

Datové a procesní modely

[Relační databáze]

Přednáška 5 e_Btree

Marian Kamenický

Synte software group a.s.
marian.kamenicky@synte.cz

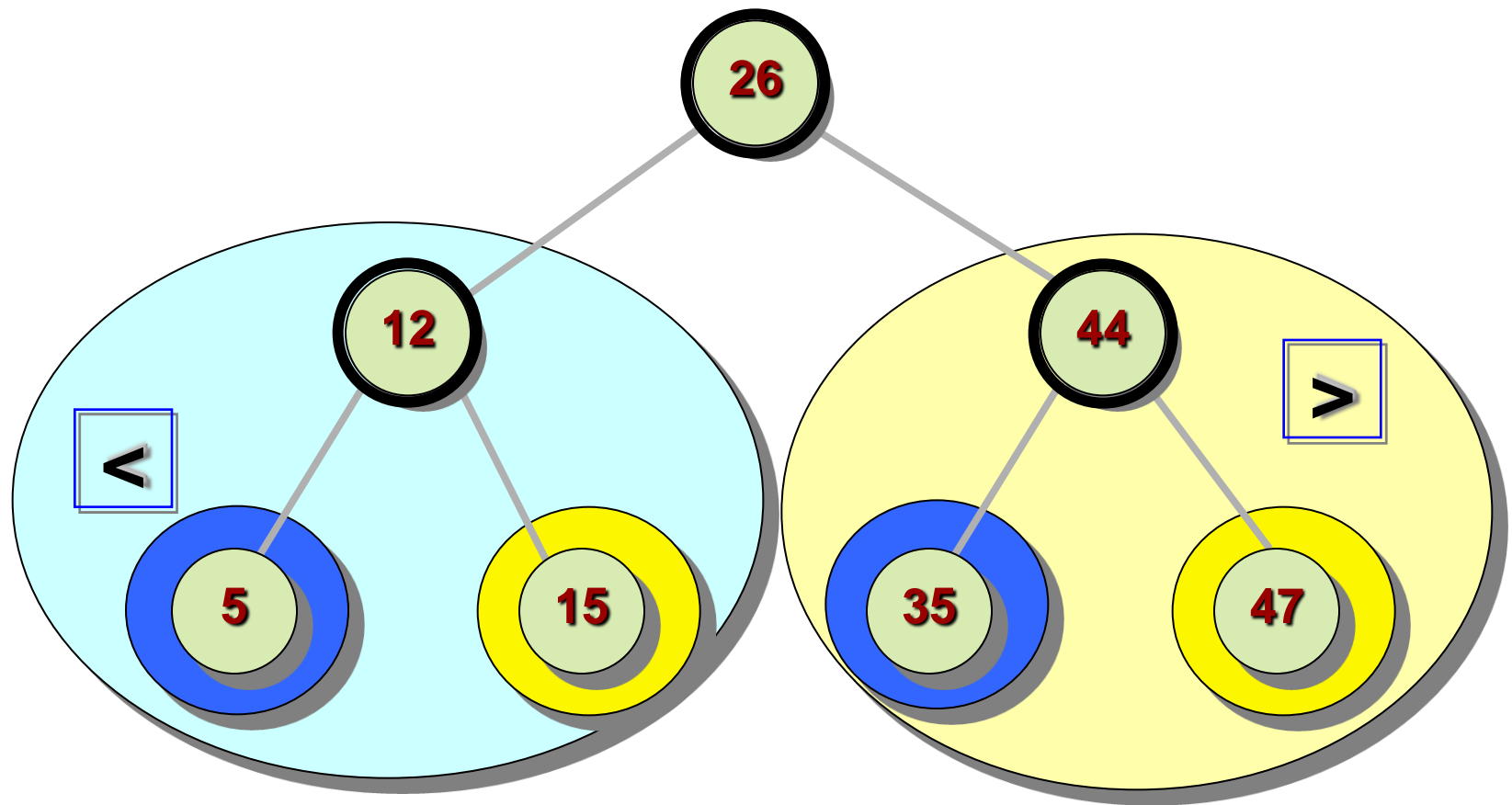
MFFUK Praha

2019/20

Typy indexů

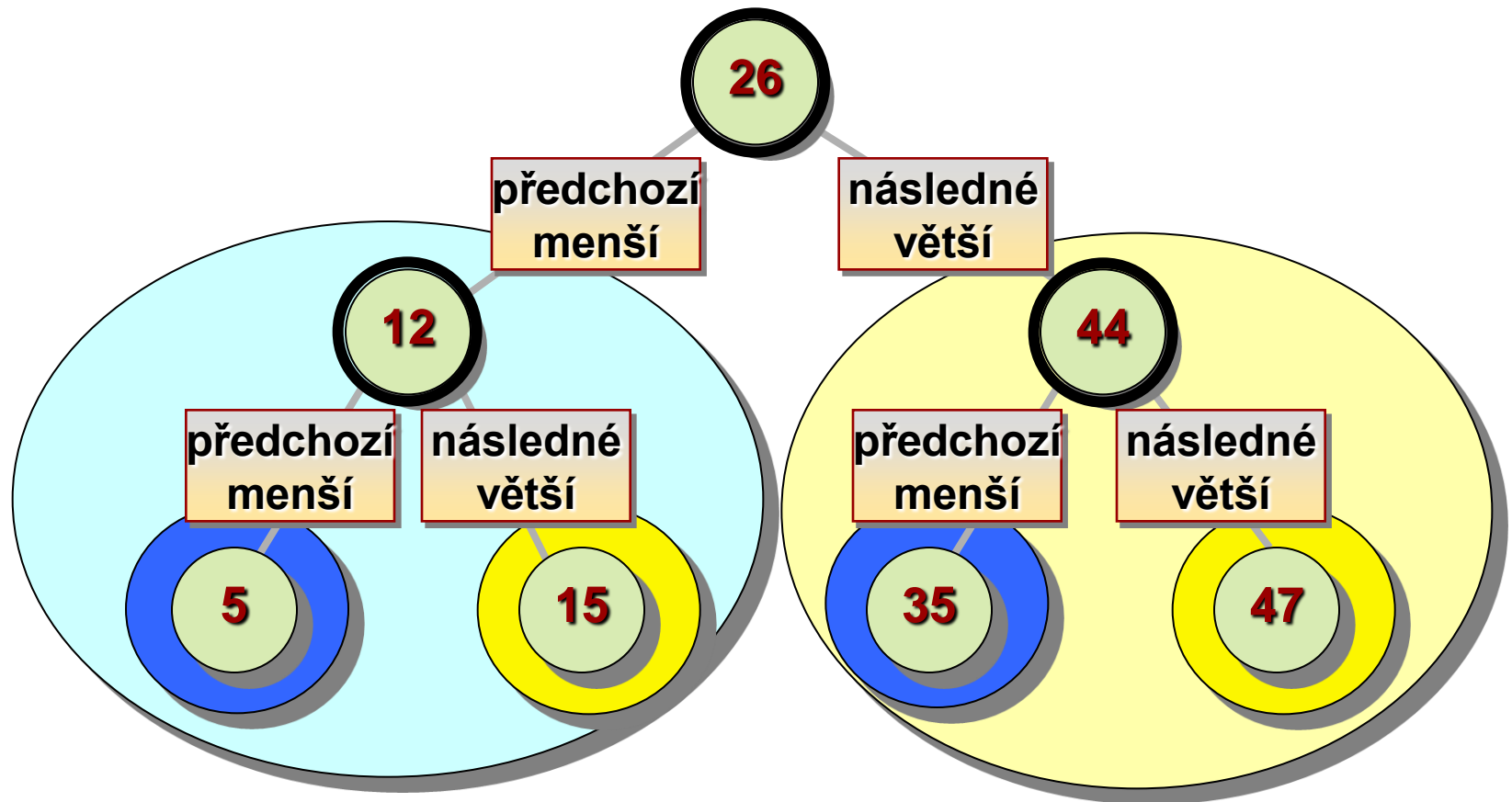
- **binární stromy**
- b-stromy
- r-stromy
- bitové mapy
- funkční indexy
- hash indexy
- doménové indexy
- členěné indexy
- klastrované indexy
- prostorové indexy
- IOT index only tables

A Binary Search Tree



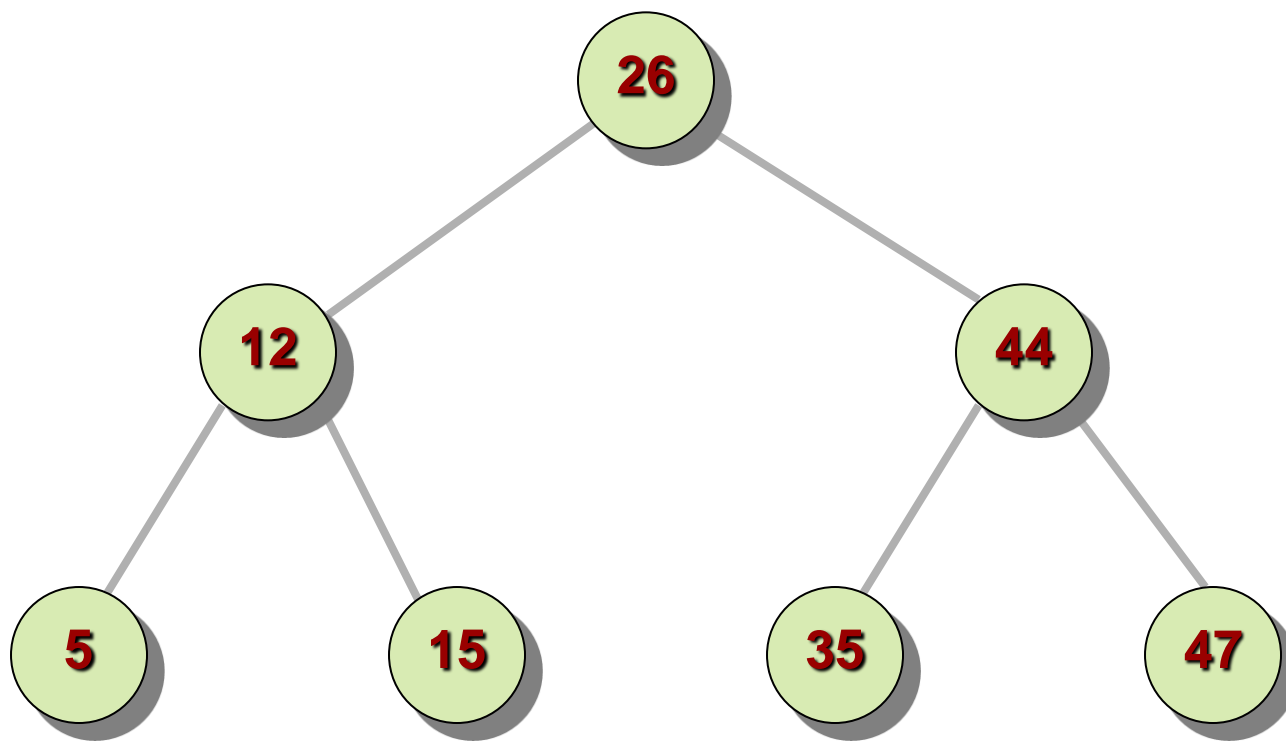
- UZEL
- **levý** substrom obsahuje **menší** hodnoty
- **pravý** substrom obsahuje **větší** hodnoty

A Binary Search Tree



- UZEL
- **levý** substrom obsahuje **menší** hodnoty
- **pravý** substrom obsahuje **větší** hodnoty

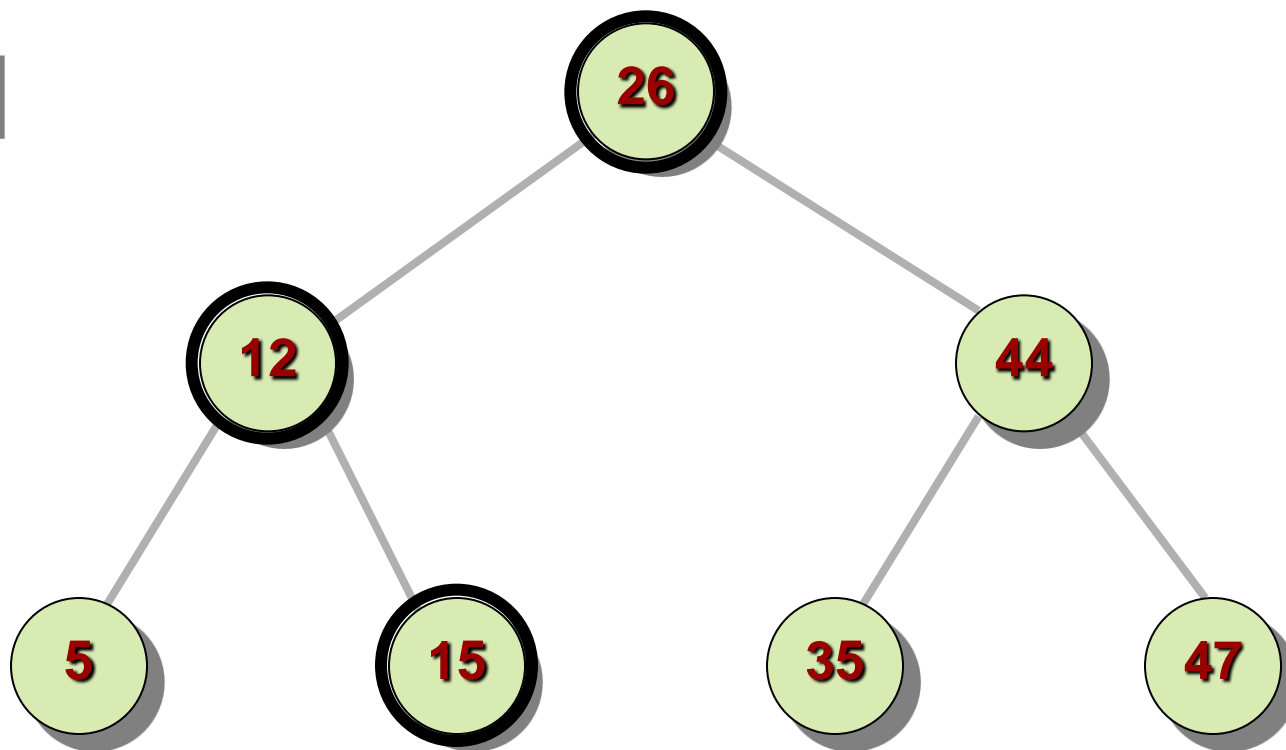
A Binary Search Tree - hledání hodnoty



- Máme **hodnotu** a hledáme ji ve stromu
- $Hodnota = Uzel$ --> **nalezeno**
- $Hodnota < Uzel$ --> jdi **levým** substromem [tam menší]
- $Hodnota > Uzel$ --> jdi **pravým** substromem [tam větší]

A Binary Search Tree - hledání hodnoty 15

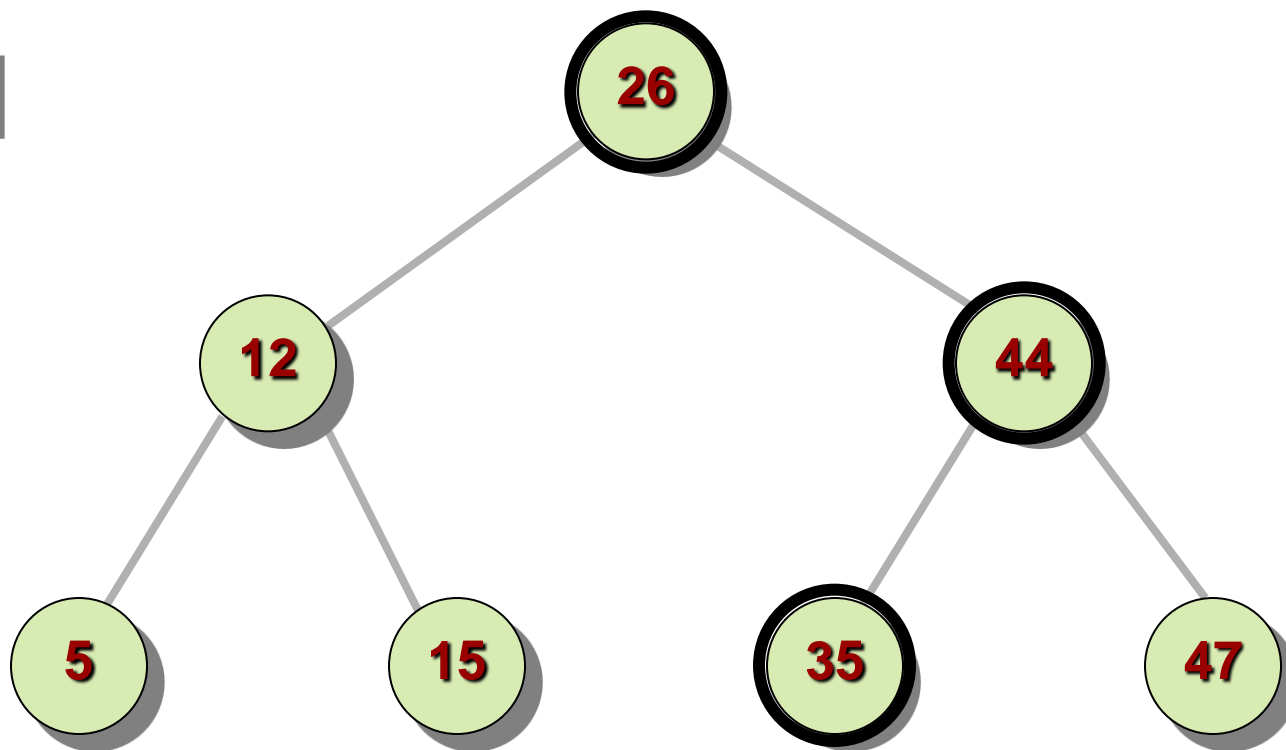
15



- hledáme 15
- $15 < 26$ --> jdi levým substromem
- $15 > 12$ --> jdi pravým substromem
- $15 = 15$ --> nalezeno

A Binary Search Tree - hledání hodnoty 37

37

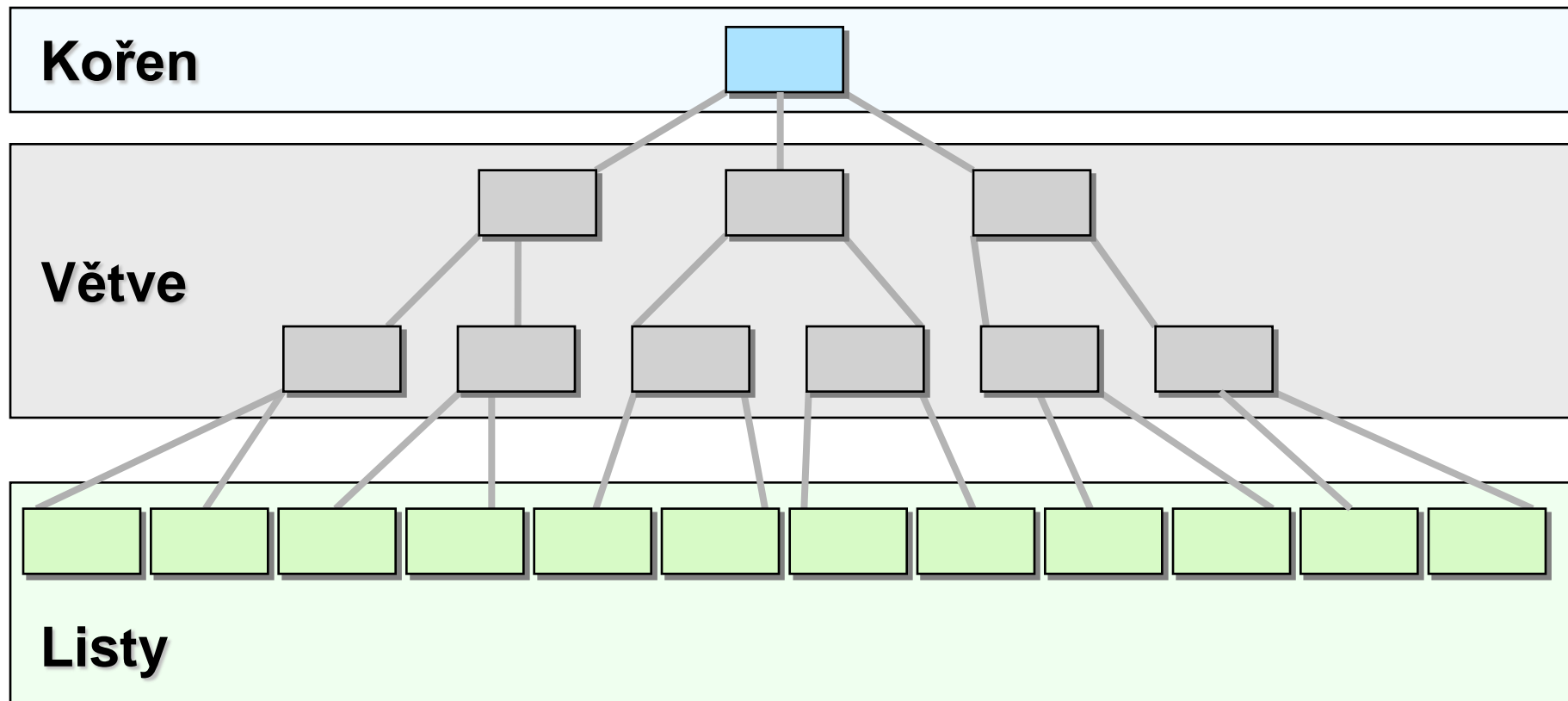


- hledáme 37
- $37 > 26$ --> jdi pravým substromem
- $37 < 44$ --> jdi levým substromem
- $37 > 35$ --> jdi pravým substromem
- žádný není --> **nenalezeno**

Typy indexů

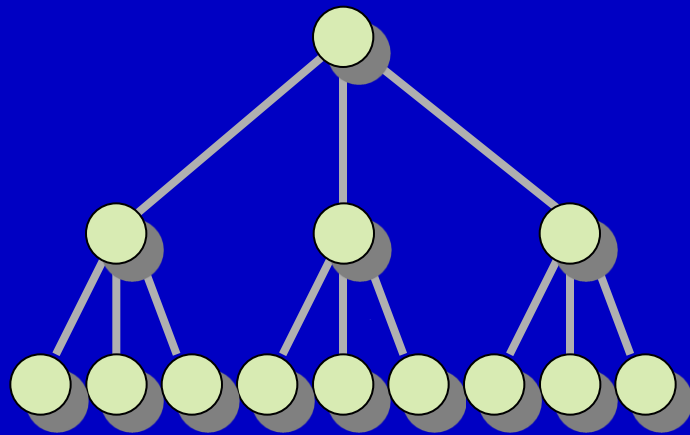
- binární stromy
- **b-stromy**
- r-stromy
- bitové mapy
- funkční indexy
- hash indexy
- doménové indexy
- členěné indexy
- klastrované indexy
- prostorové indexy
- IOT index only tables

B stromy [B-tree] – tři druhy uzlů

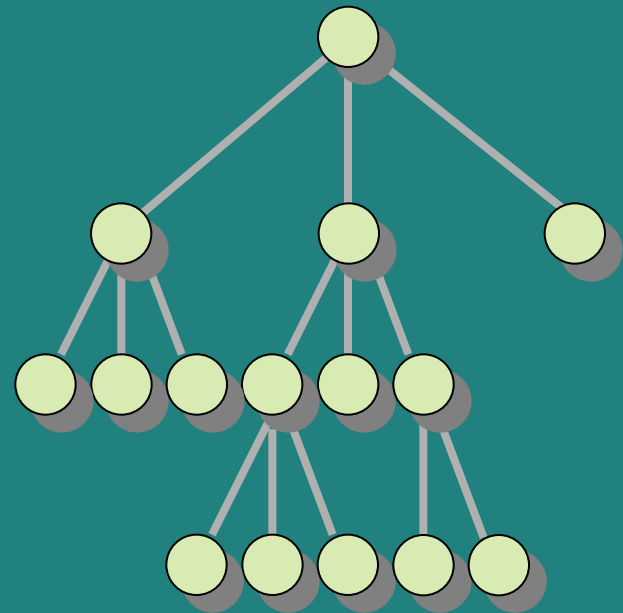


B stromy [B-tree] - pravidla

- všechny listy stromu jsou na stejné úrovni (hloubce)
 - B-strom **je balancovaný (vyvážený)**

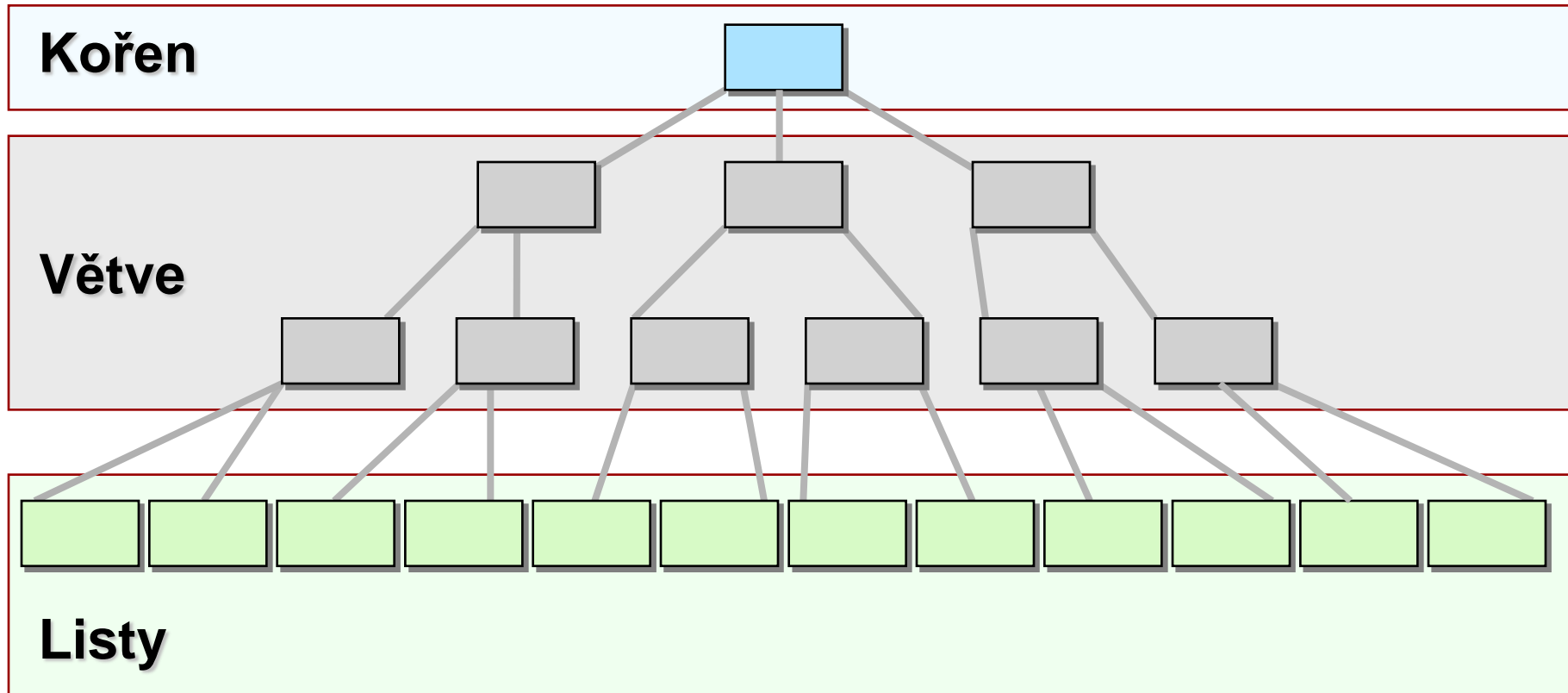


balancovaný



nebalancovaný

B stromy [B-tree]



B stromy [B-tree] - pravidla

- strom je řádu n
- uzel má minimálně n klíčů
- uzel má maximálně $n*2$ klíčů
- uzel má maximálně $n*2+1$ ukazatelů stromu
- má-li uzel p klíčů, má $p+1$ ukazatelů stromu
- kořenový uzel může mít méně klíčů
 - dosud nezaplňen
- všechny listy stromu jsou na stejné úrovni
- ukazatele stromu listů jsou **nulové**

B stromy [B-tree]

- strom je řádu n
- uzel má minimálně n klíčů
- uzel má maximálně $n*2$ klíčů Joe 38
- uzel má maximálně $n*2+1$ ukazatelů stromu
- má-li uzel p klíčů, má $p+1$ ukazatelů stromu



ukazatel stromu

klíče Joe, 38, ...

■ min počet klíčů v uzlu

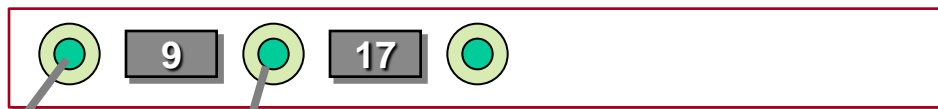
N	2
---	---

■ max počet klíčů v uzlu

N*2	4
-----	---

■ max počet ukazatelů stromu

N*2+1	5
-------	---



klíče < 9

pravý ukazatel 9 - ukazuje na větší > 9

9 < klíče < 17

levý ukazatel 17 - ukazuje na menší < 17

B stromy [B-tree]

- strom je řádu n
- uzel má minimálně n klíčů
- uzel má maximálně $n*2$ klíčů Joe 38
- uzel má maximálně $n*2+1$ ukazatelů stromu
- má-li uzel p klíčů, má $p+1$ ukazatelů stromu



ukazatel stromu

klíče Joe, 38, ...

■ min počet klíčů v uzlu

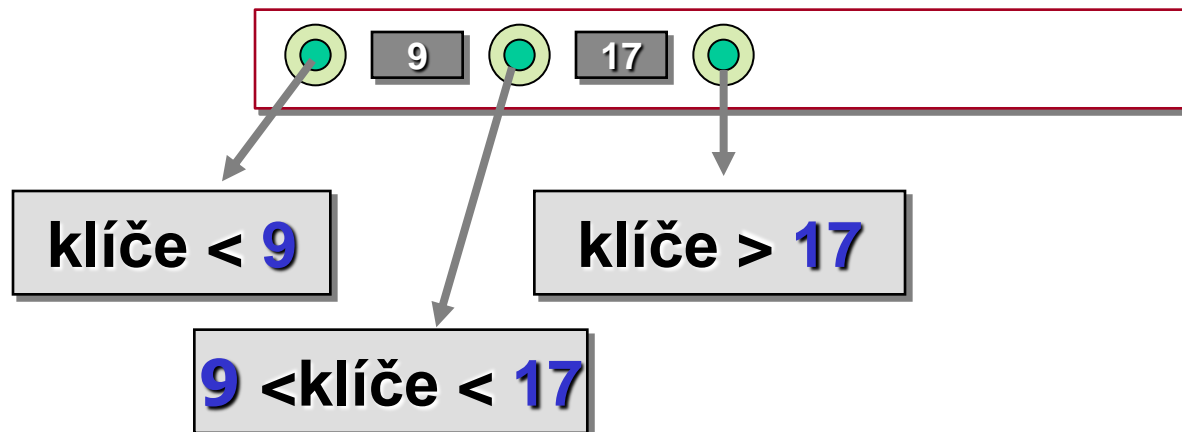
N	2
-----	---

■ max počet klíčů v uzlu

$N*2$	4
-------	---

■ max počet ukazatelů stromu

$N*2+1$	5
---------	---



B stromy [B-tree]

- strom ke řádu n
- uzel má minimálně n klíčů
- uzel má maximálně $n*2$ klíčů
- uzel má maximálně $n*2+1$ ukazatelů stromu
- má-li uzel p klíčů, má $p+1$ ukazatelů stromu



ukazatel stromu

Joe

38

klíče Joe, 38, ...

■ min počet klíčů v uzlu

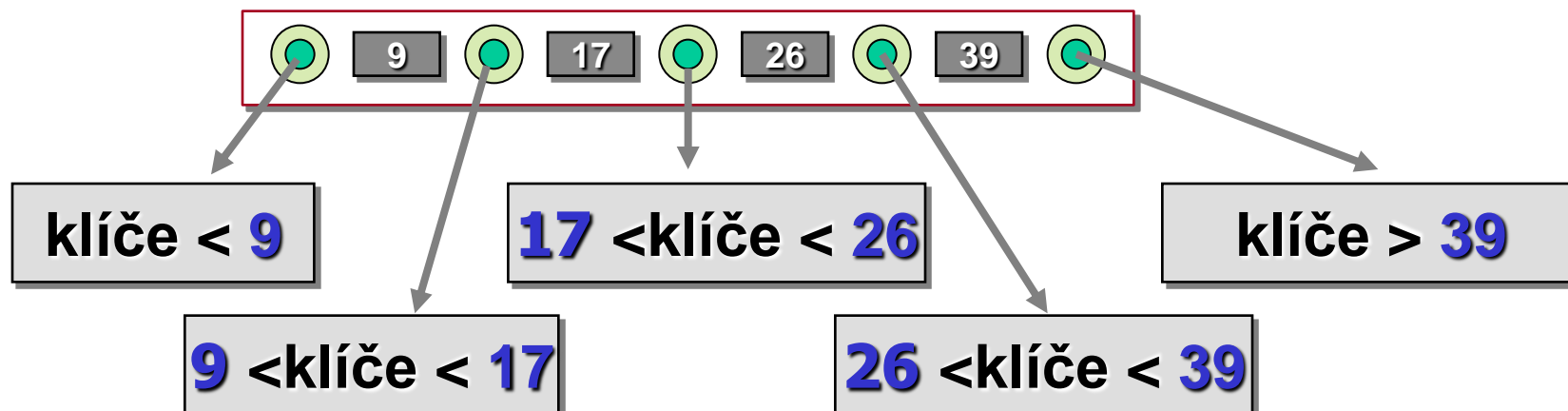
N **2**

■ max počet klíčů v uzlu

N*2 **4**

■ max počet ukazatelů stromu

N*2+1 **5**



B stromy [B-tree]

- strom ke řádu n
- uzel má minimálně n klíčů
- uzel má maximálně $n*2$ klíčů
- uzel má maximálně $n*2+1$ ukazatelů stromu
- má-li uzel p klíčů, má $p+1$ ukazatelů stromu



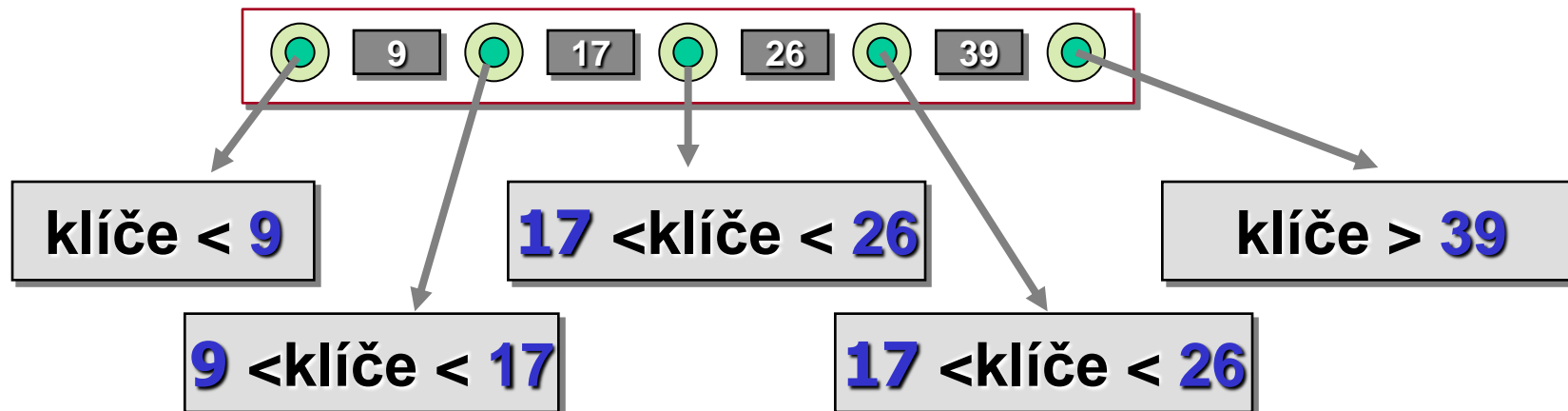
ukazatel stromu

Joe

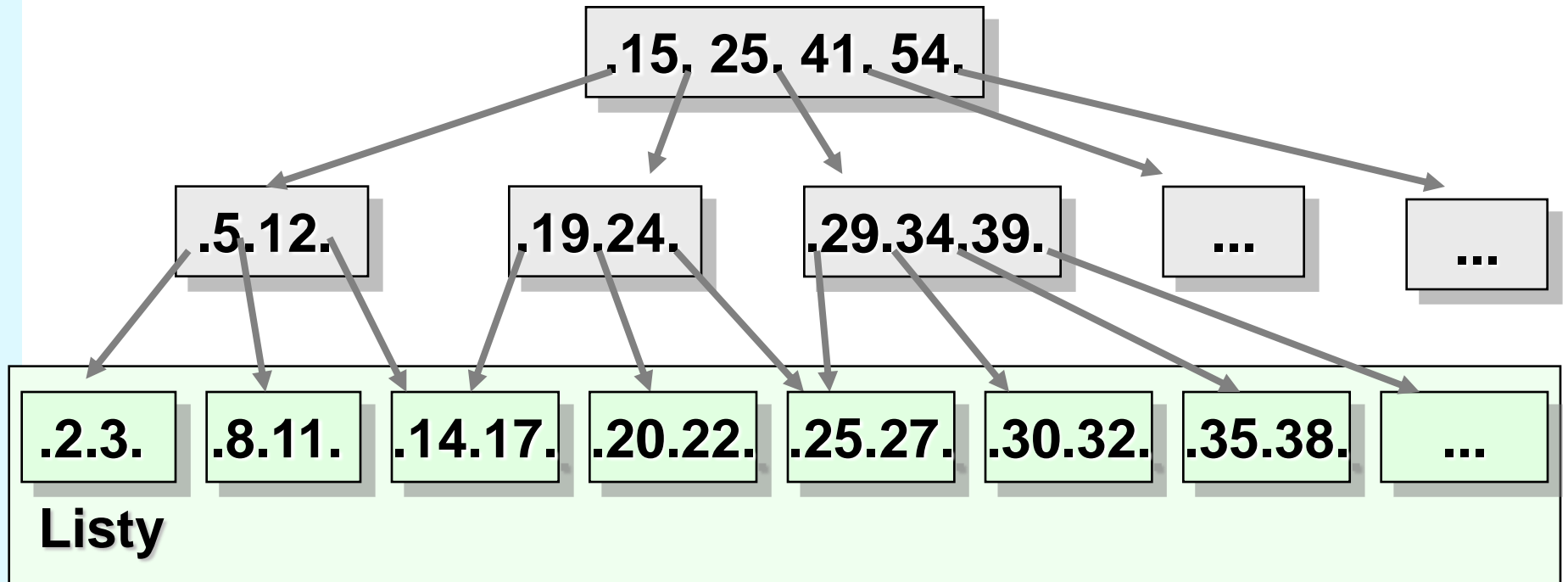
38

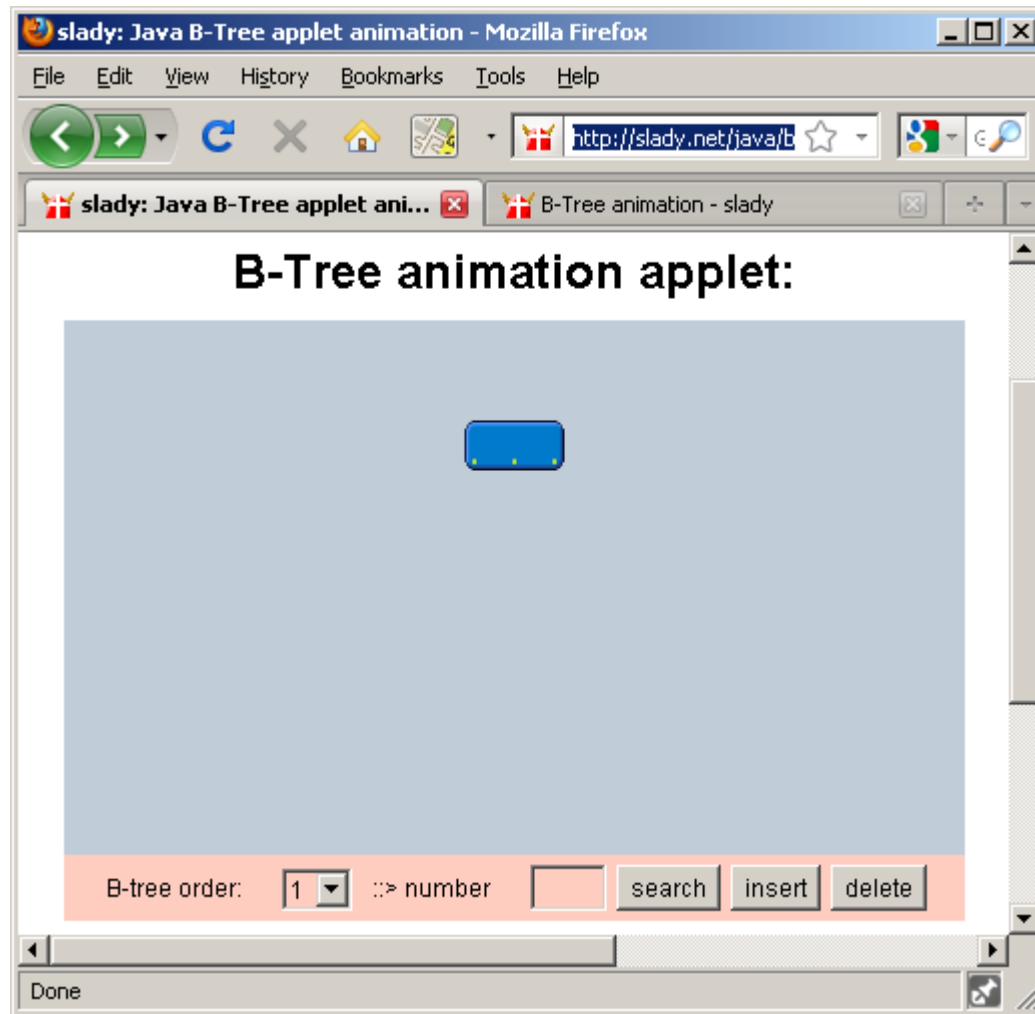
klíče Joe, 38, ...

■ min počet klíčů v uzlu	N	2	$N/2$	$N-1/2$
■ max počet klíčů v uzlu	$N*2$	4	N	$N-1$
■ max počet ukazatelů stromu	$N*2+1$	5	$N+1$	N



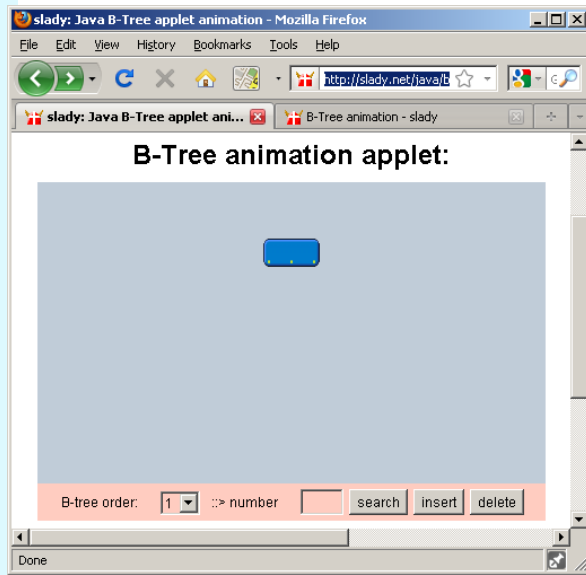
B stromy [B-tree]



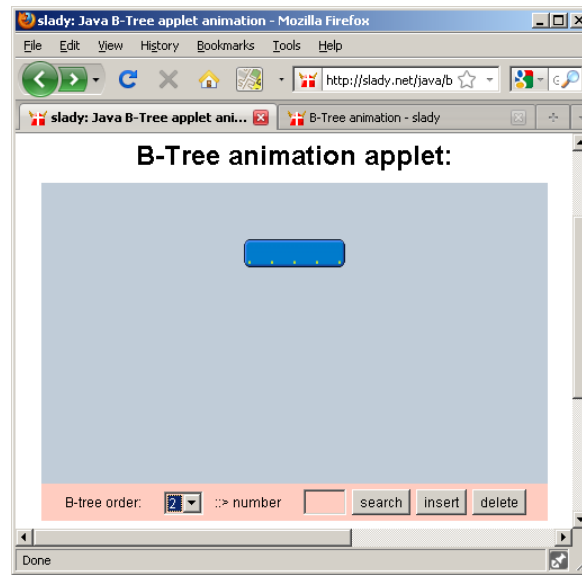


**pokud obrázek ušetří tisíc slov
pak animace ušetří tisíc obrázků**

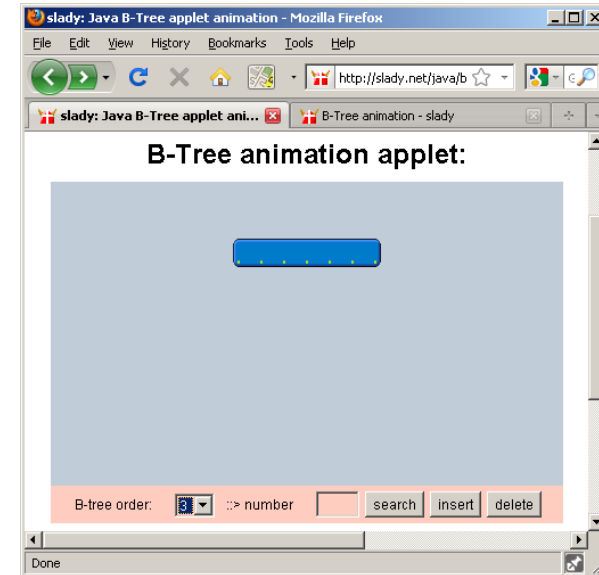
BTree - animace



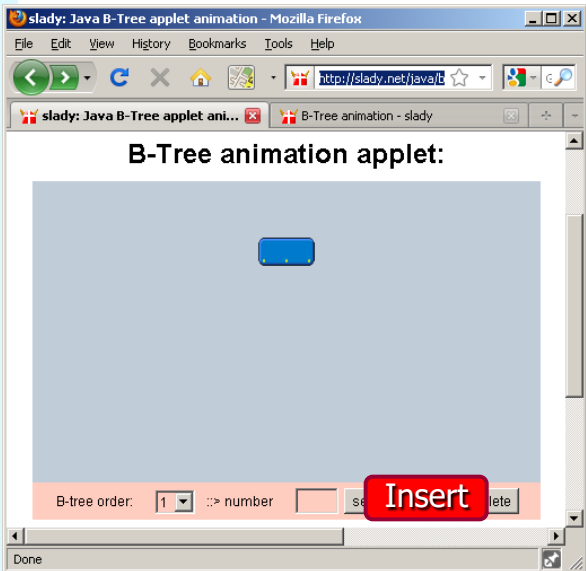
B-tree order : 1



B-tree order : 2



B-tree order : 3



- nejprve budeme B-strom vytvářet
- budeme do něho čísla vkládat
 - napíšeme číslo
 - klikneme na INSERT

BTree

- vložíme číslo 10

slady: Java B-Tree applet animation - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://slady.net/java/b

slady: Java B-Tree applet ani... B-Tree animation - slady

B-Tree animation applet:

10

B-tree order: 1 ::> number 10 search insert delete

Done

slady: Java B-Tree applet animation - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://slady.net/java/b

slady: Java B-Tree applet ani... B-Tree animation - slady

B-Tree animation applet:

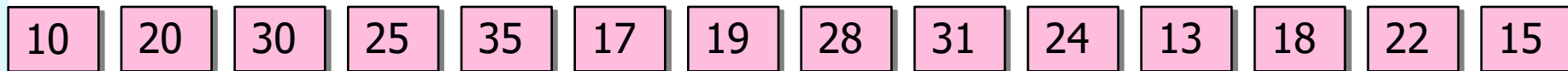
20

11 17 25 30

10 13 15 18 19 22 24 28 31 35

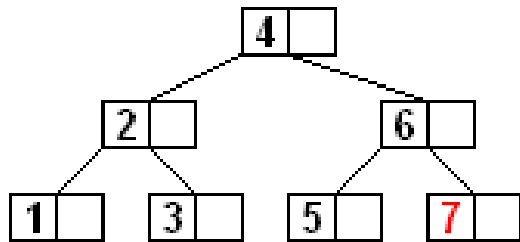
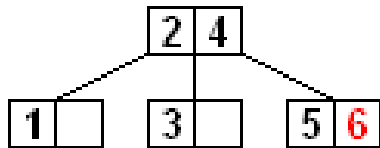
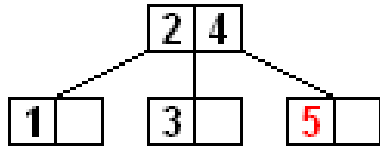
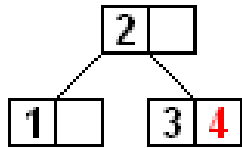
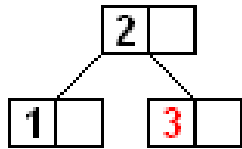
B-tree order: 1 ::> number 15 search insert delete

Done

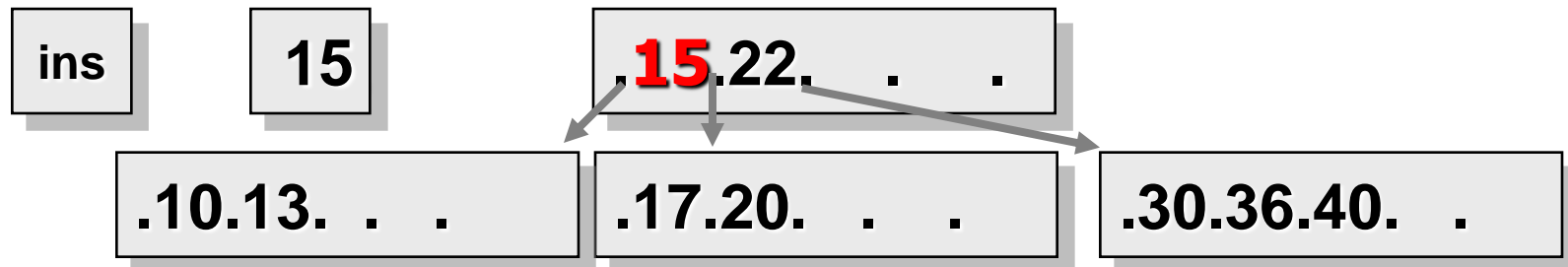
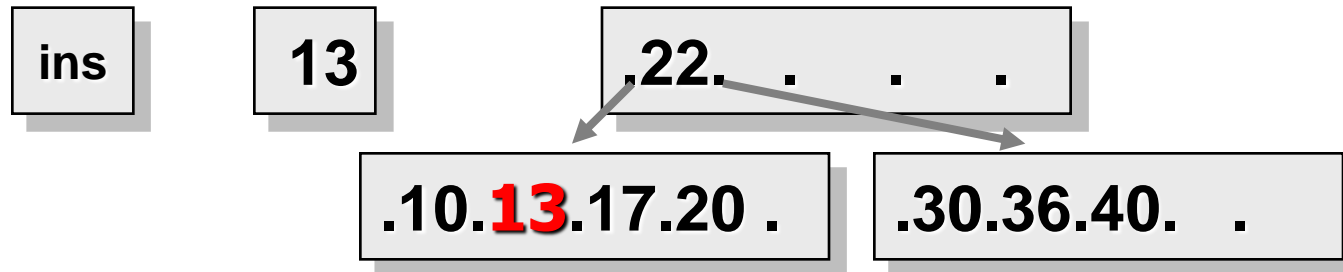
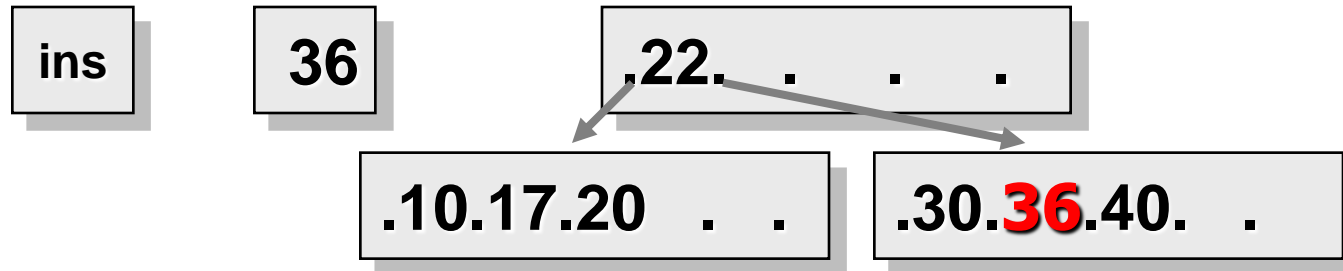
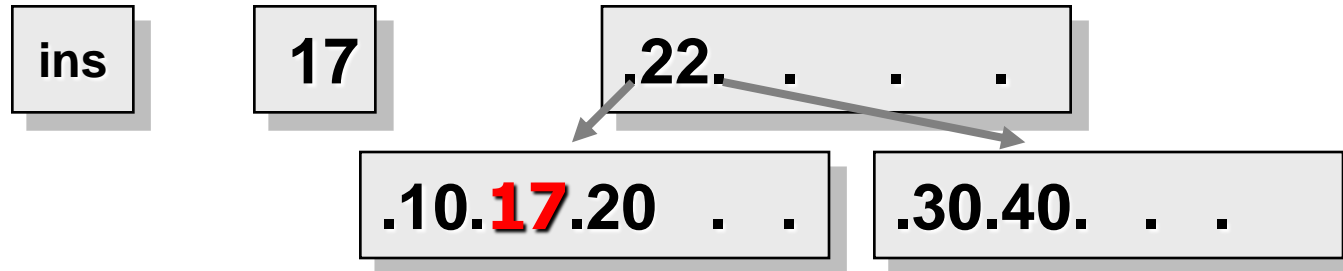


- zkoušejte nalézt čísla, která v indexu jsou i nejsou
- zkoušejte některá čísla z indexu zrušit
- a jiná nová přidat

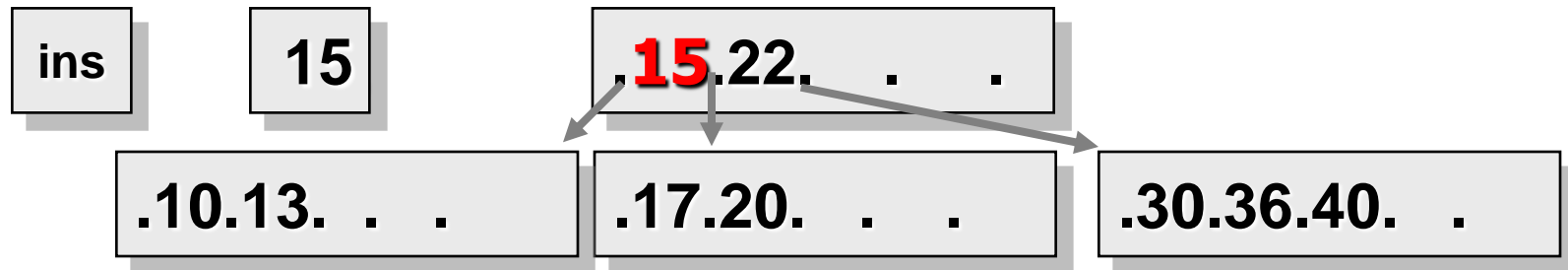
BTree



B stromy - animace

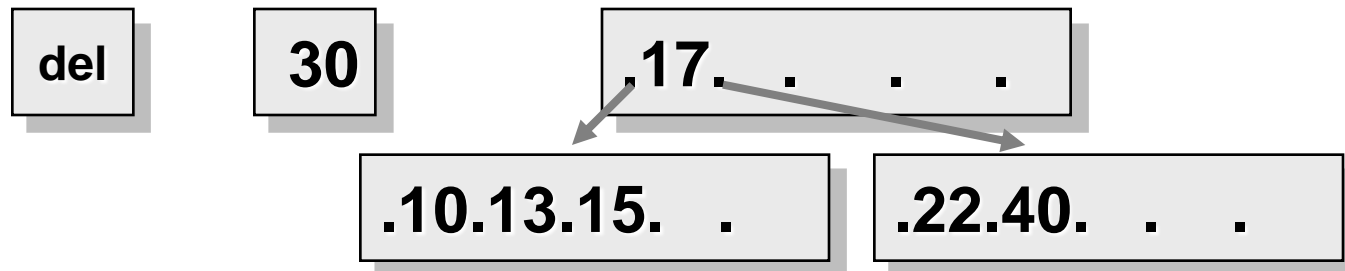
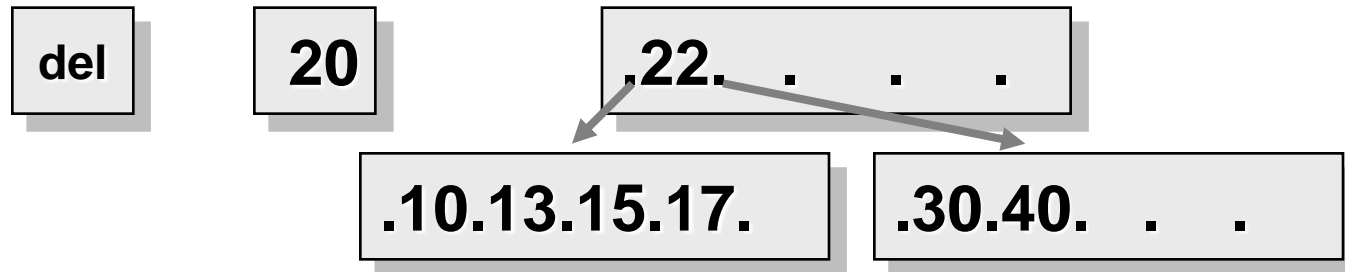
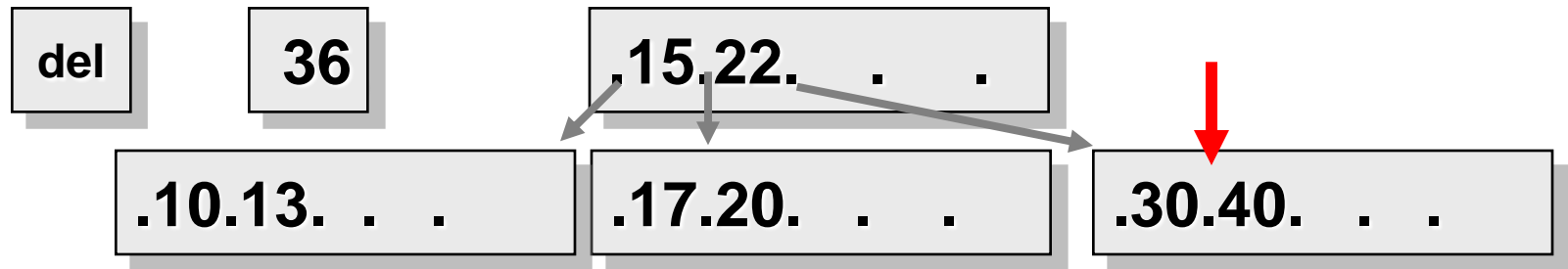
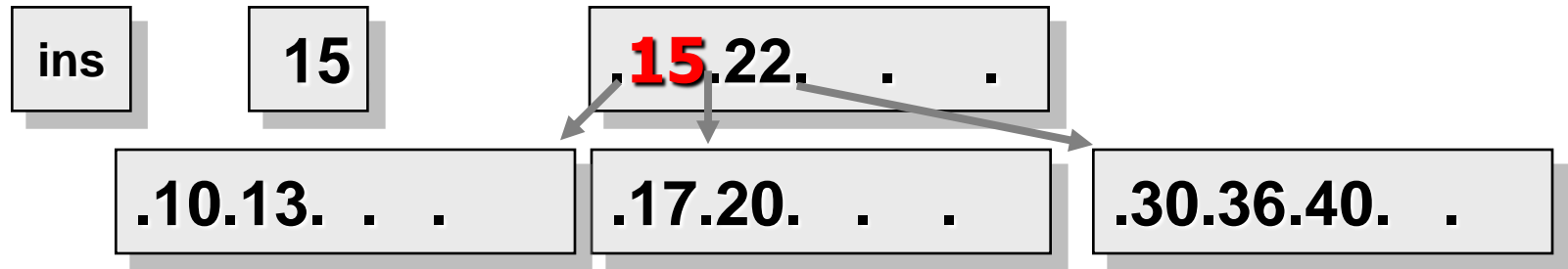


B stromy - animace

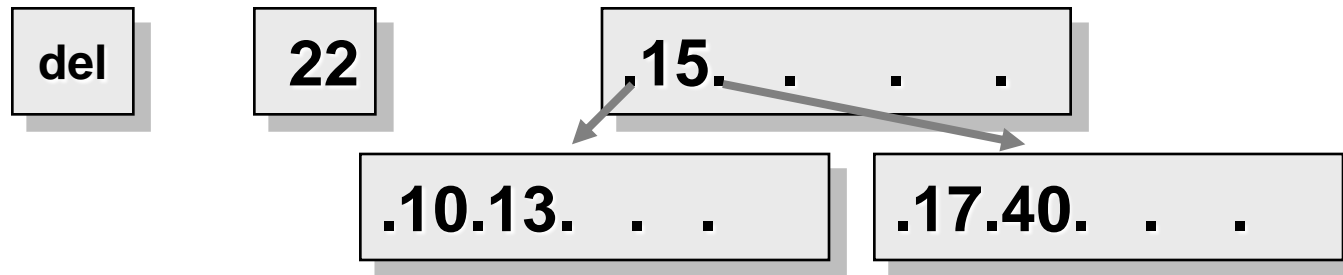
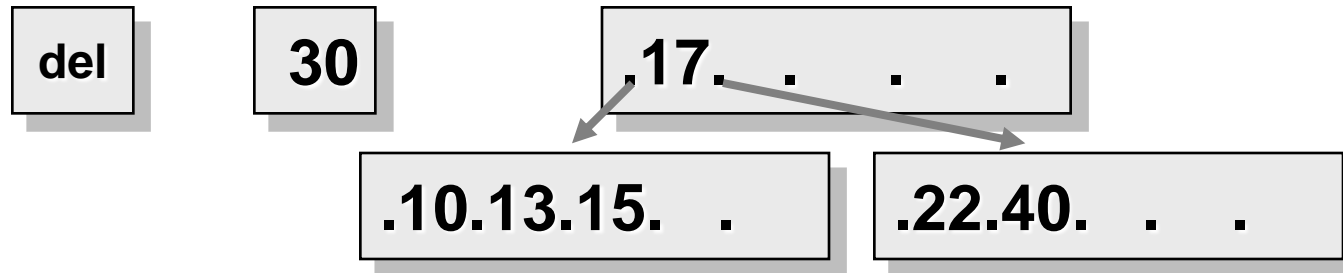


- src 13
- src 36
- src 20
- src 22
- src 18
- src 35
- src 1
- src 99

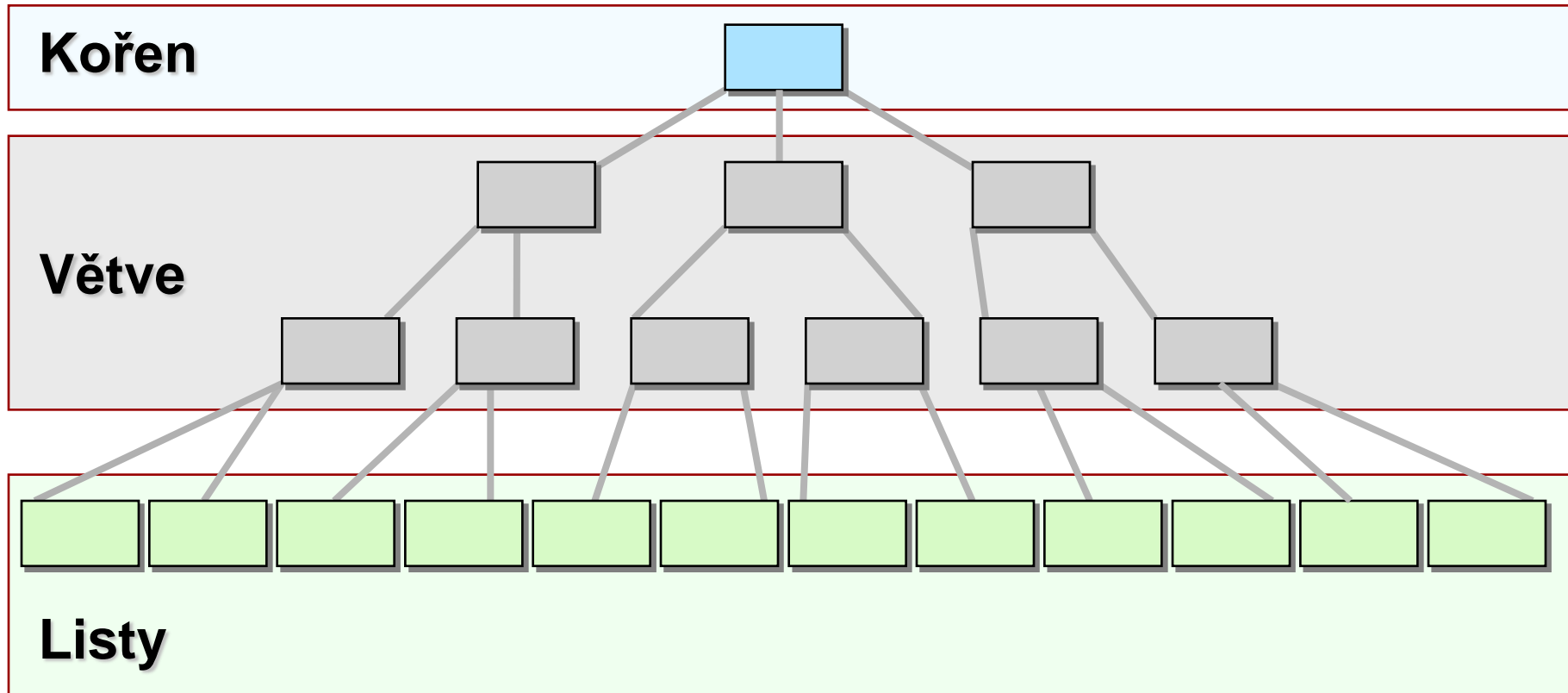
B stromy - animace



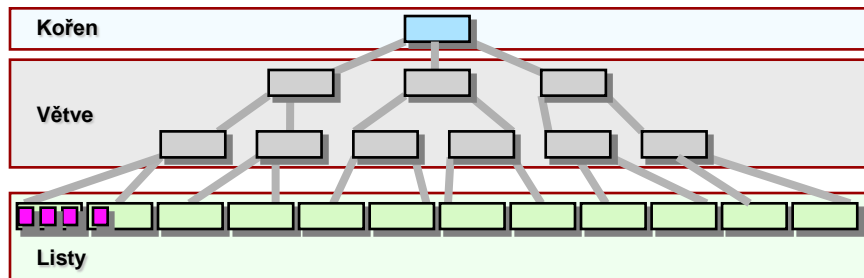
B stromy - animace



B stromy [B-tree]



B stromy [B-tree]

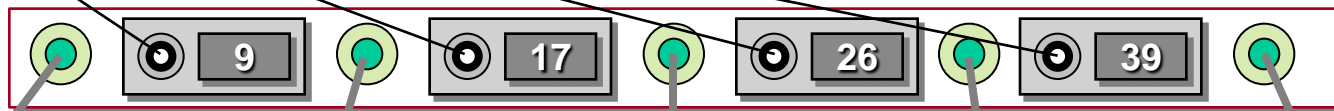
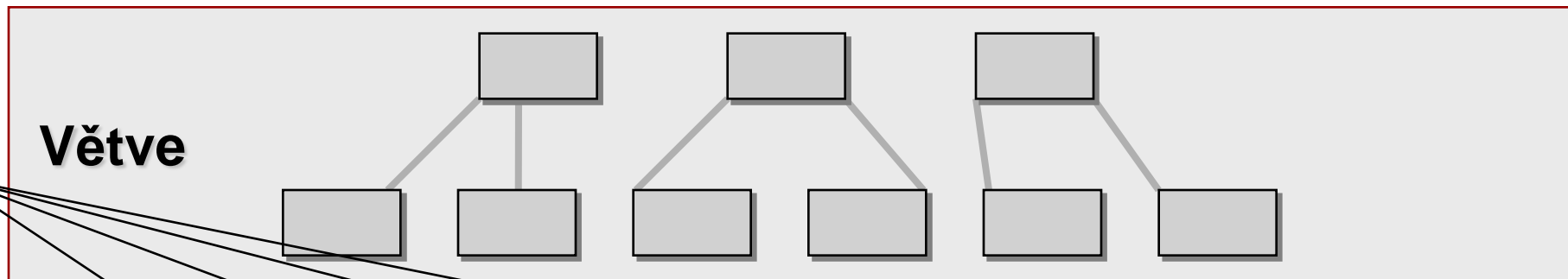


 ukazatel stromu

 **Joe**  **38** klíče Joe, 38, ...

 RowId indexované řádky

 Indexy v listech nejsou ve větvích



klíče < **9**

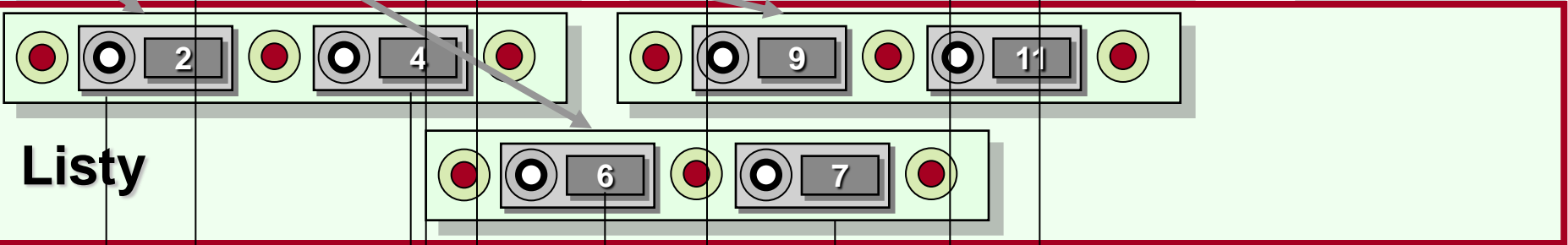
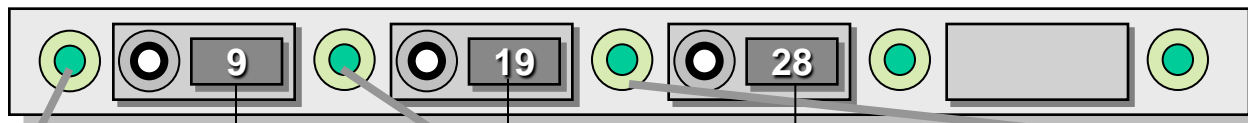
9 < klíče < **17**

17 < klíče < **26**

17 < klíče < **26**

klíče > **39**

B stromy [B-tree]



atd

Table	
6	..
8	..
9	..
7	..
2	..
16	..
4	..
5	..
28	..
19	..
..	..

B stromy [B-tree]

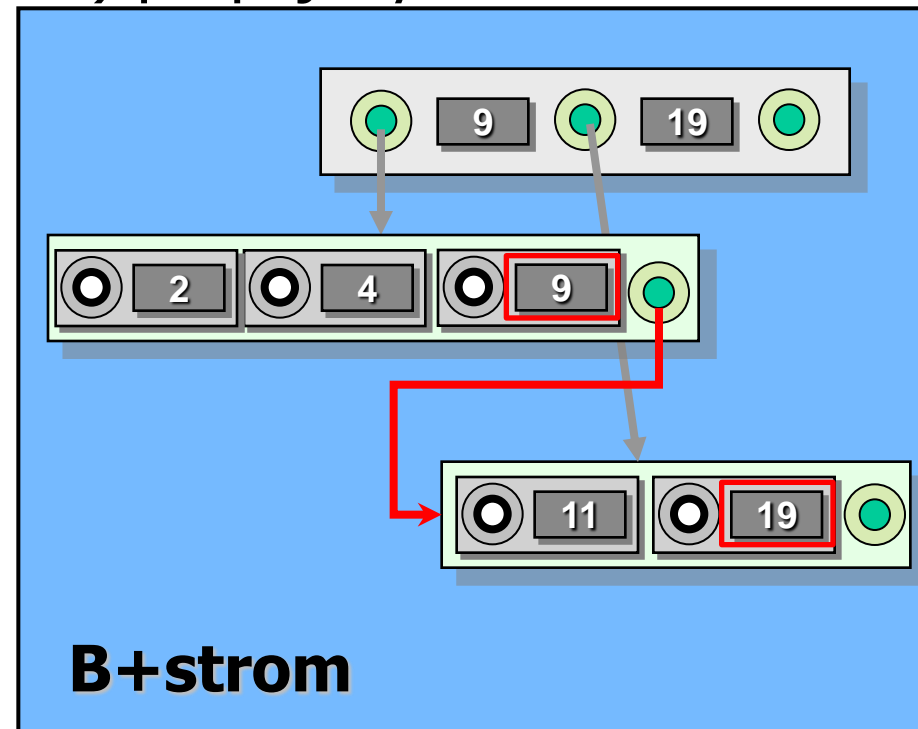
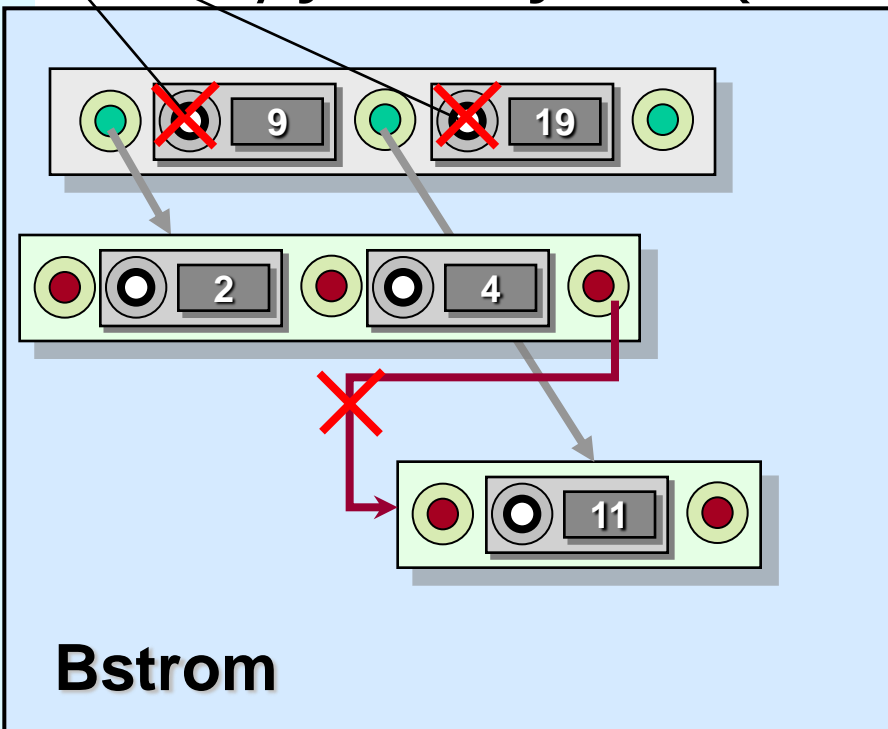
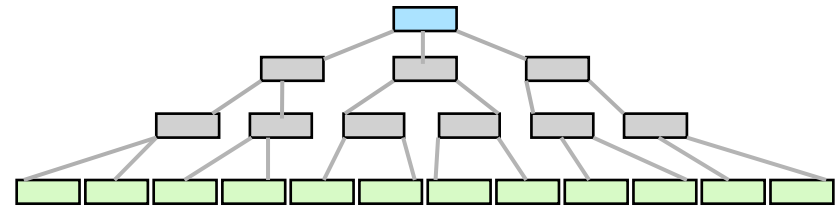
- Rudolf Bayer a Ed McCreight, 1972
- nikdy nevysvětlili, co značí **B**
- **B**alanced (vyvážený) ?
- **B**ayerův ?
- **B**oening (oba pracovali u této firmy)
- přidáním či výmazem klíče se mění počet klíčů v uzlu
- pro dodržení pravidel se v případě potřeby
- uzly rozdělují či spojují
- přidáváním klíčů se hloubka stromu zvětšuje
- rušením klíčů se hloubka stromu zmenšuje
- strom je stále vyvážený

B+ strom

- vychází z B-stromu

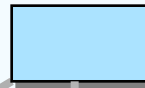
Rozdíly

- klíče ve větvích nemají data (rowid)
- všechny klíče jsou v listech i s daty (rowid)
- listy jsou vzájemně (sekvenčně) propojony

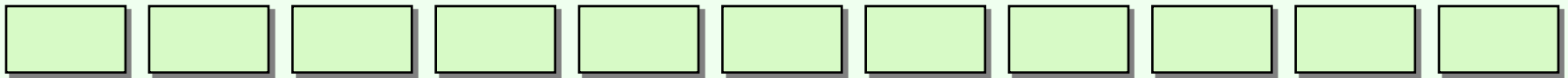
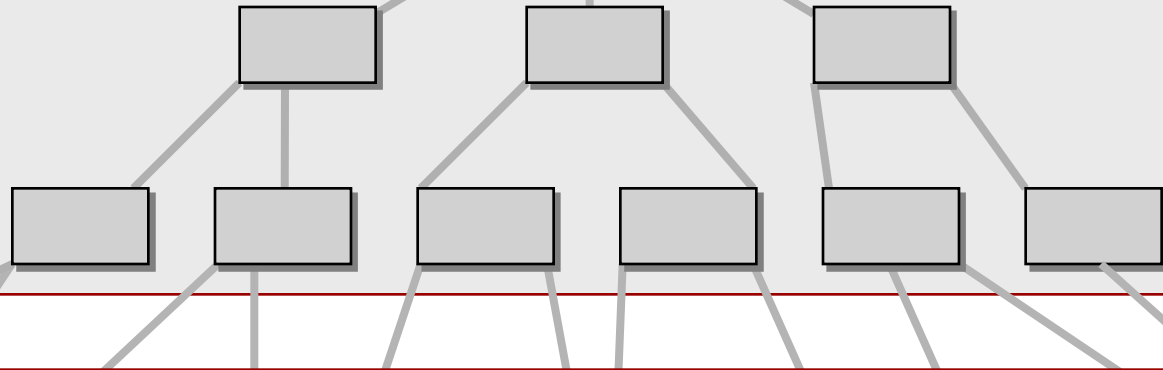


Bstrom a B+strom

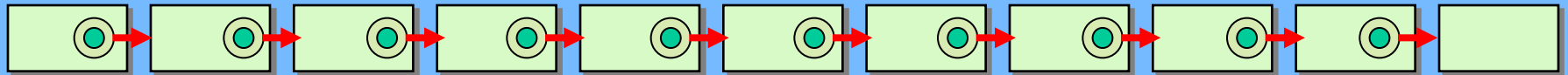
Kořen



Větve

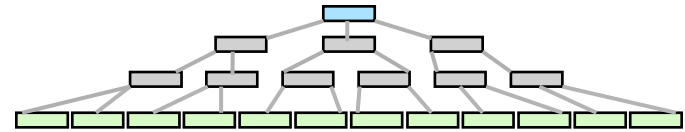


Bstrom Listy



B+strom Listy

Porovnání B-strom vs. B+Strom



- žádné duplicity indexů
- snadnější údržba indexů

přístupy

- přímý dle indexu
- rychlý pro indexy větví

- NE
- NE

Bstrom

- každý index ve větvích
- má duplicitu v listech
- horší údržba indexů

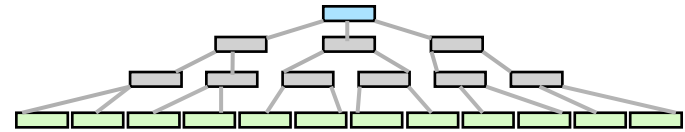
přístupy

- přímý dle indexu
- pomalejší pro indexy větví
- [pro RowId musí do listů]
- **sekvenční indexově**
- **sekvenční - interval (Od,Do)**

index - range

B+strom

Porovnání B-strom vs. B+Strom - **klastrovaný**



- žádné duplicity indexů
- snadnější údržba indexů

přístupy

- přímý dle indexu
- rychlý pro indexy větví

■ **NE**

■ **NE**

Bstrom

- každý index ve větvích
- má duplicitu v listech
- horší údržba indexů

přístupy

- přímý dle indexu
- pomalejší pro indexy větví
- [pro RowId musí do listů]

■ Full Scan Table [sekvenční]

■ Range Scan Table [sekvenční]

index - range

B+strom



a zase něco jiného