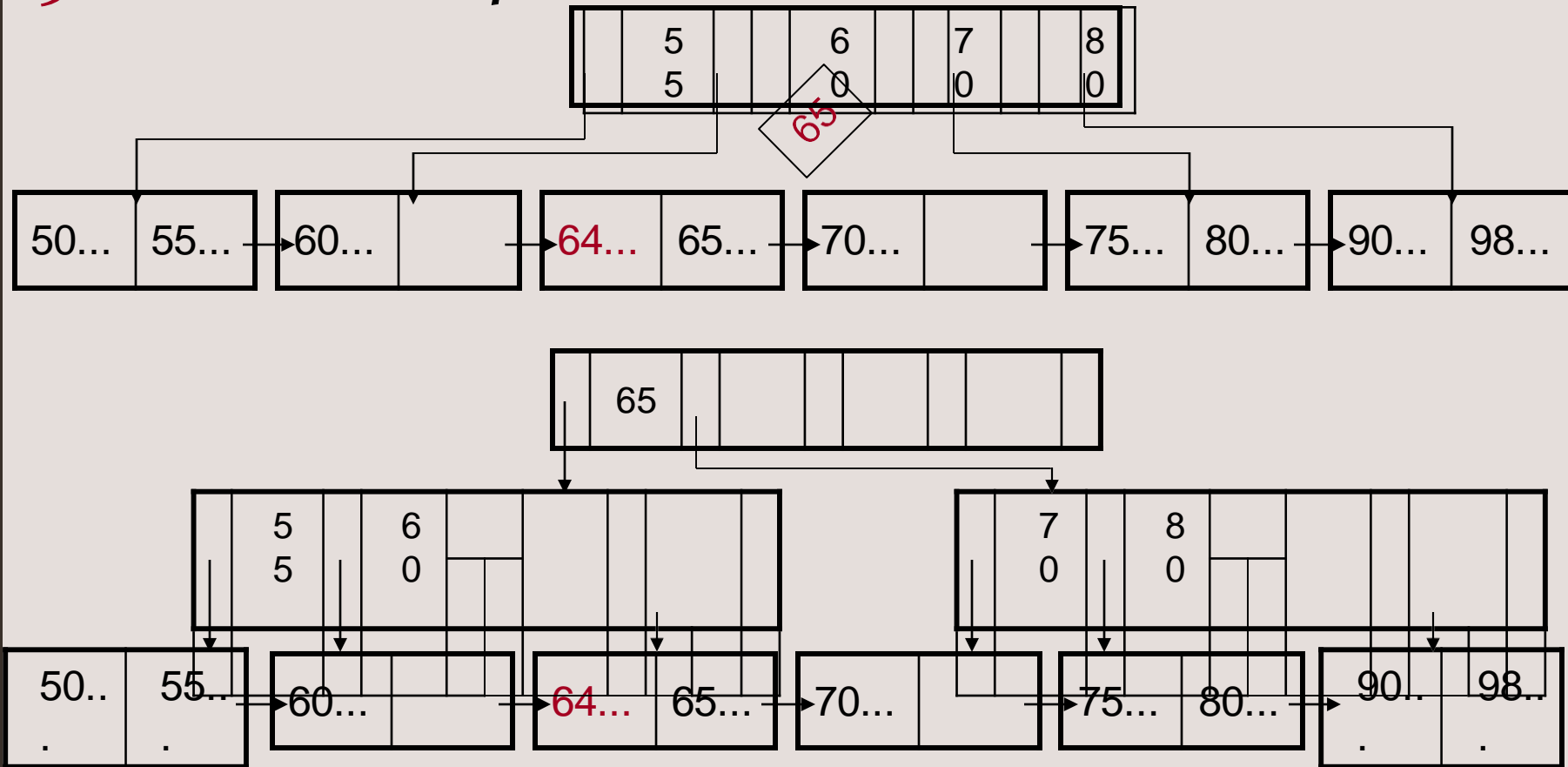
9 Incluindo 64:



Árvore B+

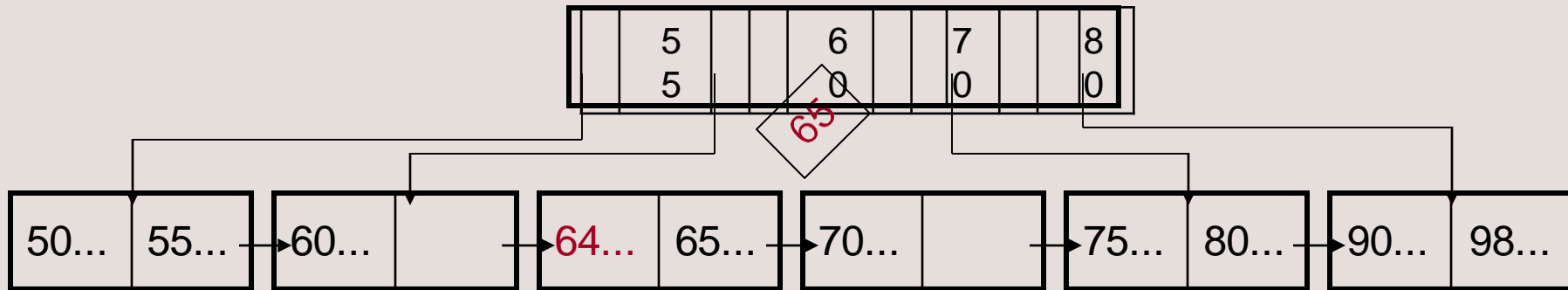
INCLUSÃO

9 Incluindo 64:



Árvore B+

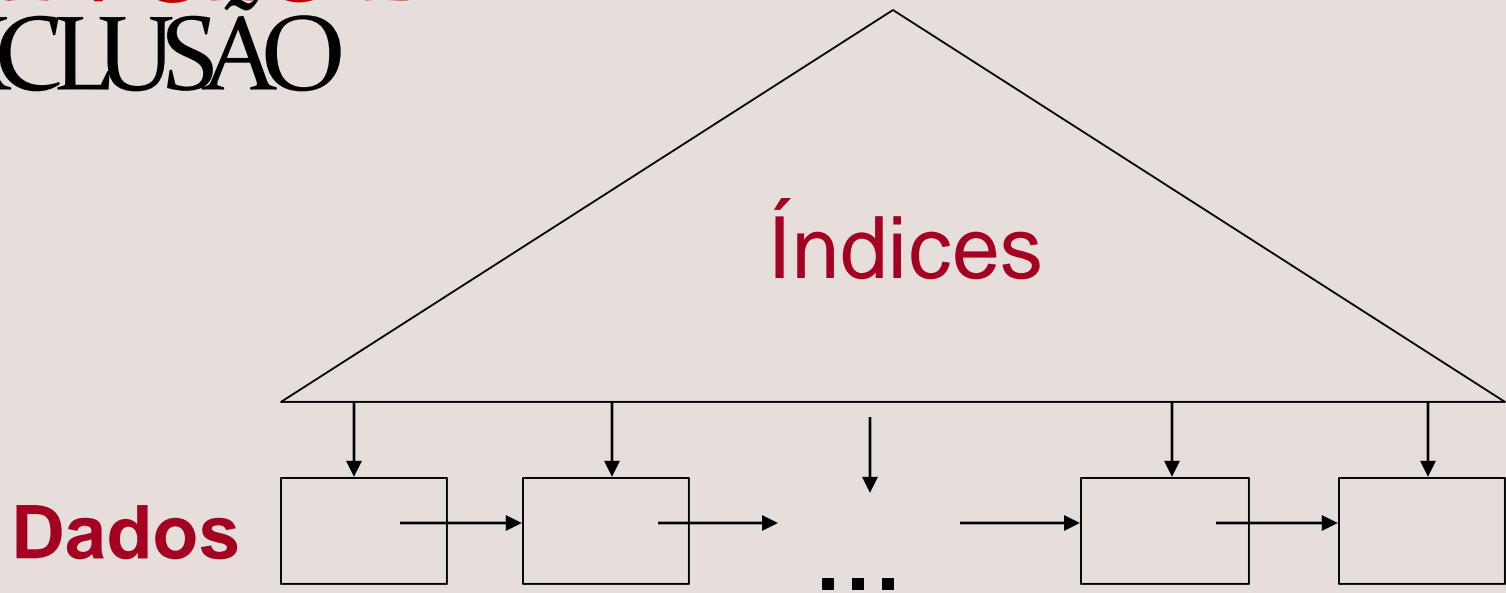
INCLUSÃO



- A inserção de 64 provoca o estouro do nó 65||70 e o encaminhamento de 65 ao nó ascendente que, por sua vez, implica no estouro do nó 55||60||70||80 e consequente encaminhamento de 65 ao nível ascendente, com crescimento *bottom up* da estrutura.

Árvore B+

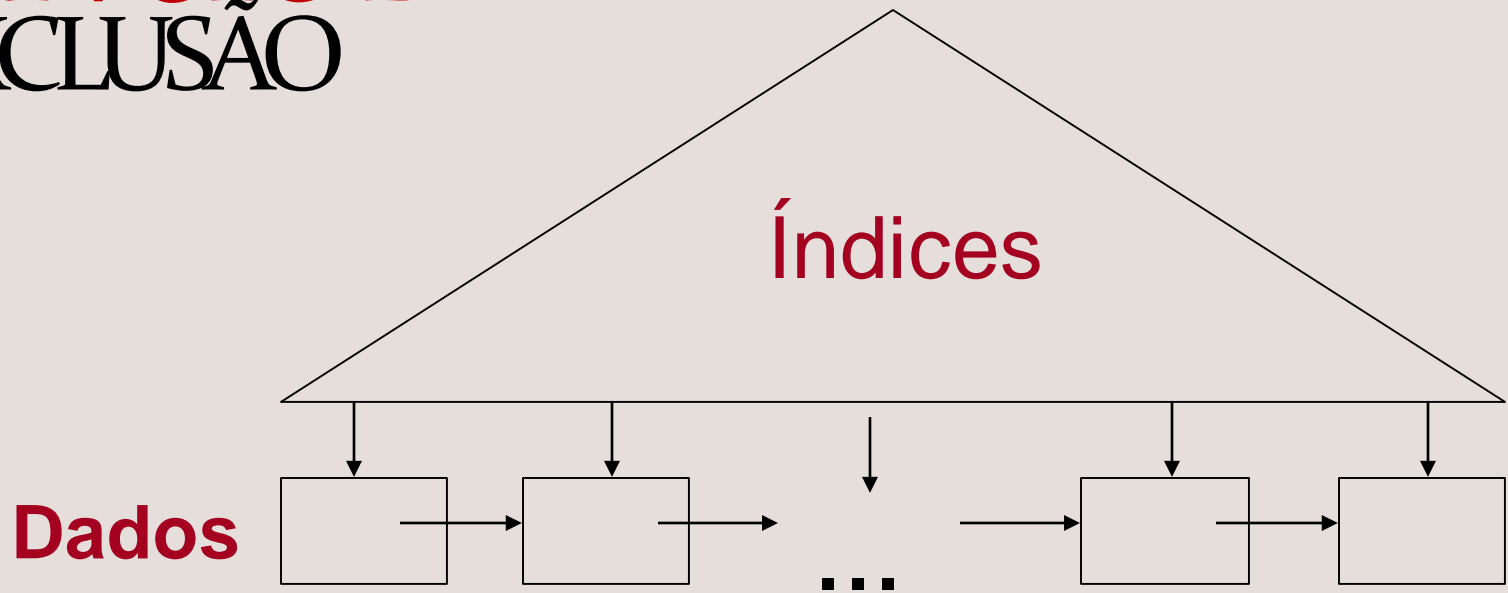
EXCLUSÃO



Em função da forma de organização das árvores B+, nem toda exclusão provoca alterações nas duas áreas de composição da estrutura de dados e de índices.

Árvore B+

EXCLUSÃO



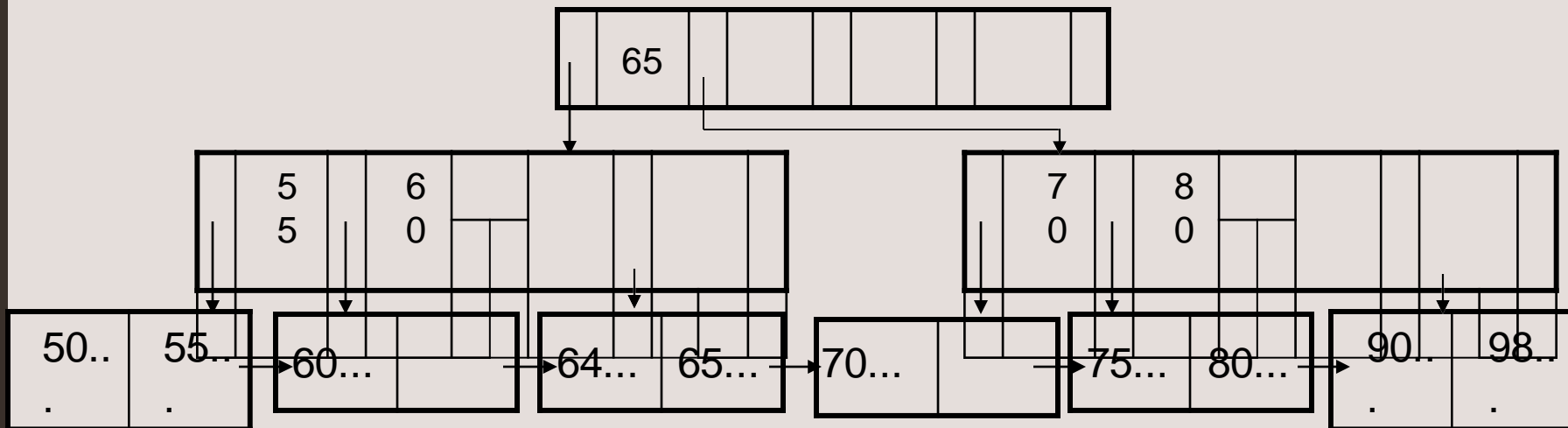
A remoção de nós da área de dados é equivalente à remoção em estruturas lineares.

Já a exclusão na área de índices é similar a que ocorre em árvores B.

Árvore B+

EXCLUSÃO

Discutir remoção das chaves 55 e 60 da estrutura dada abaixo:



Atividade Periódica II6

QUESTIONÁRIO

_Atividade proposta
no SIGAA_

Complementar Estudos...

File Organization and Processing

Allan L Tharp

Capítulo

9 B-Trees and Derivatives

Complementar Estudos...

Estruturas de Dados e Seus Algoritmos

Szwarcfiter e Markenzon

Capítulo

5 Árvores Balanceadas

Complementar Estudos...

Árvores B

<https://www.ime.usp.br/~pf/estruturas-de-dados/aulas/B-trees.html>

Complementar Estudos...

B-Trees

☐ Max. Degree = 3
☒ Max. Degree = 4
☐ Max. Degree = 5
☐ Max. Degree = 6
☐ Max. Degree = 7

```
graph TD; Root[0045] --- Leaf1[0019]; Root --- Leaf3[0234];
```

Visualiza Árvores B

<https://www.cs.usfca.edu/~galles/visualization/BTree.html>

Próximos Passos...



Estruturas de
Busca em Texto