

Índices

Banco de Dados II

(Capítulos 8, 10 e 11 – Ramakrishnan

Capítulo 12 – Silberschatz

Capítulo 14 – Garcia-Molina, Ullman, Widom)

Berkeley → <http://www-inst.eecs.berkeley.edu/~cs186/>

Índices

Why Indexing is important?



Un-Indexed Database

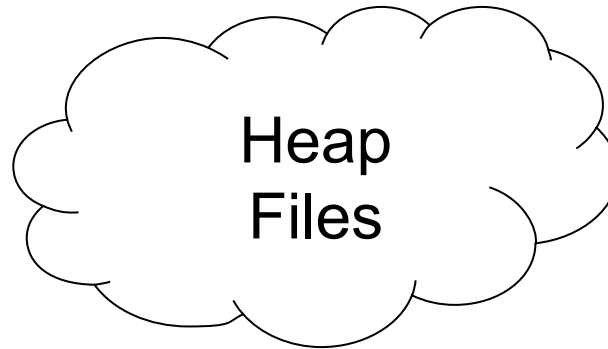


INDEXED Database

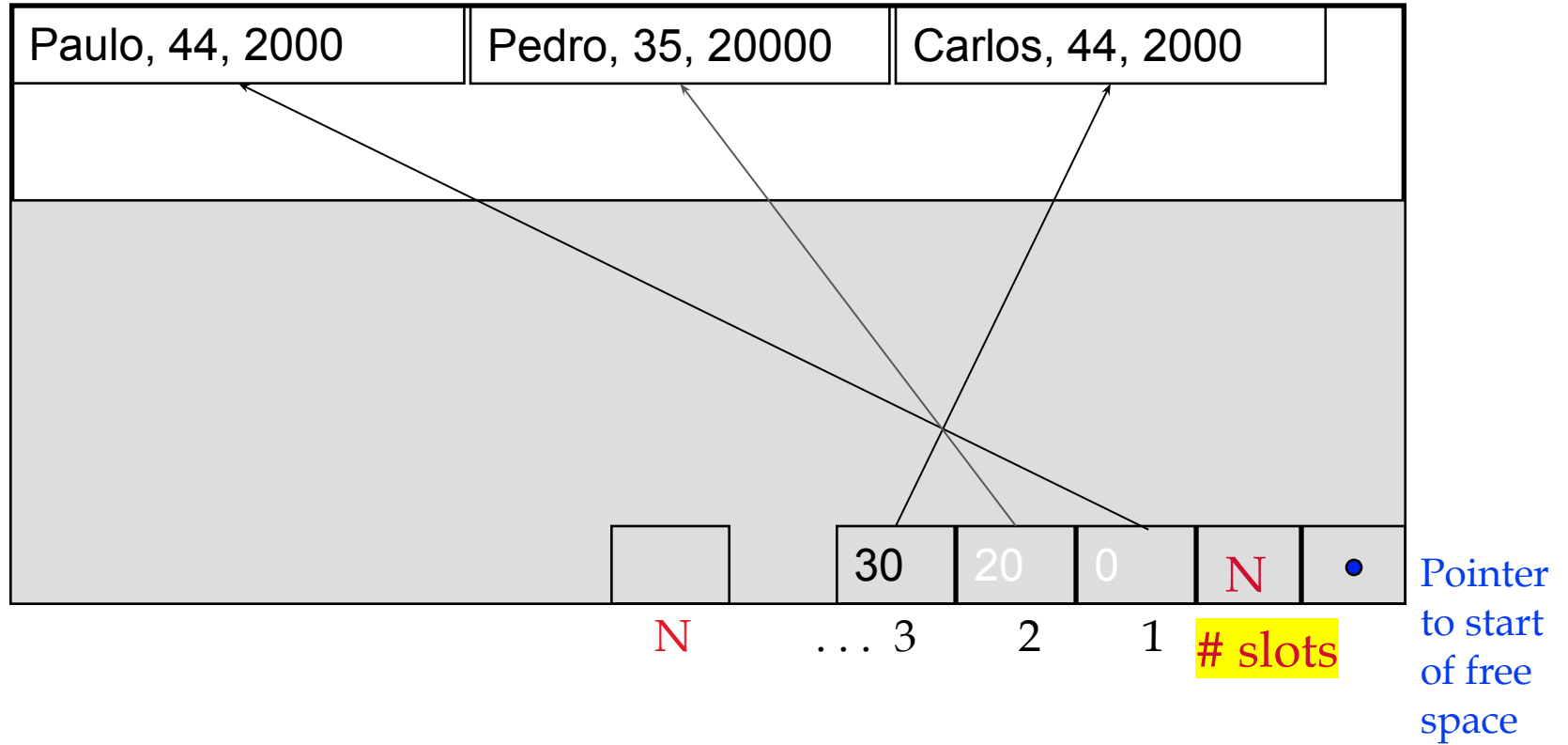
Introdução

Dados sem organização

Paulo, 44, 2000
Pedro, 35, 20000
Carlos, 44, 2000
José, 40, 2500
João, 35, 3000
Ilmério, 40, 3500
Rodrigo, 40, 3500
Maria, 30, 4000
Sara, 35, 4000
Sabrina, 31, 5000



Heap Files



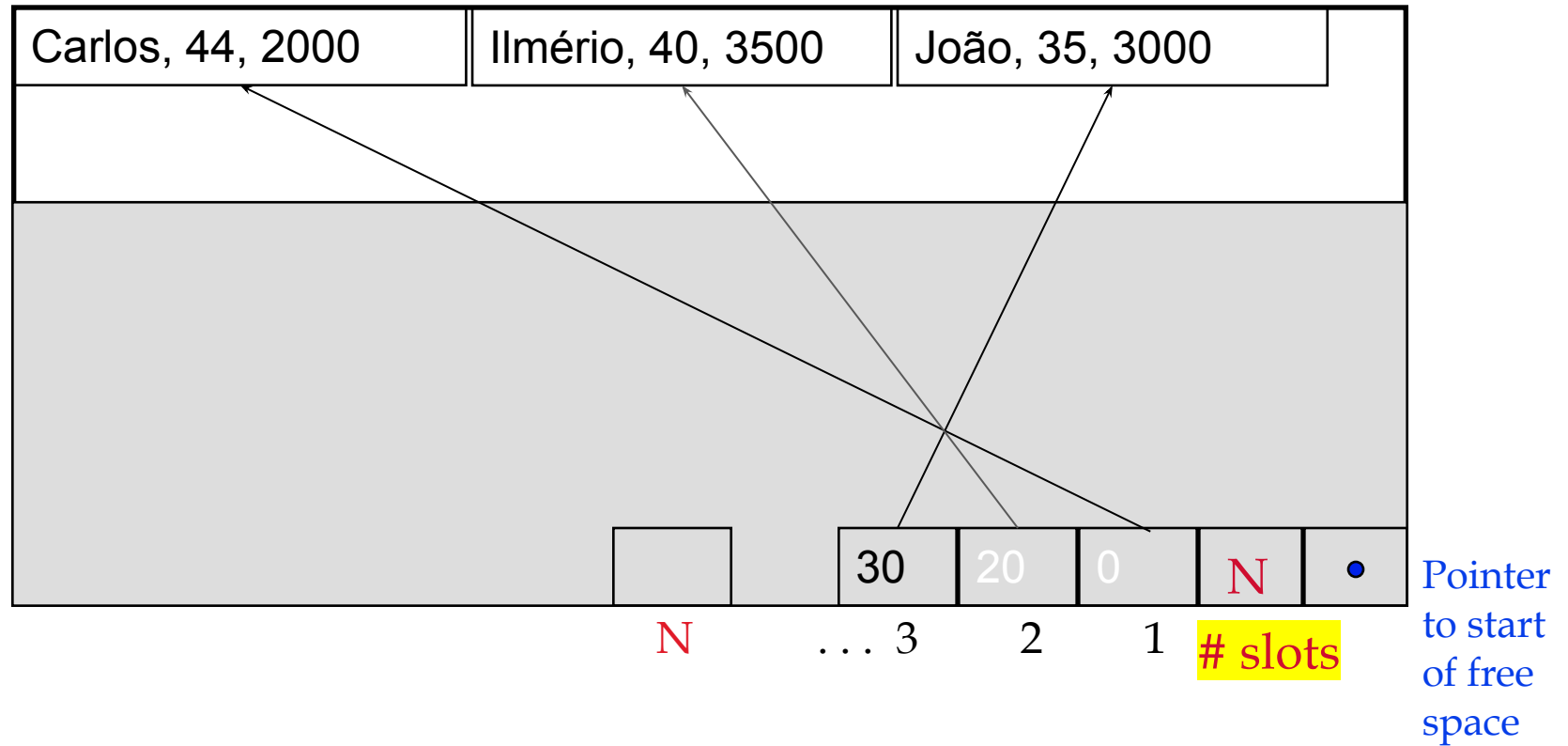
Introdução

Dados Organizados (nome)

Carlos, 44, 2000
Ilmério, 40, 3500
João, 35, 3000
José, 40, 2500
Maria, 30, 4000
Paulo, 44, 2000
Pedro, 35, 2000
Rodrigo, 40, 3500
Sabrina, 31, 5000
Sara, 35, 4000

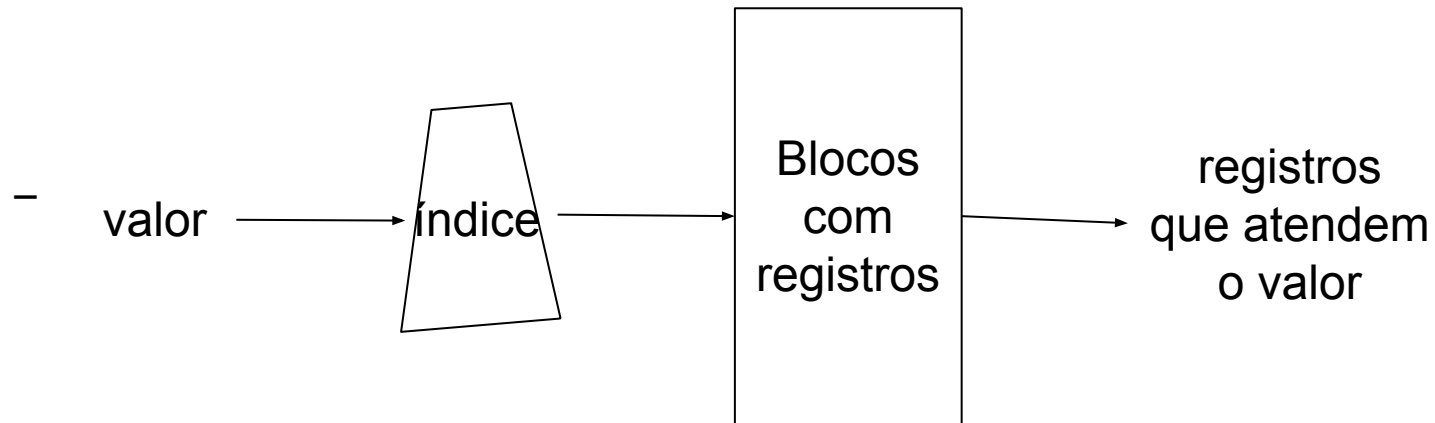


Heap Files - Exemplo



Índices

Estrutura auxiliar projetada para agilizar operações de busca, inserção e supressão



Alteração nos dados pode levar à alteração no índice

Espaço extra de armazenamento

Índices

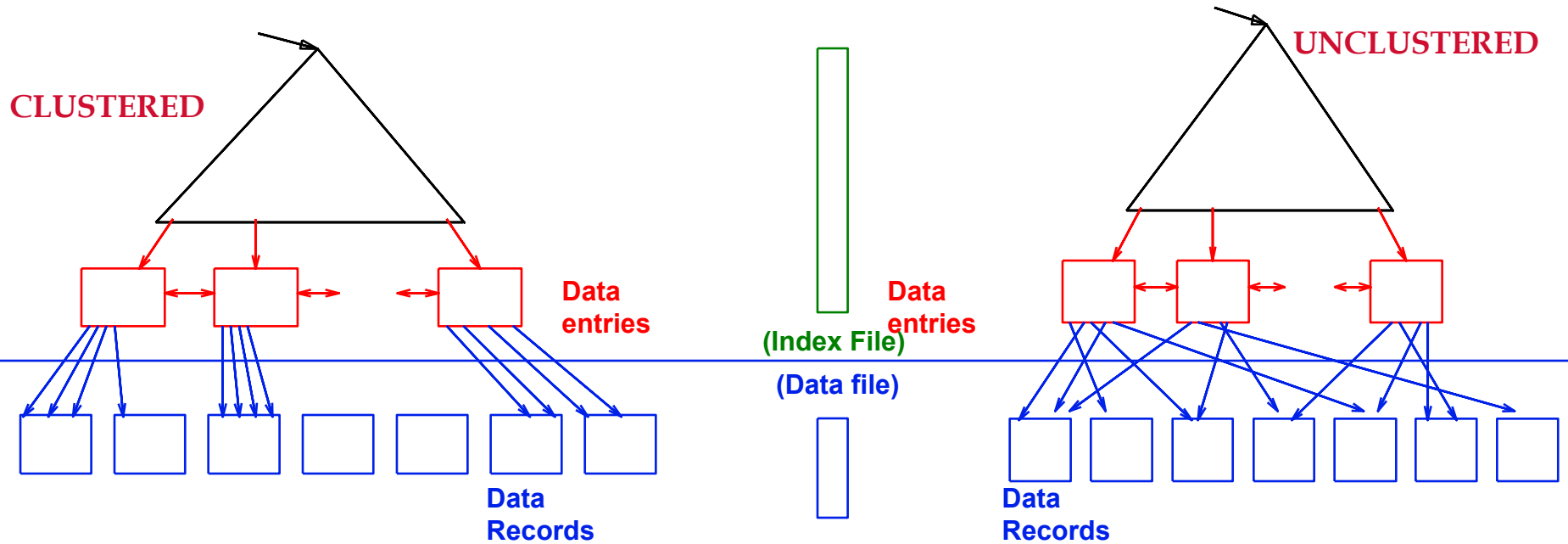
- O que armazenar em cada entrada do arquivo de índice
 - Alternativa 1** -> Entrada = Registro inteiro
 - Alternativa 2** -> Entrada = chave, rid
 - Alternativa 3** -> Entrada = chave, conjunto de rids
- Essa escolha é ortogonal ao índice;

Classificação dos índices

- *Clustered*: se a ordem dos registros são os mesmos da ordem do índice de dados de entrada.
- *Unclustered*: os registros são armazenadas sem um ordenamento

Um arquivo pode ser clusterizado por somente uma chave

Clustered vs. Unclustered Index



Atividade

1 - Criação de dados

```
create table table_teste as select generate_series(1,10000000) AS id1,  
floor(random() * 10000000 + 1)::int as id2;
```

```
SET enable_bitmapscan TO off;
```

2- visualização dos dados

```
select *,ctid from table_teste;
```

3 - Criação de um índice sobre desc

```
create index lteste on table_teste(id2);
```

4- Reestruturação dos registros

```
cluster table_teste using lteste;
```

5 - select *,ctid from table_teste;

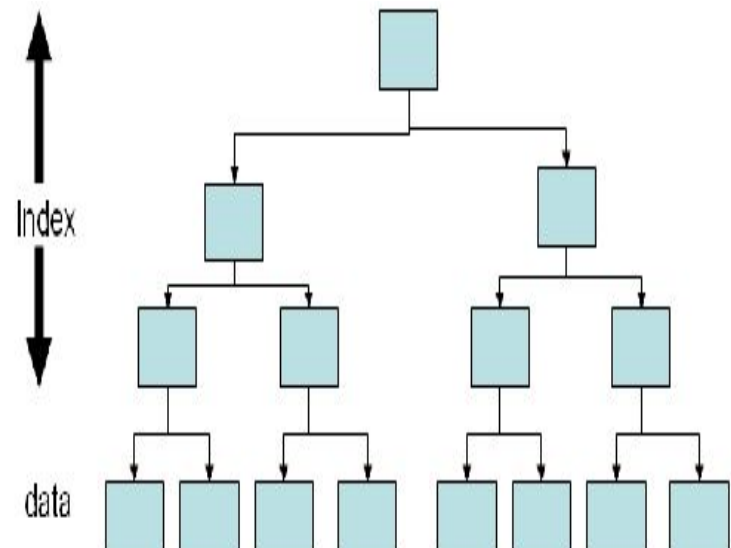
Descreva qual ação que o código promoveu na tabela table_teste. O que acontece com o ordenamento em caso de um novo insert na tabela?

Clustered vs. Unclustered

- Clustered Pros
 - Eficiente em buscas entre faixas ($>$ $<$)
 - Pode ser executado algum tipo de compressão
 - Possível benefício na localidade dos dados
- Clustered Contras
 - Caro para manter
 - Espaço extra de armazenamento

Índices

- Primário
 - A chave do índice é composta pela chave primária da tabela
 - A maioria dos SGB cria índices primários automaticamente
 - Não permitem duplicatas
- Secundário
 - Outras colunas da tabela participam
 - Permitem duplicatas



Índices

- O que precisa saber:
 - Característica dos dados
 - Como dada é usado:
 - Tipos de consultas
 - Frequência de consultas

Índices

- Colunas “boas” para indexação:
 - PK e FK
 - consultas por range
 - consultas em ordem
- Colunas “ruins” para indexação:
 - consultas esparsas
 - colunas com poucos valores únicos
 - colunas com texto

Índices

<i>last_name</i>	<i>first_name</i>
Randall	Joshua
.	.
.	.
.	.
Randall	Cynthia
Randall	Tristan
.	.
.	.
.	.
Ota	Lani
.	.
.	.
.	.

Alta Densidade

```
SELECT *  
FROM member  
WHERE last_name =  
'Randall'
```

Baixa Densidade

```
SELECT *  
FROM member  
WHERE last_name = 'Ota'
```


Índices

<i>member_no</i>	<i>last_name</i>	<i>first_name</i>
1	Randall	Joshua
2	Flood	Kathie
.		
.		
.		
10000	Anderson	Bill

```
SELECT *  
FROM member  
WHERE member_no > 8999
```

Alta Seletividade

$$\frac{\text{Número de linhas atende ao critério}}{\text{Total de linhas}} = \frac{1000}{10000} = 10\%$$

<i>member_no</i>	<i>last_name</i>	<i>first_name</i>
1	Randall	Joshua
2	Flood	Kathie
.		
.		
.		
10000	Anderson	Bill

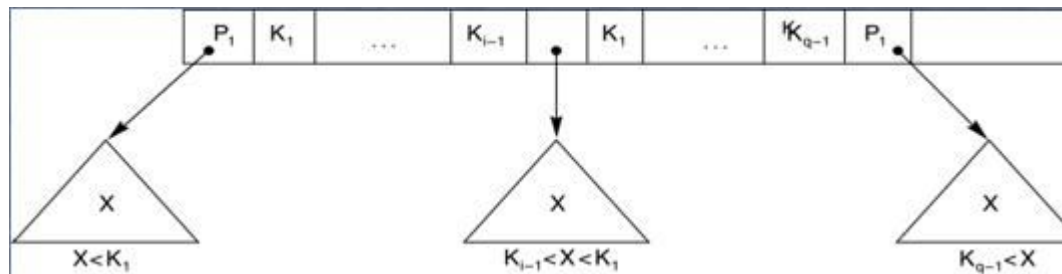
```
SELECT *  
FROM member  
WHERE member_no < 9001
```

Baixa Seletividade

$$\frac{\text{Número de linhas atende ao critério}}{\text{Total de linhas}} = \frac{9000}{10000} = 90\%$$

Índices

- Os índices de múltiplos níveis podem formar uma árvore



- Atualização nos dados, implica na atualização em todos os níveis

Índices B+

- Os SGBs implementam índices multiníveis através de árvores B+
- Atualização dos níveis mais eficientes
- Cada nível elimina vários acessos
- O grau (ou ordem) da árvore indica o número de acessos

Índices B+

- Insert/delete com custo de I/O: $\log F (N)$

Onde N = número de folhas e F = ordem ou grau da árvore

- Mínimo de 50% de ocupação (exceto nó root).

- Suporte de seleção por igualdade e por range.

- Busca sempre no nó folha

- Exemplo:

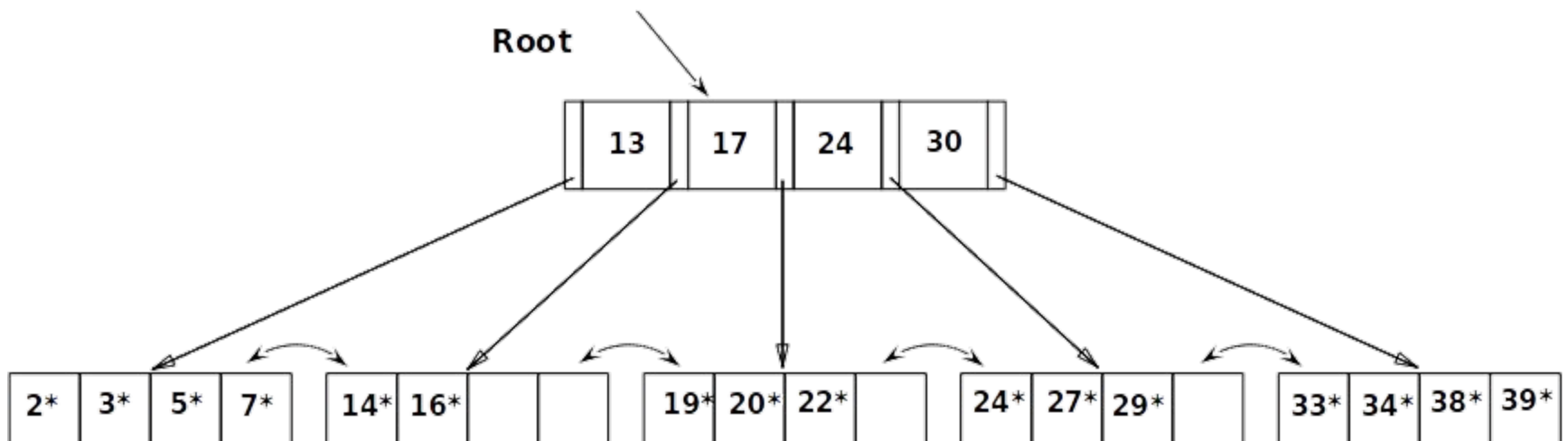
Ordem 5:

Número mínimo de chaves = $(M-1)/2$

Número máximo de chaves = 4

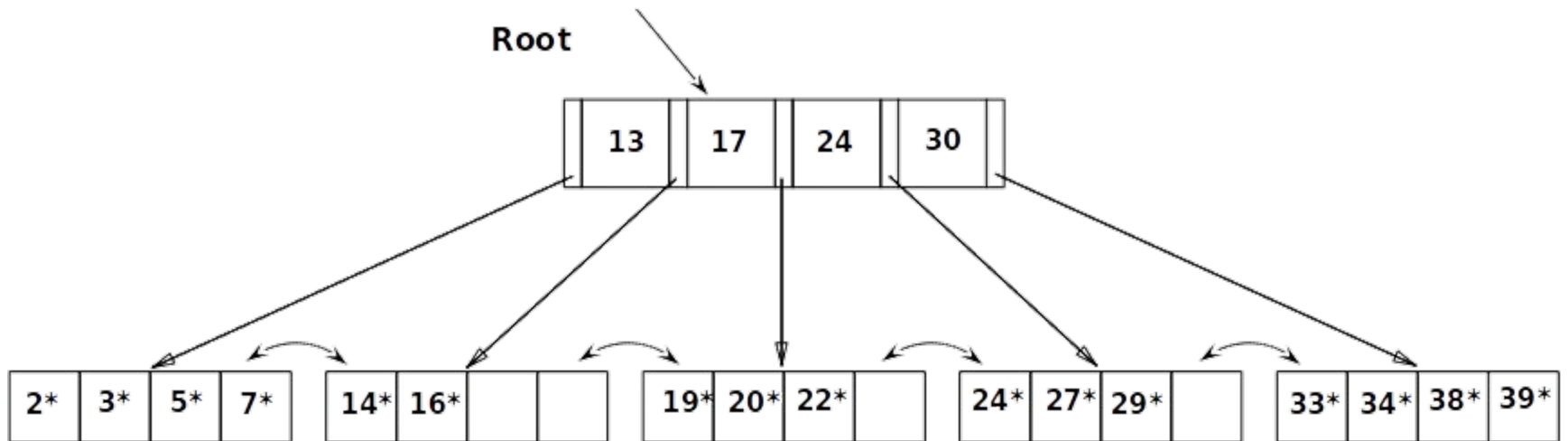
Índices B+

Ordem 5



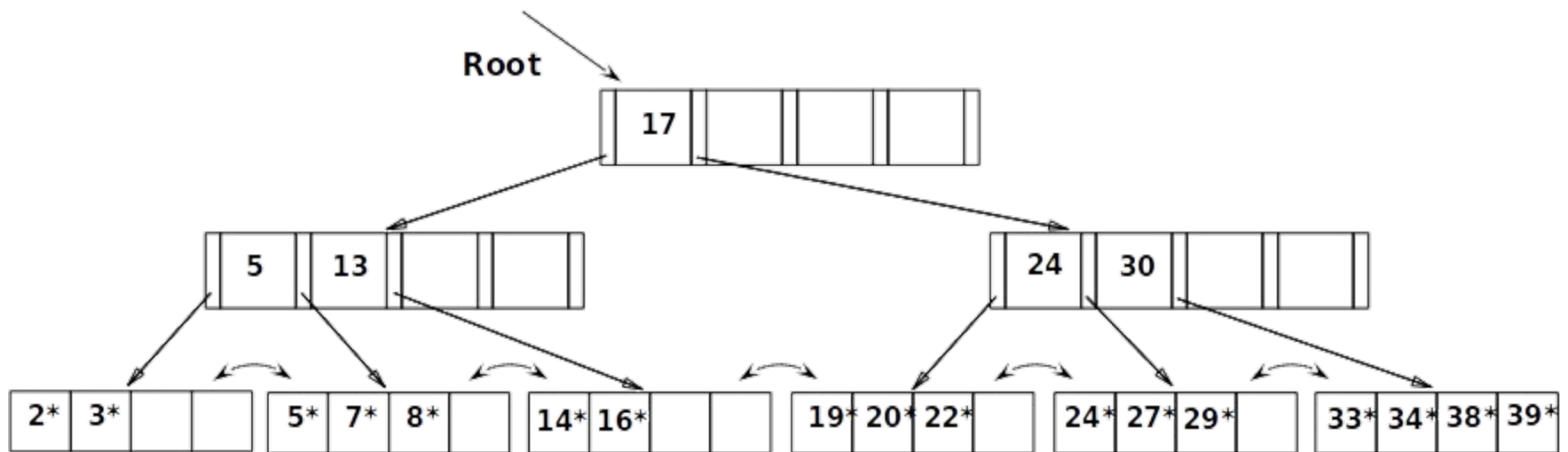
Índices B+

Inserte valor 8*



Índices B+

Deletar a entrada 19*



Índices B+

Use o simulador (<http://www.cs.usfca.edu/~galles/visualization/BPlusTree.html>)

1- Inserir em um B+ de ordem 5 os seguintes valores: 10, 15, 20, 25, 30, 35, 40, 45, 50

2 -Remover os seguintes valores: 35 30 25 20

Atividade

- 1) Inserir em um B+ com ordem 4 as seguintes entradas :2, 6, 17, 20, 24, 25, 27, 29, 30, 31, 32, 5, 21, 1, 40, 45, 50, 70
- 2) Remover os itens 25, 6, 5 e 20.
- 3) Qual a altura mínima e máxima de uma árvore de ordem 100 com 1 milhões de chaves.

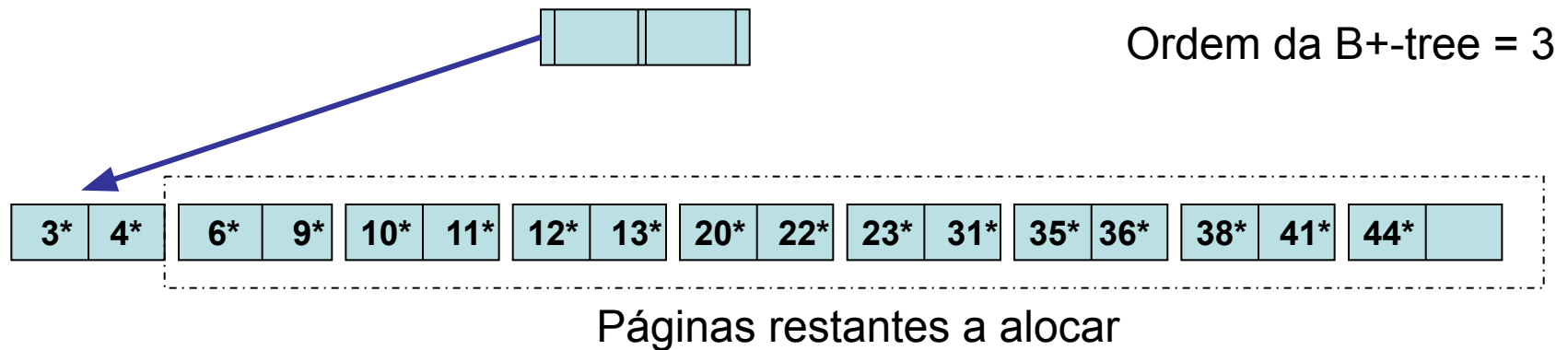
Índices

- Let's check it out:
 - Animação ([here](http://www.cs.usfca.edu/~galles/visualization/BPlusTree.html))
 - Or type `http://www.cs.usfca.edu/~galles/visualization/BPlusTree.html`

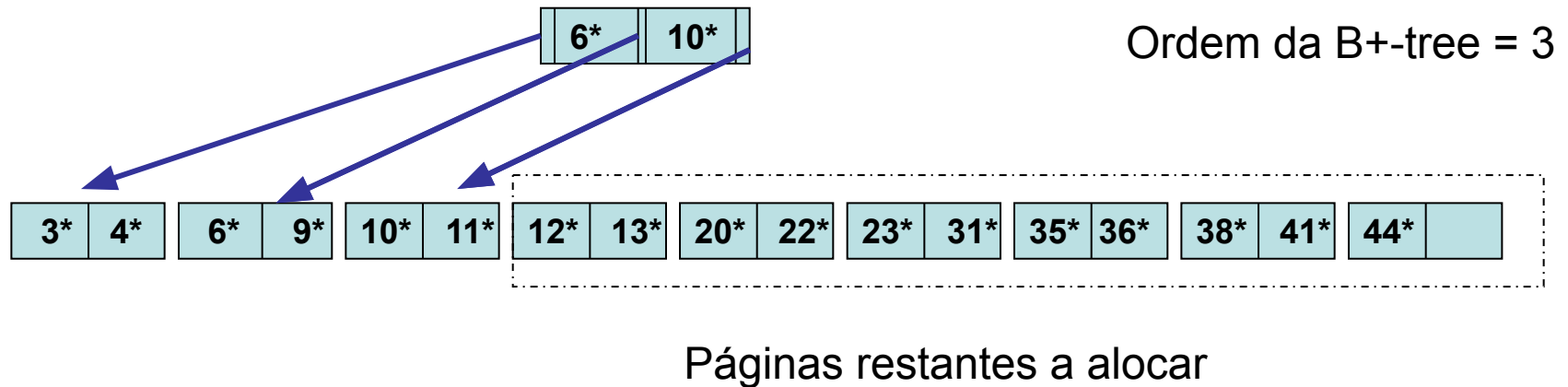
Bulking Load (Construção da B+ Tree)

- Utilizando o arquivo de índice denso (ou arquivo ordenado)
- Aloca-se uma página vazia para a raiz
- Insere nesta página um ponteiro para a primeira página do arquivo contendo as entradas.

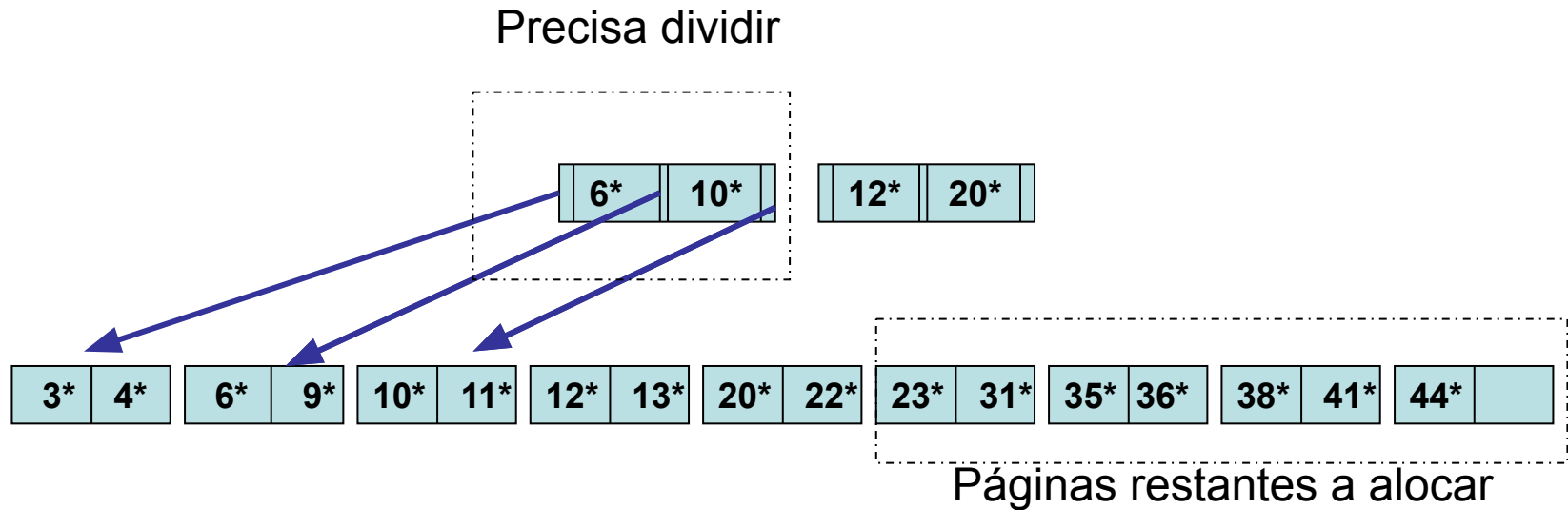
Bulking Load (Construção da B+ Tree)



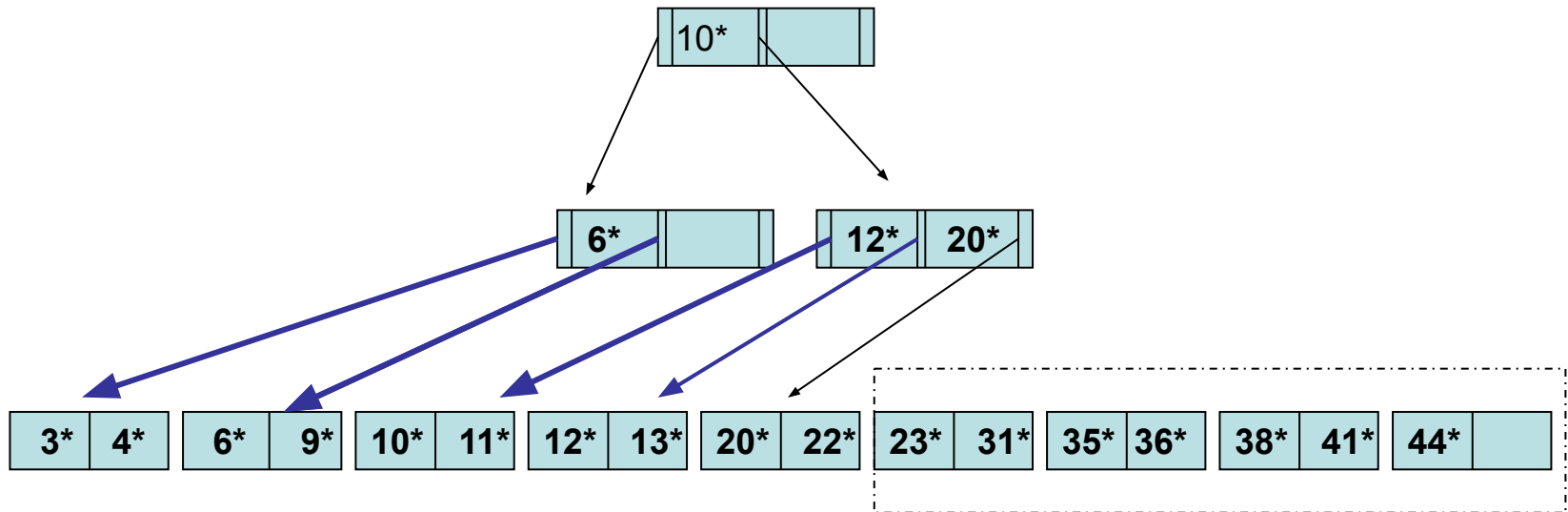
Bulking Load (Construção da B+ Tree)



Bulking Load (Construção da B+ Tree)

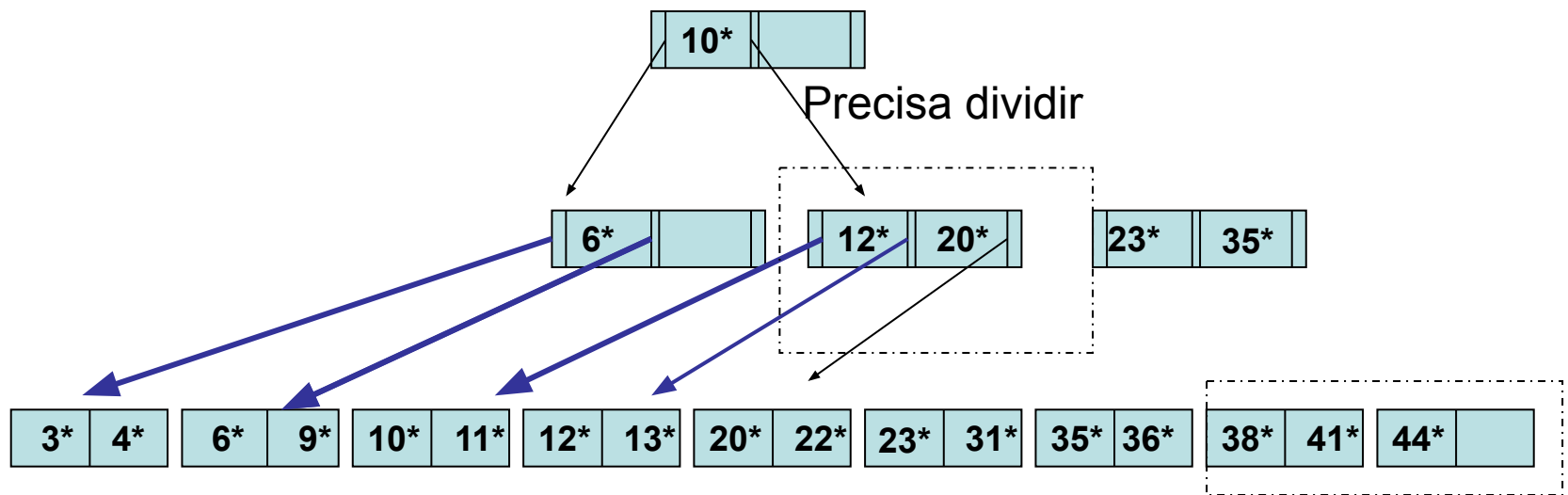


Bulking Load (Construção da B+ Tree)

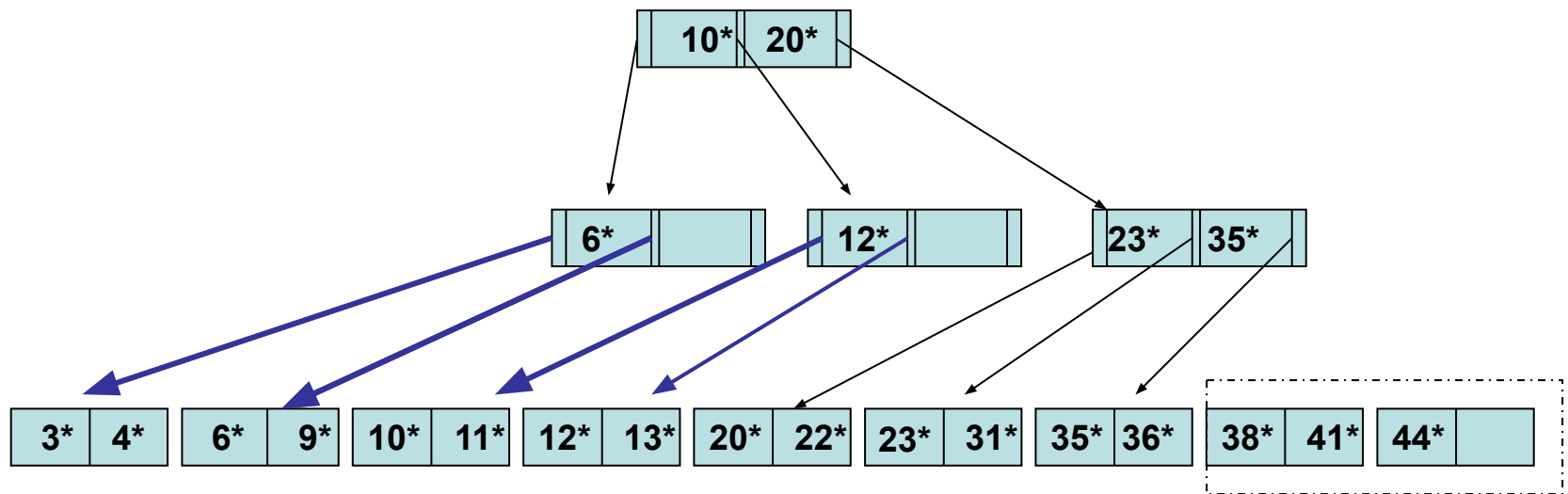


Páginas restantes a alocar

Bulking Load (Construção da B+ Tree)

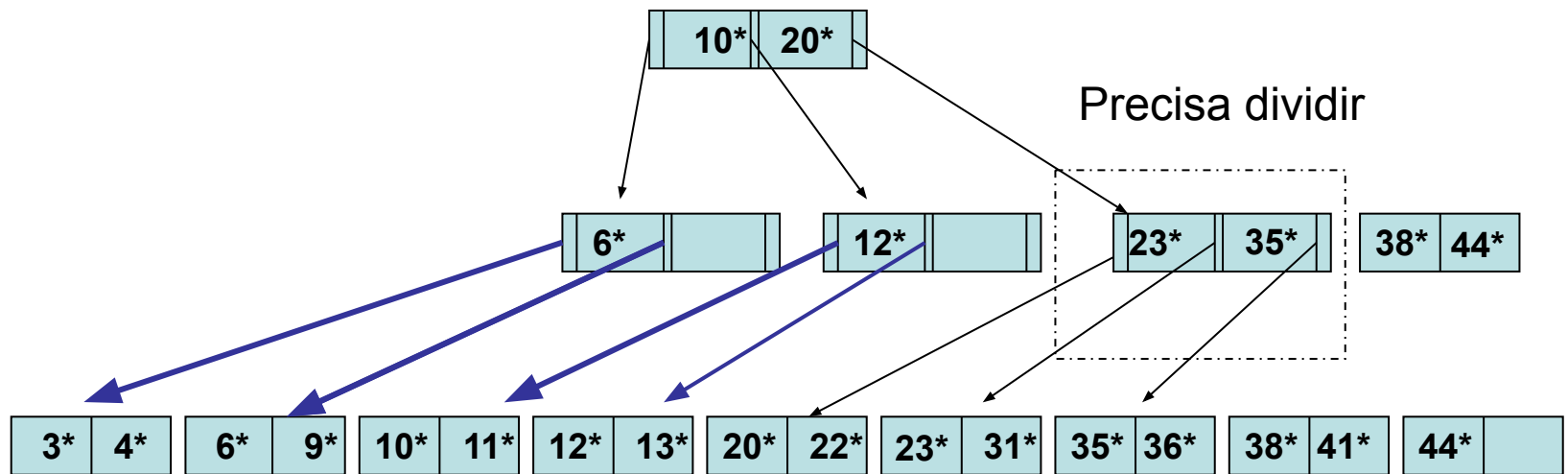


Bulking Load (Construção da B+ Tree)



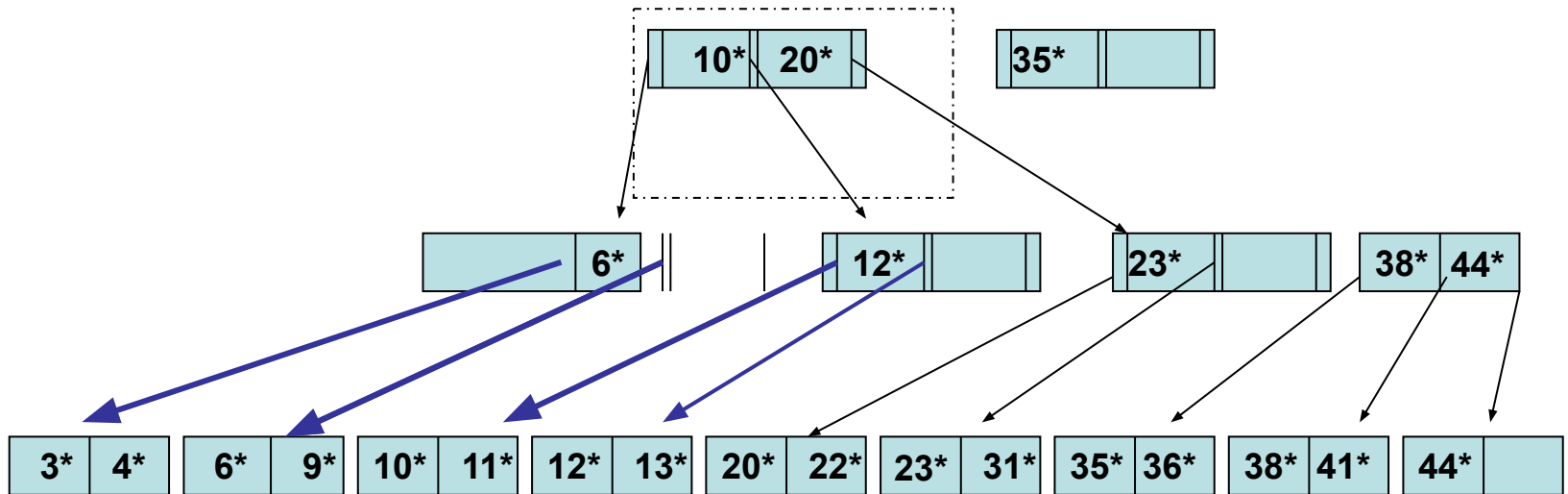
Páginas restantes a alocar

Bulking Load (Construção da B+ Tree)

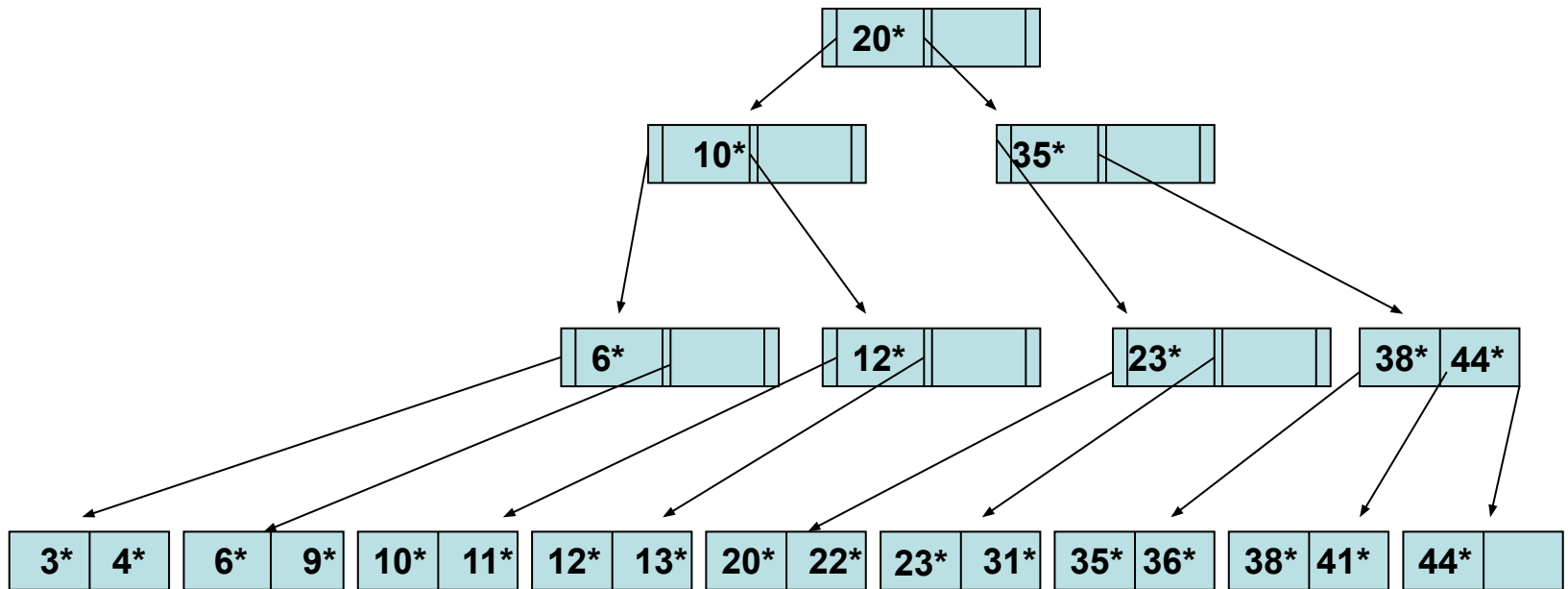


Bulking Load (Construção da B+ Tree)

Precisa dividir



Bulking Load (Construção da B+ Tree)



Árvore construída!

Atividade

- 1) Faça a leitura usando bulking load com a ordem 4 dos seguintes valores: 1 ,2 ,5 ,10 ,15 , 20, 40,60, 80 ,100 ,120 ,140,

Índices

- Conclusões
 - Os dados são mais acessados que atualizados
 - Necessário existir uma estrutura auxiliar para melhorar o desempenho das consultas
 - Para dados relacionais árvores B+ são os índices mais utilizados
 - O Otimizador de consultas utiliza índices sempre que possível