

Universidade Federal de São Carlos
Departamento de Computação

Estruturas de índices de banco de dados

Estrutura de dados - CCO510
Prof. Dr. Paulo Matias

Antônio M. dos S. Almeida Neto, André Ribeiro de Brito
{antonio.neto, andrerbrito}@ufscar.br

Contextualização

- As aplicações necessitam de alguma forma de armazenar as informações de modo permanente
- Essas informações tem por objetivo serem utilizadas em um momento futuro

Contextualização

Banco de dados

- Forma de armazenamento de dados em memória secundária
- Informações armazenadas em forma de linhas e colunas
- Forma padronizada de pesquisa nesses dados



Contextualização

Banco de dados

ORACLE®

DATABASE



Contextualização

Banco de dados

Log	Result1						
	TicketEventID	TicketID	Change Text	EventDateTime	TicketLoginID	Event Text	IsEmailEvent
	79	14	Opened	4/9/2007 7:11:3...	2	Create a new tabl...	<input type="checkbox"/>
	92	14	Assigned to Evg...	4/9/2007 7:43:4...	2	Please review th...	<input type="checkbox"/>
▶	140	23	Opened	4/13/2007 6:00:...	2	We discussed thi...	<input type="checkbox"/>
	172	14	Assigned to Mik...	4/17/2007 1:38:...	3	It sounds like a pr...	<input type="checkbox"/>
	211	34	Opened	4/19/2007 11:19:...	3	This possibility w...	<input type="checkbox"/>
	215	34	Edited	4/19/2007 2:32:...	3		<input type="checkbox"/>
	243	43	Opened	4/23/2007 9:03:...	4	Create a table wit...	<input type="checkbox"/>
	245	44	Opened	4/23/2007 9:04:...	4	For a given table ...	<input type="checkbox"/>
	247	45	Opened	4/23/2007 9:11:...	4	This is a post fro...	<input type="checkbox"/>
	300	55	Opened	4/25/2007 9:36:...	3	http://www.vista...	<input type="checkbox"/>

Contextualização

Problema

- A quantidade de registros pode ser consideravelmente grande
- A pesquisa em uma tabela da base é feita de forma sequencial
- O *join* (junção) entre tabelas com milhares de registro pode ser um problema

- Estrutura a coluna que possui o índice em uma estrutura específica
- Essas estruturas, de modo geral são:
 - Grafo
 - Hash

- Vantagem
 - Forma mais eficiente para recuperar informações
- Desvantagem
 - Custo maior nas operações de *Data Manipulation Language* (DML)

Contextualização

Índices

- Uma implementação simples de um índice seria utilizando uma árvore binária

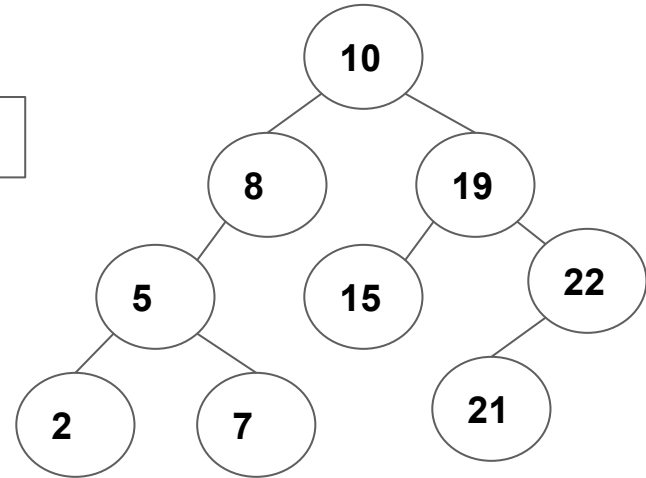
10	8	19	5	2	15	22	21	7
----	---	----	---	---	----	----	----	---

Contextualização

Índices

- Uma implementação simples de um índice seria utilizando uma árvore binária

10	8	19	5	2	15	22	21	7
----	---	----	---	---	----	----	----	---



PostgreSQL

- Sistema Gerenciador de Banco de Dados (SGBD)
- Desenvolvido inicialmente pela Universidade da Califórnia em Berkeley
- Modelo Relacional
- Propriedades ACID
- A primeira versão de demonstração foi lançada em 1985

PostgreSQL

- Desde 1995 está sendo desenvolvido pela comunidade, atualmente sendo gerenciado pela PostgreSQL Global Development Group
- Open Source, utilizando uma licença derivada da BSD
- A sua última versão (10.4) foi lançada em maio de 2018



- Suporta os seguintes tipos de índices
 - B-tree
 - Hash
 - GiST (**G**eneralized **S**earch **T**ree)
 - SP-GiST (**S**pace-**P**artitioned GiST)
 - GIN (**G**eneralized **I**nverted **I**ndex)
 - BRIN (**B**lock **R**ange **I**ndex)

Generalized Inverted Index

- Utilizado para indexar strings
- Alternativa a busca sequencial por palavras

Generalized Inverted Index

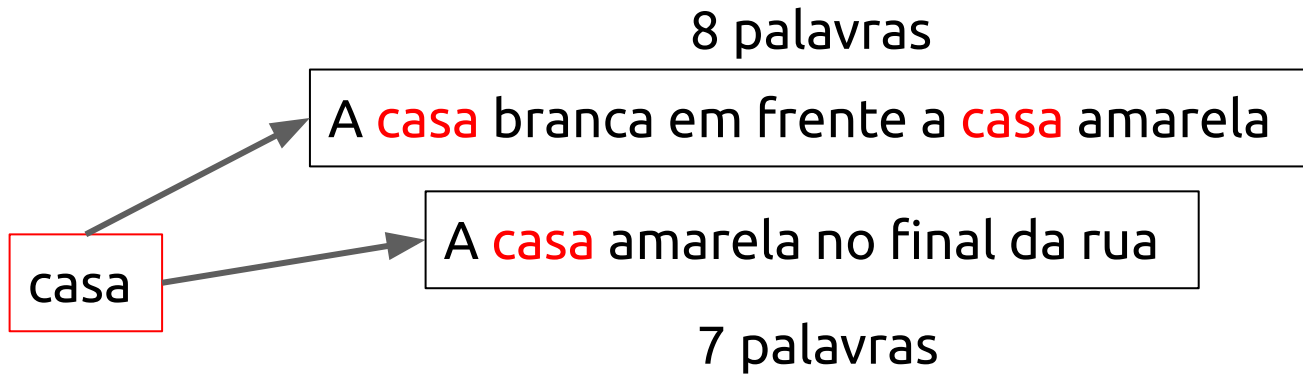
- Utilizado para indexar strings
- Alternativa a busca sequencial por palavras

A casa branca em frente a casa amarela

A casa amarela no final da rua

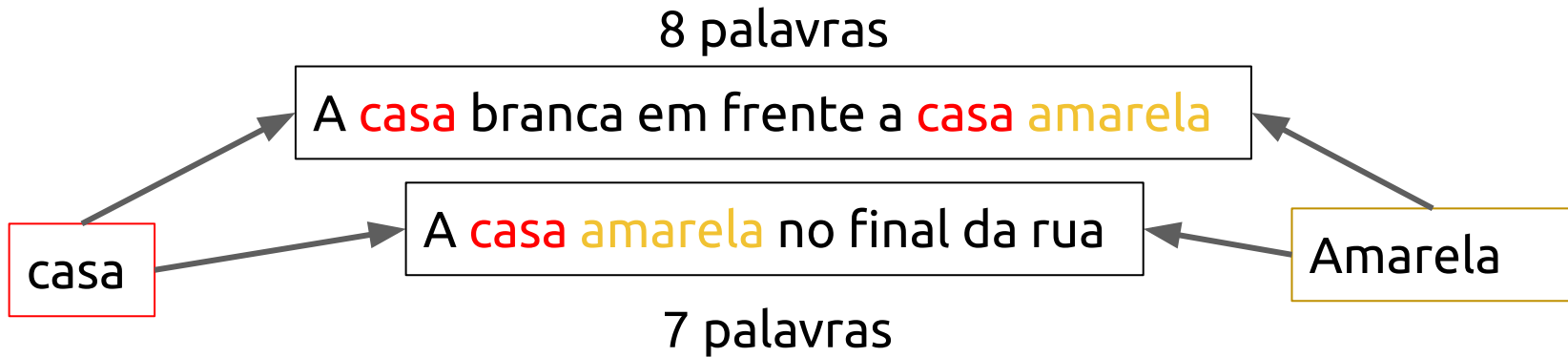
Generalized Inverted Index

- Utilizado para indexar strings
- Alternativa a busca sequencial por palavras



Generalized Inverted Index

- Utilizado para indexar strings
- Alternativa a busca sequencial por palavras



Generalized Inverted Index

- Necessita de um pré-processamento
- Contém duas partes:
 - Vocabulário ou Dicionário
 - Lista de ocorrências

Generalized Inverted Index

- Necessita de um pré-processamento
- Contém duas partes:
 - Vocabulário ou Dicionário
 - Lista de ocorrências

A casa branca em frente a casa amarela	Documento 1
A casa amarela no final da rua	Documento 2

Generalized Inverted Index

Vocabulário	Documento 1	Documento 2
casa	2	1
branca	1	-
frente	1	-
amarela	1	1
final	-	1
rua	-	1

Generalized Inverted Index

- Granularidade
 - Grossa - Identifica apenas um bloco de texto
 - Moderada - Identifica apenas o documento
 - Fina - Identifica cada palavra

Generalized Inverted Index

Granularidade fina

- Para cada palavra, é necessário armazenar:
 - O número de ocorrências em todos os documentos
 - O documento
 - A quantidade de ocorrências para cada documento
 - A posição da palavra para cada ocorrência em cada documento

Generalized Inverted Index

Granularidade fina

- Para cada palavra, é necessário armazenar além o documento, o número de ocorrências a sua respectiva posição.

A casa branca em frente a casa amarela

A casa amarela no final da rua

casa	3; (1; 2; <2, 7>), (2; 1; <2>)
amarela	2; (1; 1; <8>), (2; 1; <3>)
rua	1; (2; 1; <7>)

B-Tree

- São árvores balanceadas, projetadas para trabalhar com dispositivos de armazenamento secundário como discos magnéticos.
- Generalização da árvore binária de pesquisa.
- Múltiplas chaves em cada nó e mais de dois filhos para cada nó.

B-Tree

Propriedades

- Em uma árvore B de ordem m , tem-se:
 - Páginas contém entre 1 e $m-1$ itens;
 - Cada registro aponta para dois nós filhos;
 - itens: aparecem dentro de uma página em ordem crescente, de acordo com suas chaves, da esquerda para a direita.

B-Tree

Operações em tempo logarítmico

- Inserção
- Remoção
- Pesquisar

B-Tree

Inserção

- ❖ Inserção: **10**,40,15,80,52,60,70,63,22,2,8,1.
- ❖ $M=4$;

10

B-Tree

Inserção

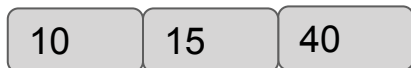
- ❖ Inserção: 10, **40**, 15, 80, 52, 60, 70, 63, 22, 2, 8, 1.
- ❖ $M=4$;



B-Tree

Inserção

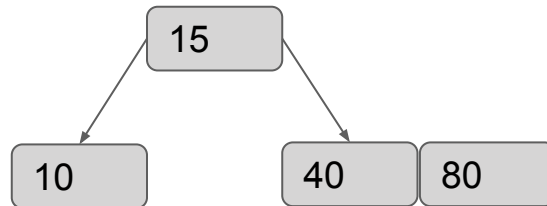
- ❖ Inserção: 10, 40, **15**, 80, 52, 60, 70, 63, 22, 2, 8, 1.
- ❖ $M=4$;



B-Tree

Inserção

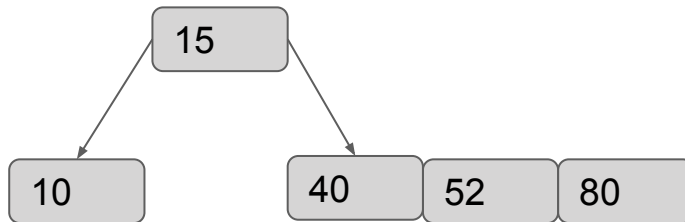
- ❖ Inserção: 10, 40, 15, **80**, 52, 60, 70, 63, 22, 2, 8, 1.
- ❖ $M=4$;



B-Tree

Inserção

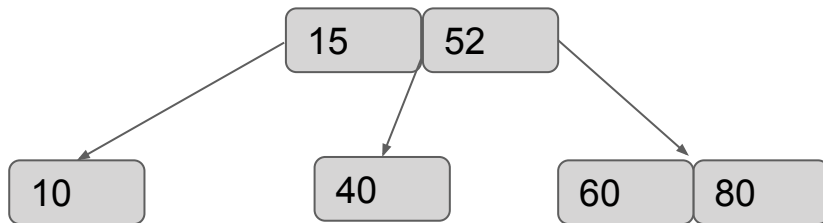
- ❖ Inserção: 10, 40, 15, 80, **52**, 60, 70, 63, 22, 2, 8, 1.
- ❖ $M=4$;



B-Tree

Inserção

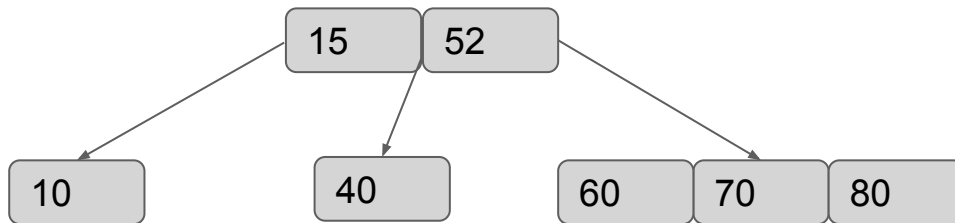
- ❖ Inserção: 10, 40, 15, 80, 52, **60**, 70, 63, 22, 2, 8, 1.
- ❖ $M=4$;



B-Tree

Inserção

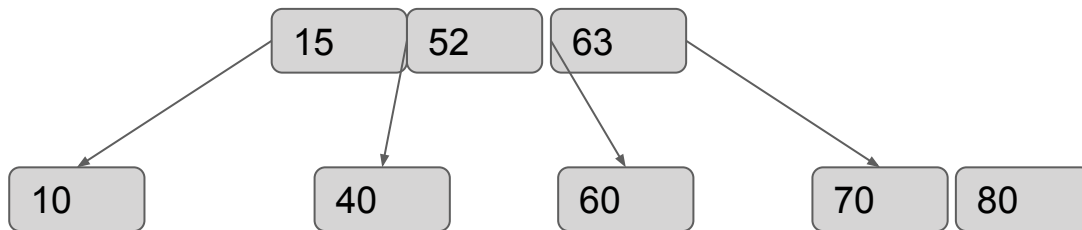
- ❖ Inserção: 10, 40, 15, 80, 52, 60, **70**, 63, 22, 2, 8, 1.
- ❖ $M=4$;



B-Tree

Inserção

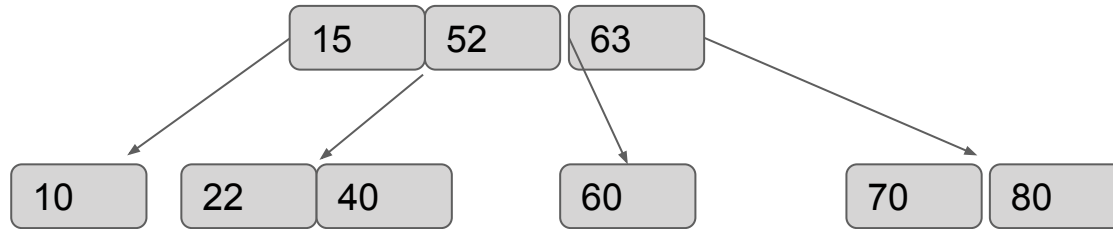
- ❖ Inserção: 10, 40, 15, 80, 52, 60, 70, **63**, 22, 2, 8, 1.
- ❖ $M=4$;



B-Tree

Inserção

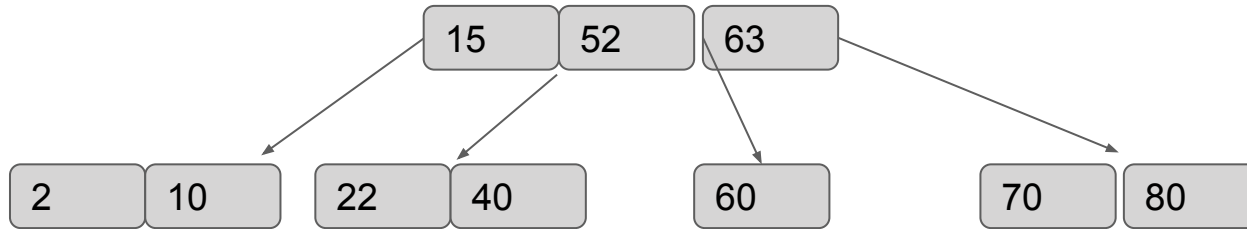
- ❖ Inserção: 10, 40, 15, 80, 52, 60, 70, 63, **22**, 2, 8, 1.
- ❖ $M=4$;



B-Tree

Inserção

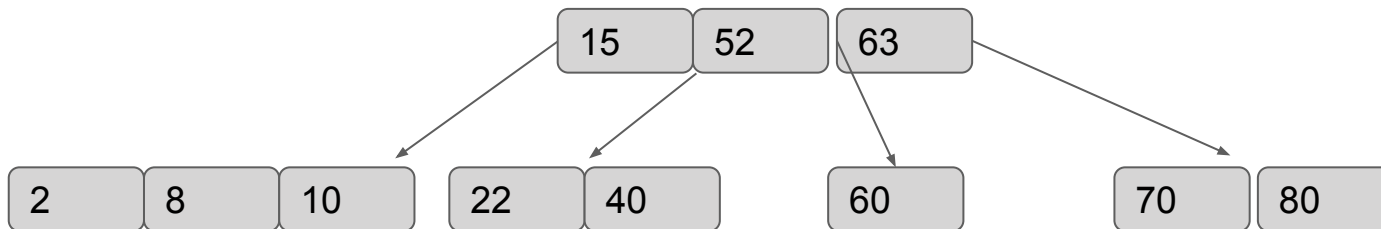
- ❖ Inserção: 10, 40, 15, 80, 52, 60, 70, 63, 22, **2**, 8, 1.
- ❖ $M=4$;



B-Tree

Inserção

- ❖ Inserção: 10, 40, 15, 80, 52, 60, 70, 63, 22, 2, **8**, 1.
- ❖ $M=4$;

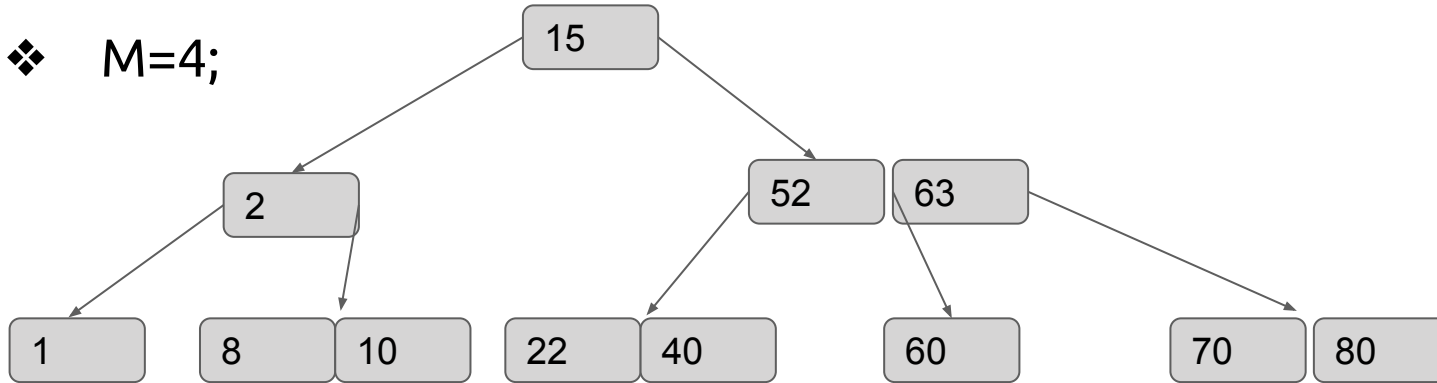


B-Tree

Inserção

❖ Inserção: 10, 40, 15, 80, 52, 60, 70, 63, 22, 2, 8, 1.

❖ $M=4$;

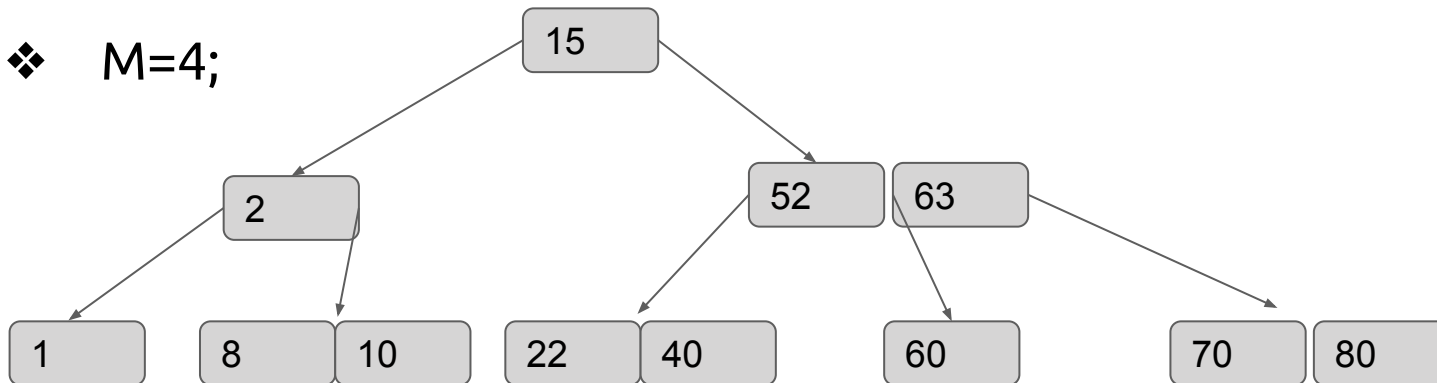


B-Tree

Remoção

❖ Remoção:15.

❖ $M=4$;

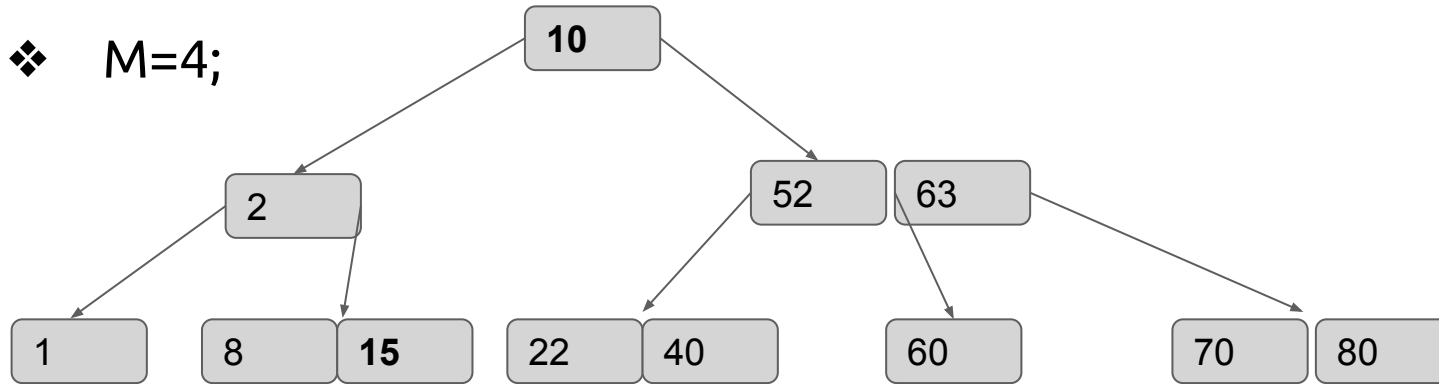


B-Tree

Remoção

❖ Remoção:15.

❖ $M=4$;

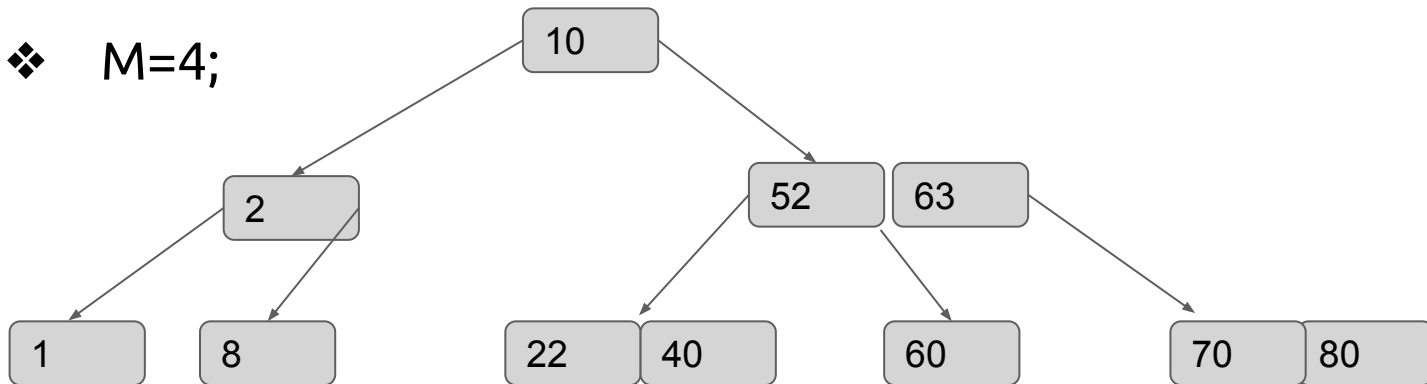


B-Tree

Remoção

❖ Remoção:15.

❖ $M=4$;

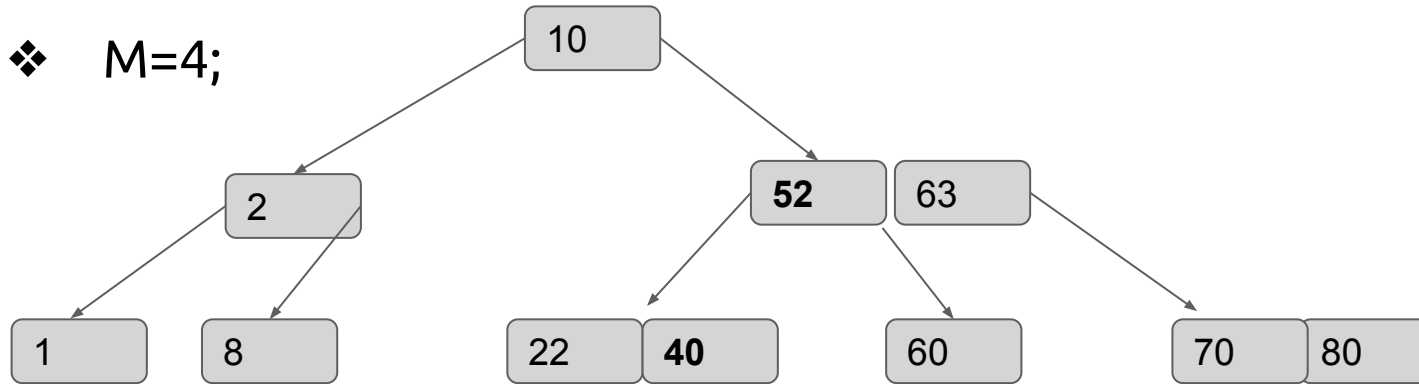


B-Tree

Remoção

❖ Remoção:52.

❖ $M=4$;

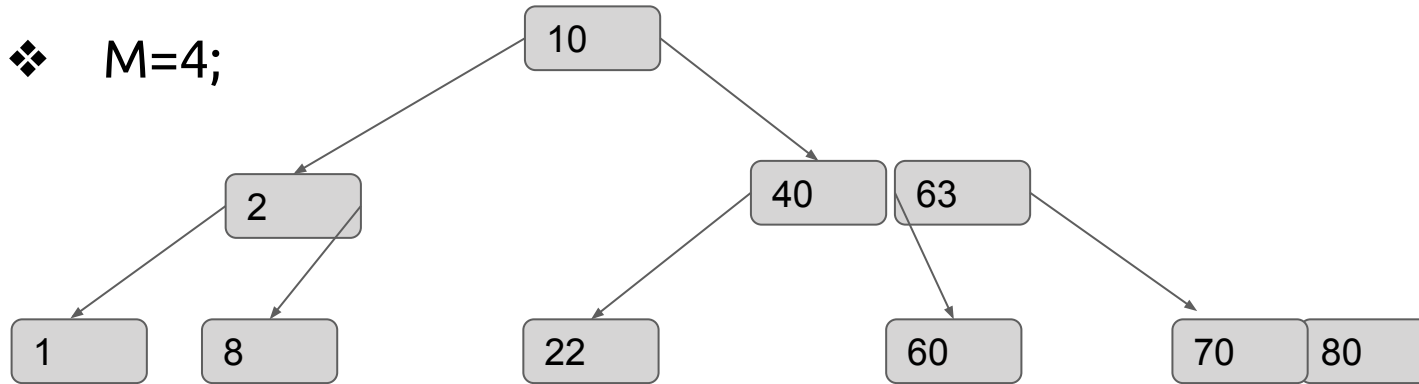


B-Tree

Remoção

❖ Remoção:52.

❖ $M=4$;

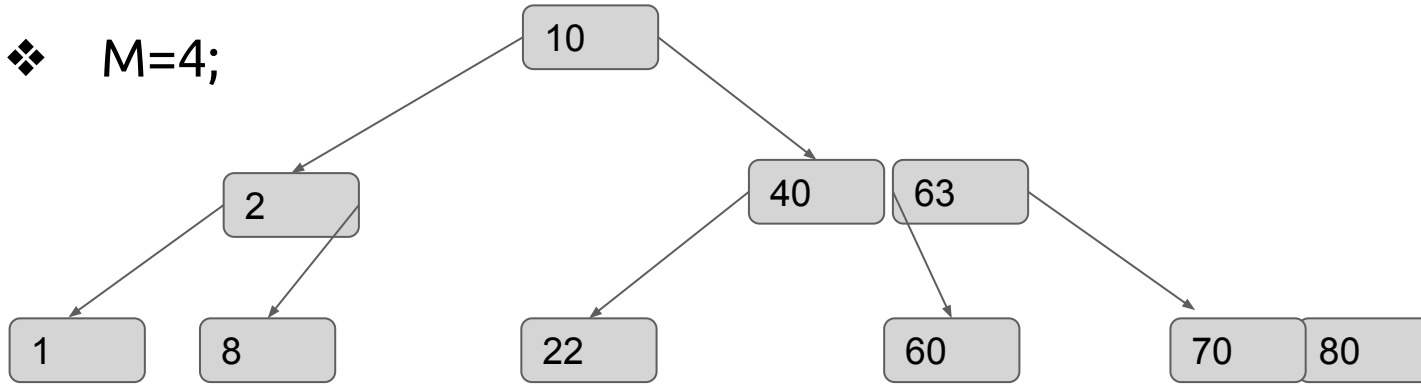


B-Tree

Remoção

❖ Remoção:2.

❖ $M=4$;

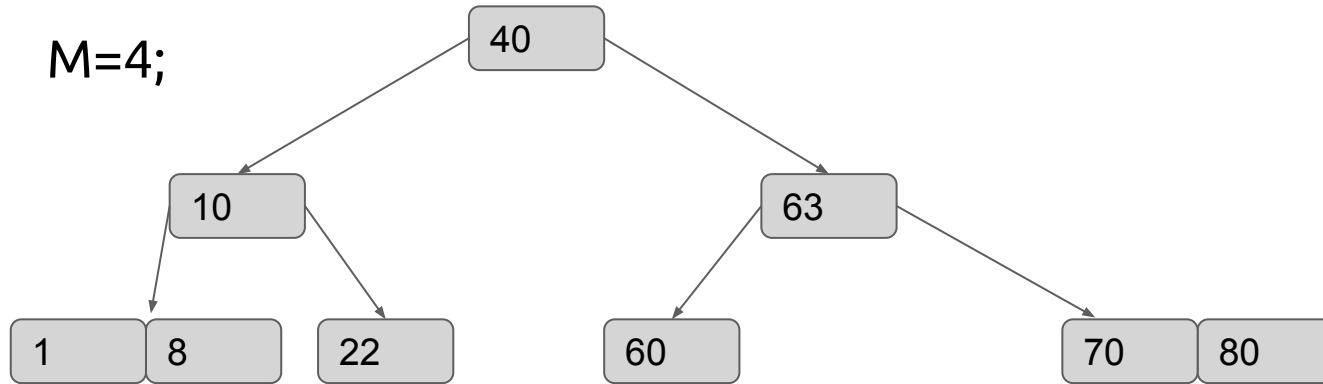


B-Tree

Remoção

❖ Remoção:2.

❖ $M=4$;



B-Tree

Árvore B

- Pesquisa: a operação de pesquisa é semelhante à da Árvore de pesquisa binária.

B-Tree

Pesquisa

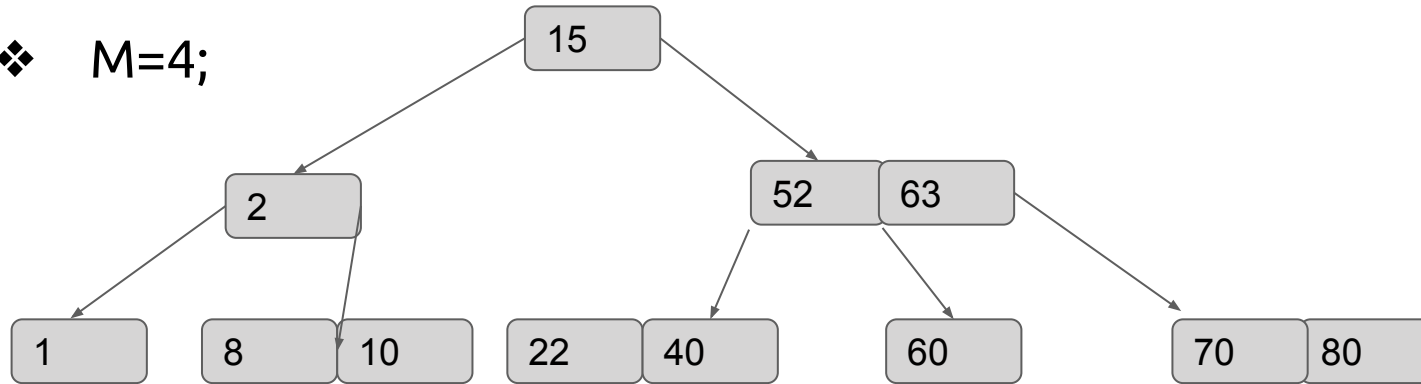
```
43 void Pesquisa(TipoRegistro *x, TipoApontador Ap) {  
44     long i = 1;  
45     if (Ap == NULL) {  
46         cout << "\n registro nao esta presente na arvore" << endl;  
47         return;  
48     }  
49     //pesquisa sequencial para encontrar o intervalo desejado  
50     while (i < Ap->n && x->chave > Ap->r[i - 1].chave) i++;  
51     if (x->chave == Ap->r[i - 1].chave) { // verifica se a chave foi localizada  
52         *x = Ap->r[i - 1];  
53         cout << "\n registro encontrado" << endl;  
54         return;  
55     } else if (x->chave < Ap->r[i - 1].chave) {  
56         cout << "esta a esquerda" << endl;  
57         Pesquisa(x, Ap->p[i - 1]);  
58     } else {  
59         cout << "esta a direita" << endl;  
60         Pesquisa(x, Ap->p[i]);  
61     }  
62 }  
63 }  
64 }  
65 }
```

B-Tree

Pesquisa

❖ Pesquisa:10.

❖ $M=4$;



Referências

- <http://www.unipac.br/site/bb/tcc/tcc-1aa82beefca34a5fce893238479fb.pdf>
- <http://homepages.dcc.ufmg.br/~nivio/cursos/ri05/transp/indexing.ps>
- http://www.lcad.icmc.usp.br/~nonato/ED/B_arvore/btree.htm
- <https://www.cs.usfca.edu/~galles/visualization/BTree.html>
- <https://www.postgresql.org/docs>