



# Algoritmizace

M. Genyg-Berezovskyj, D. Průša

2010 – 2021

# Dnešní témata

- AVL strom (samovyvažující se BVS)
- B-strom (více klíčů v uzlech)



slido



**Join at [slido.com](https://slido.com)  
#543701**

① Start presenting to display the joining instructions on this slide.

slido



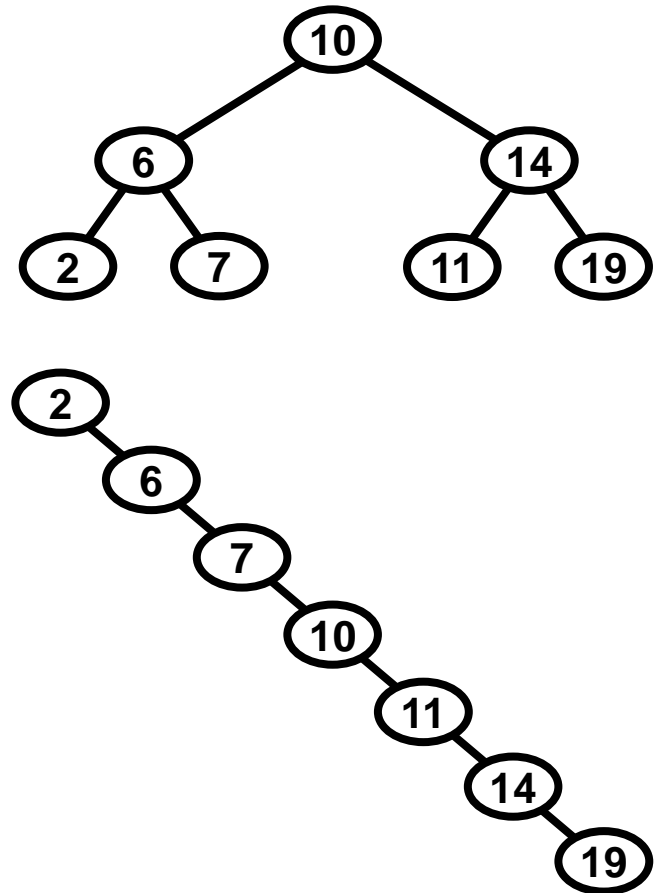
# Audience Q&A Session

① Start presenting to display the audience questions on this slide.

Prosím, mohl byste stručně srovnat výhody a nevýhody vyhledávání BVS a v poli? Kdy konkrétně je vhodnější zvolit jaký typ?

# Z minula

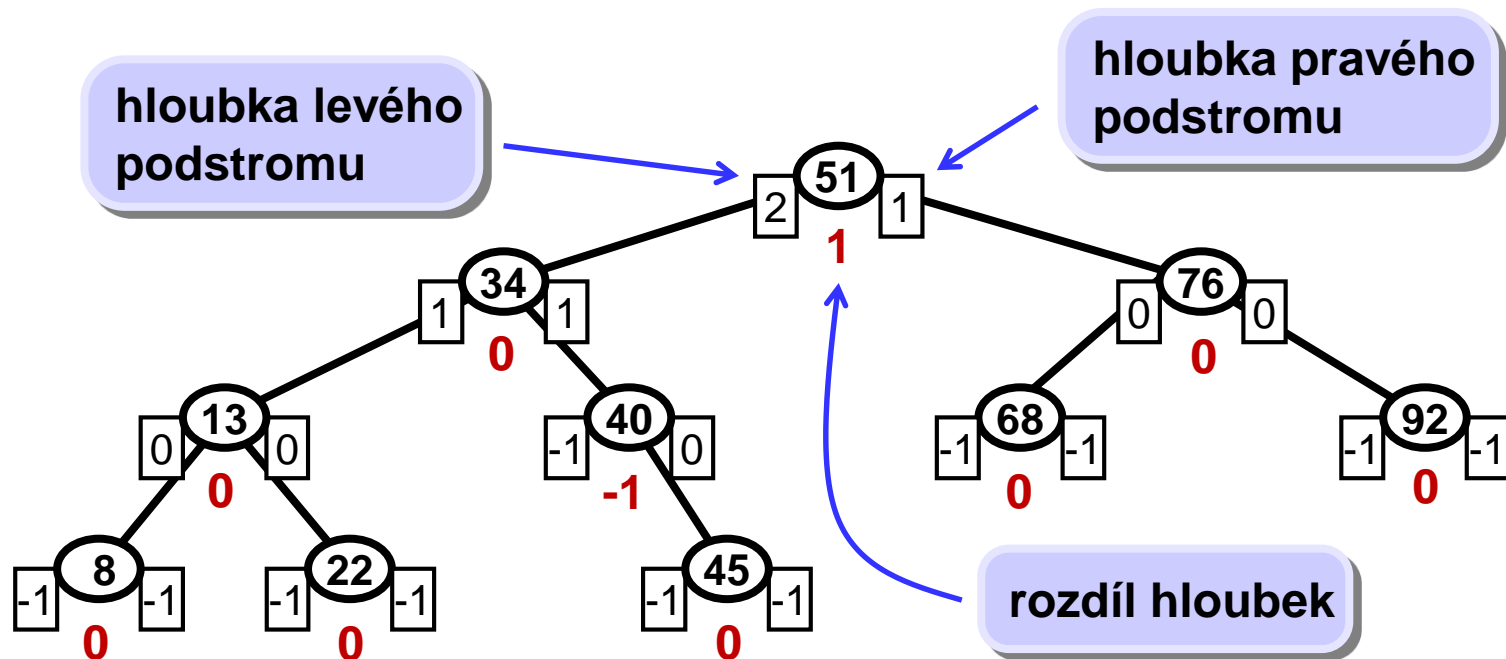
Asymptotická složitost operací v BVS je úměrná maximální hloubce uzlů, které navštívíme.



| Operace | BVS s $n$ uzly |                  |
|---------|----------------|------------------|
|         | Obecný         | Vyvážený         |
| Find    | $\Theta(n)$    | $\Theta(\log n)$ |
| Insert  | $\Theta(n)$    | $\Theta(\log n)$ |
| Delete  | $\Theta(n)$    | $\Theta(\log n)$ |

# AVL strom

- Autoři: **Adelson-Velsky & Landis** (1962)
- Адельсон-Вельский & Ландис, AVL-дерево
- BVS splňující: pro každý uzel je rozdíl hloubek jeho levého a pravého podstromu roven  $-1$ ,  $0$  nebo  $1$ .

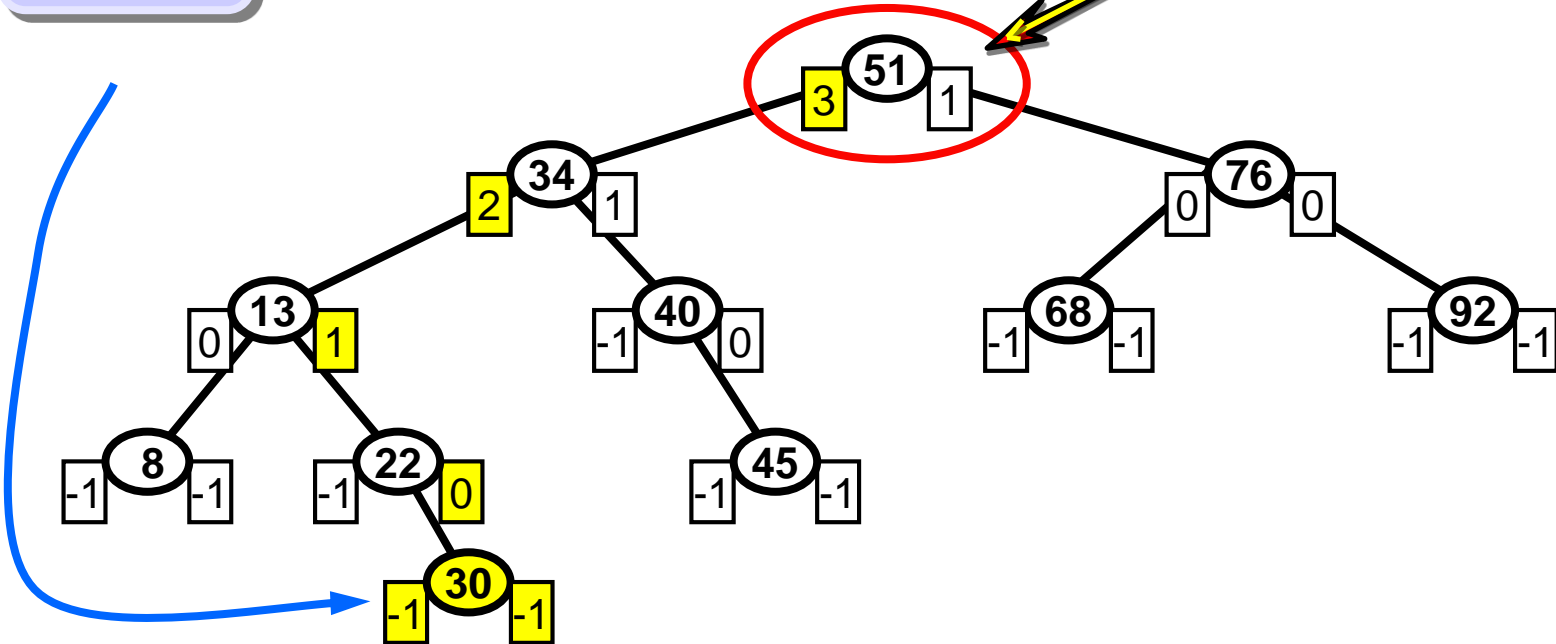


- Hloubka AVL stromu s  $n$  uzly není větší než  $1.45 \log_2(n + 2)$ .

# Operace Insert v AVL

V každém uzlu má být rozdíl hloubek obou podstromů roven -1, 0, 1 !!

Insert 30

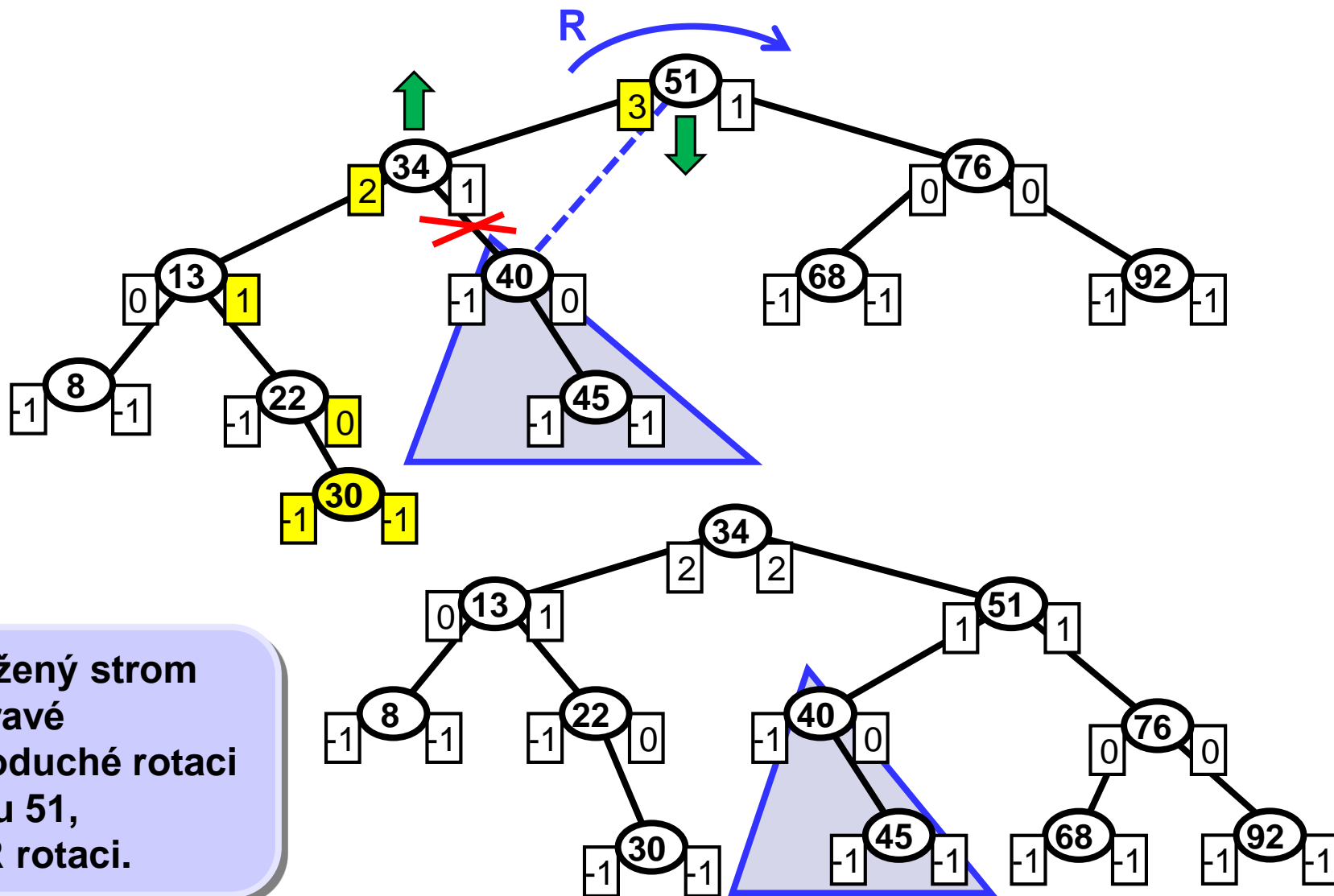


Změněné hloubky

Levý podstrom uzlu 51 je příliš hluboký, strom přestal být AVL stromem.



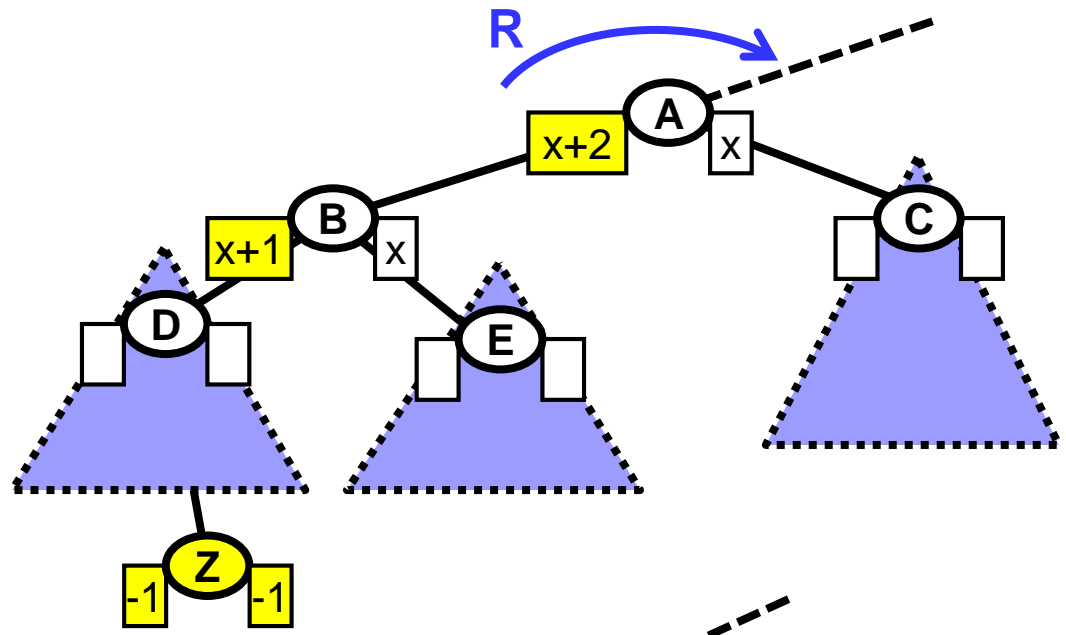
# Náprava rozvážení pomocí rotace



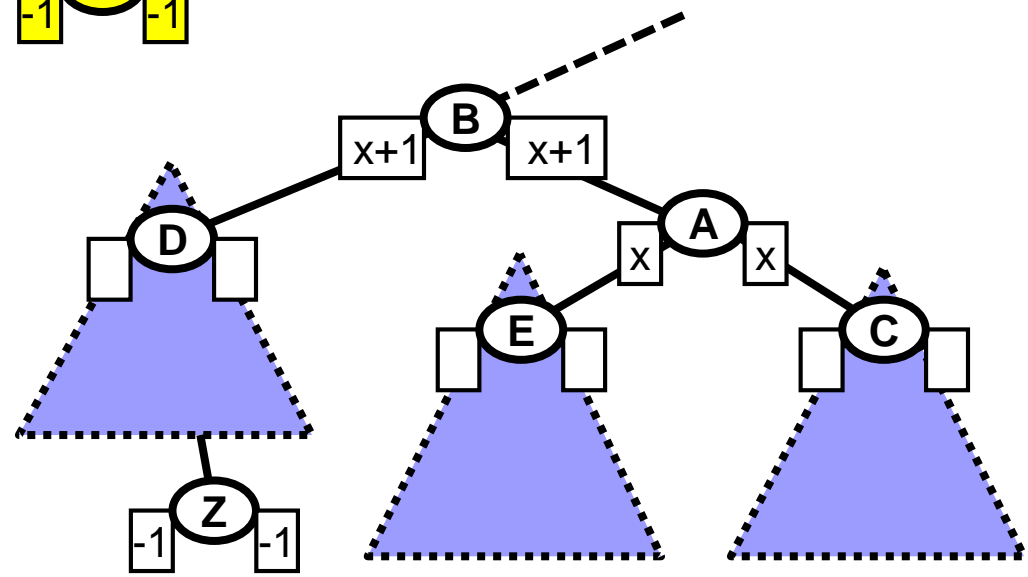
Vyvážený strom  
po pravé  
jednoduché rotaci  
v uzlu 51,  
tzv. R rotaci.

# Rotace R obecně

Před

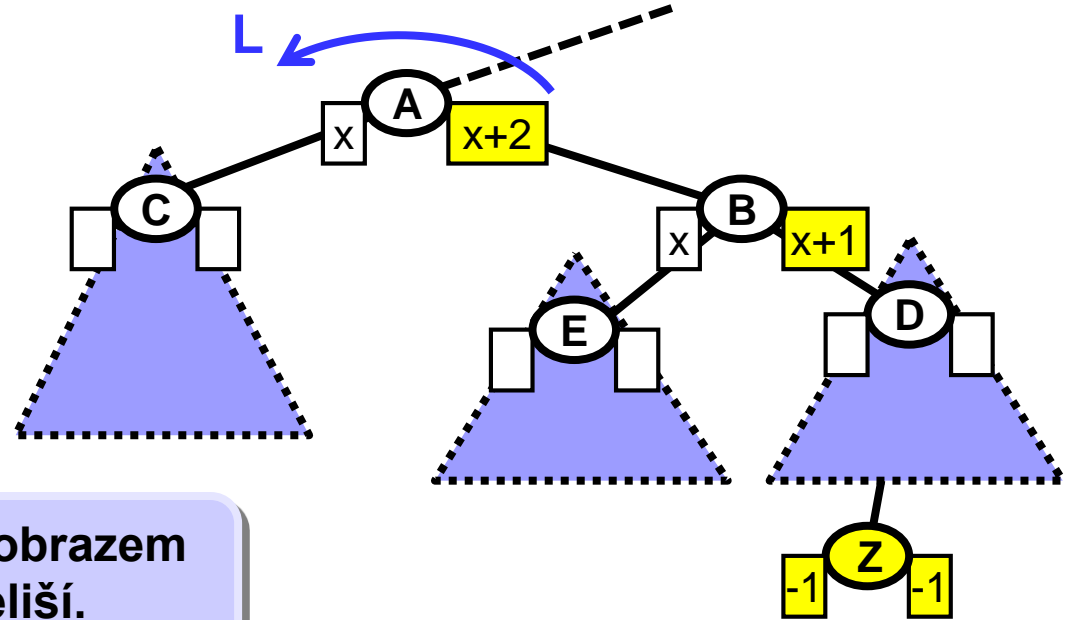


Po



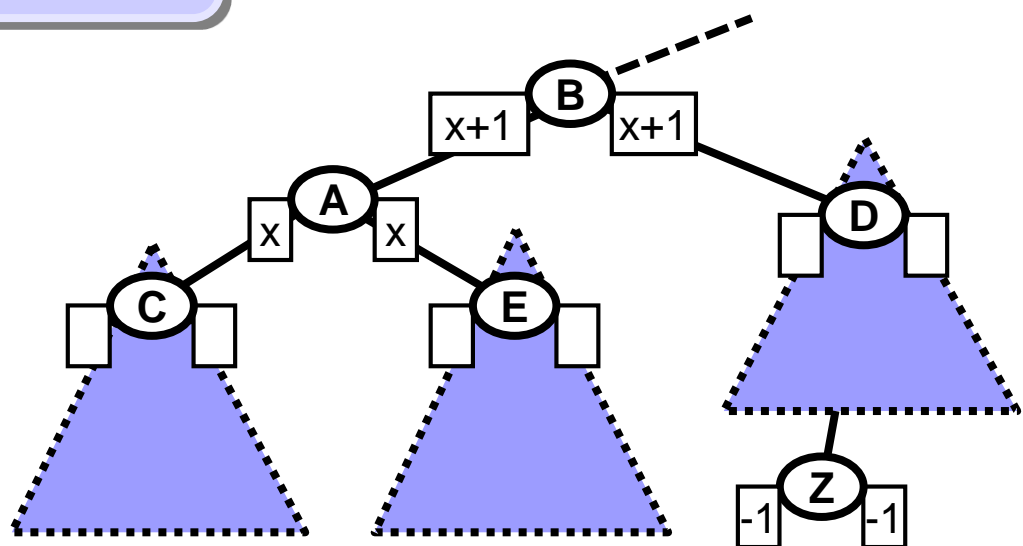
# Rotace L obecně

Před



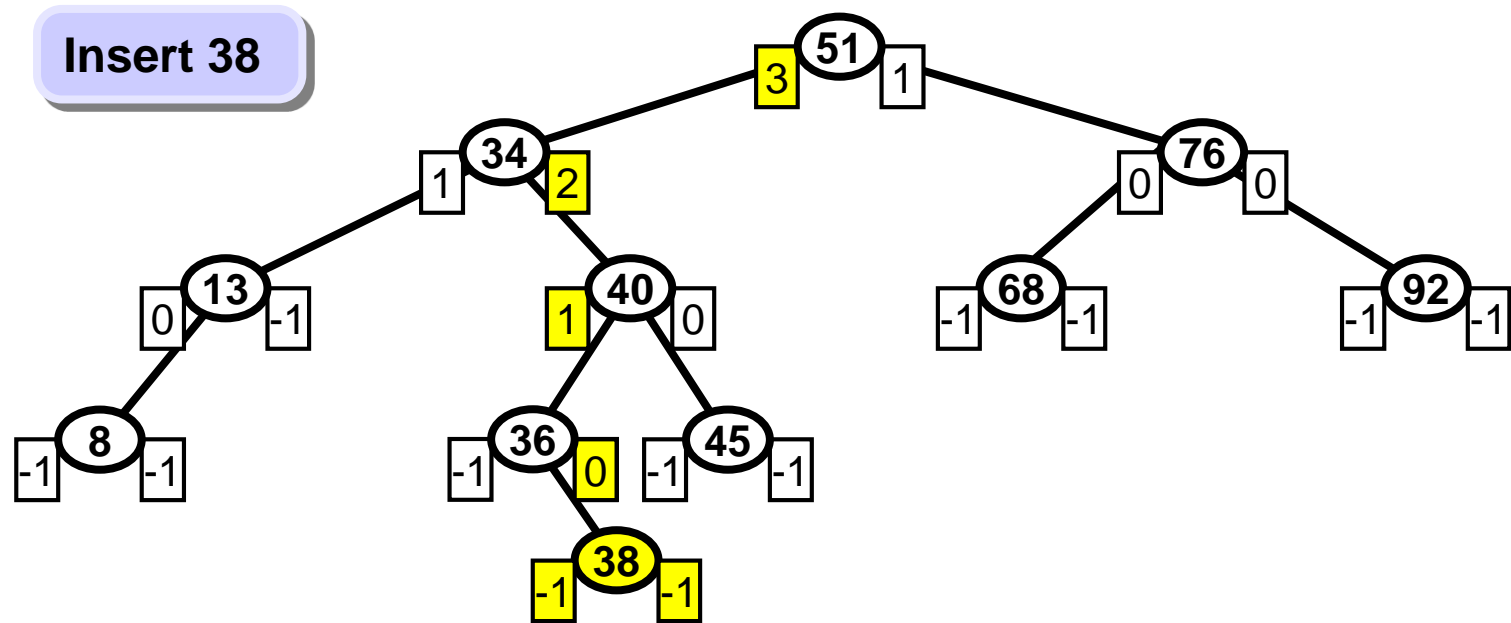
Rotace L je symetrickým obrazem rotace R, jinak se od ní neliší.

Po



# Otázka

Jak můžeme vyvážit zobrazený strom pomocí jedné rotace?



- A. R rotací v kořeni.
- B. L rotací v uzlu s klíčem 34.
- C. Nelze vyvážit pomocí jedné L nebo R rotace aplikované na libovolný uzel.

slido

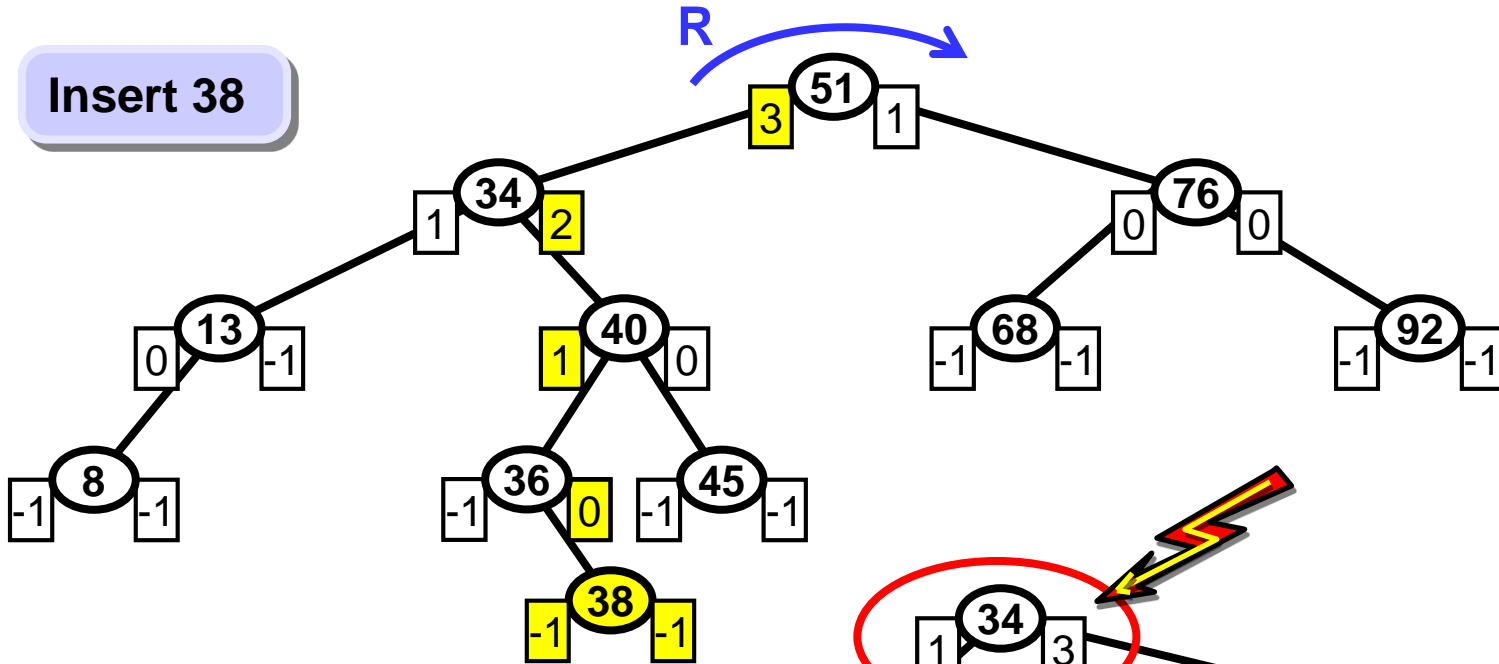


**Jak můžeme vyvážit  
zobrazený strom pomocí  
jedné rotace?**

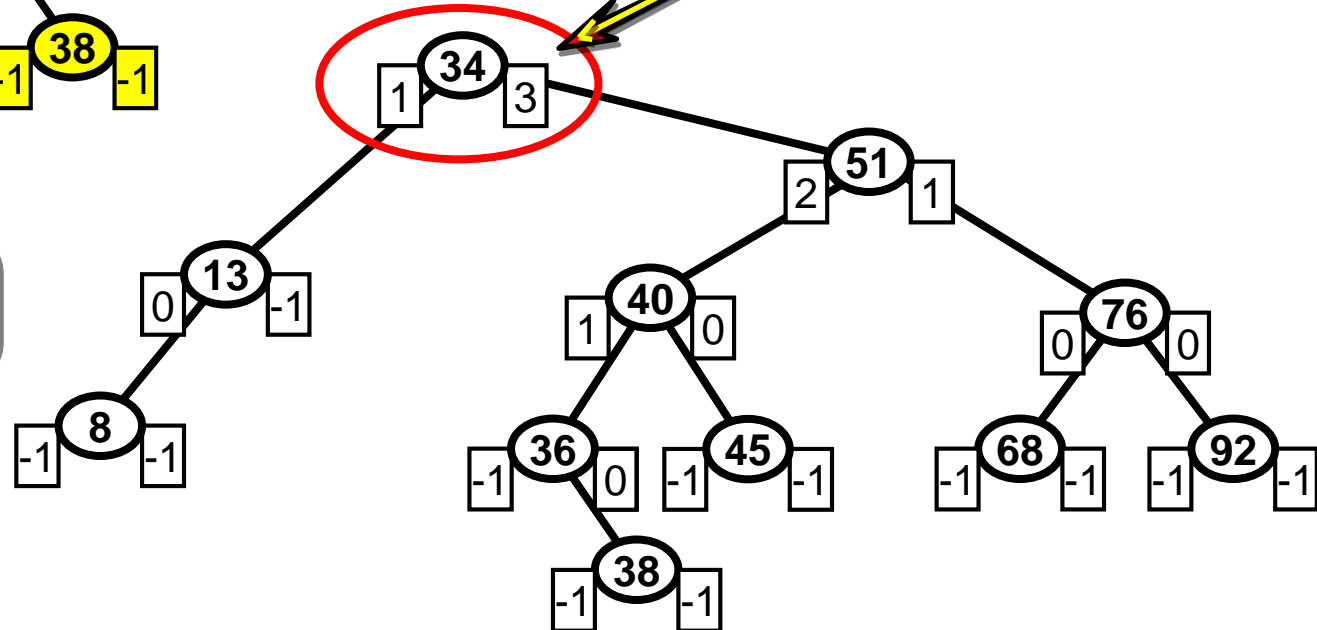
ⓘ Start presenting to display the poll results on this slide.

# R nebo L rotace nemusí stačit

Insert 38

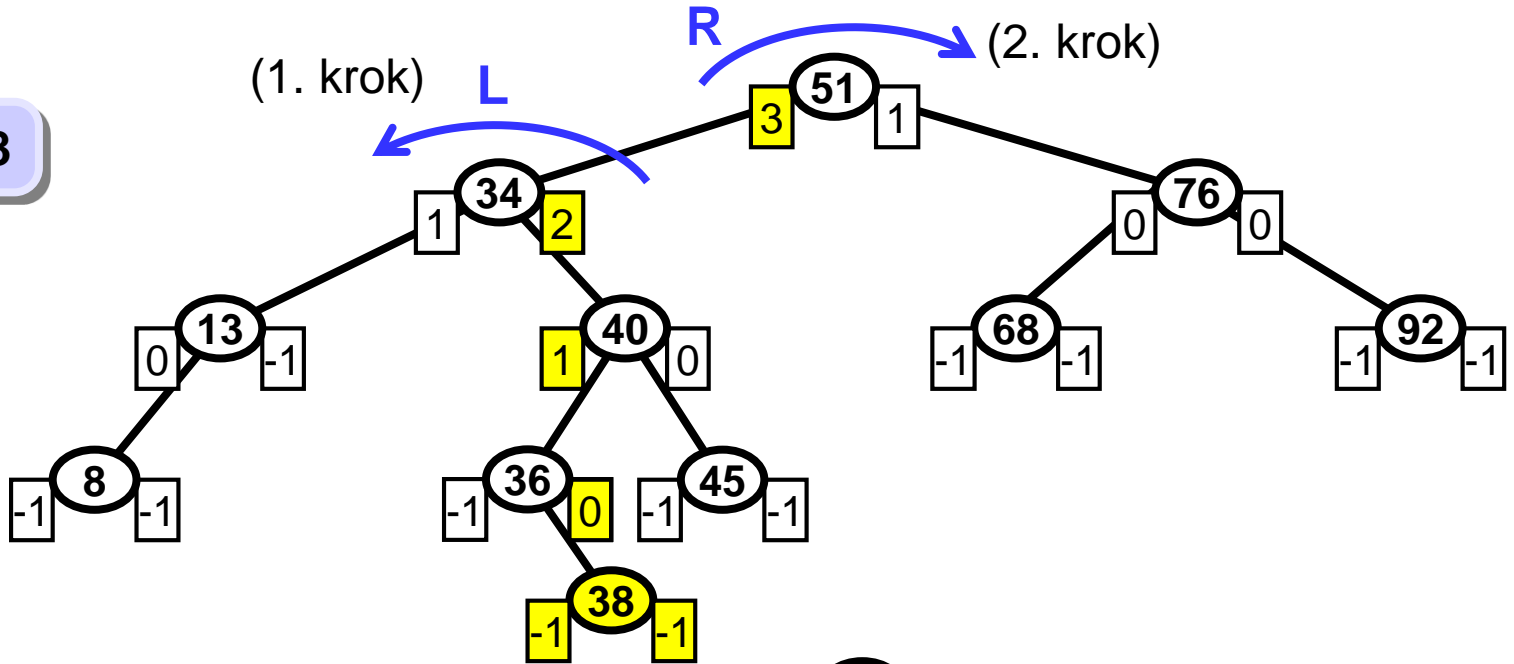


Výsledek po R rotaci

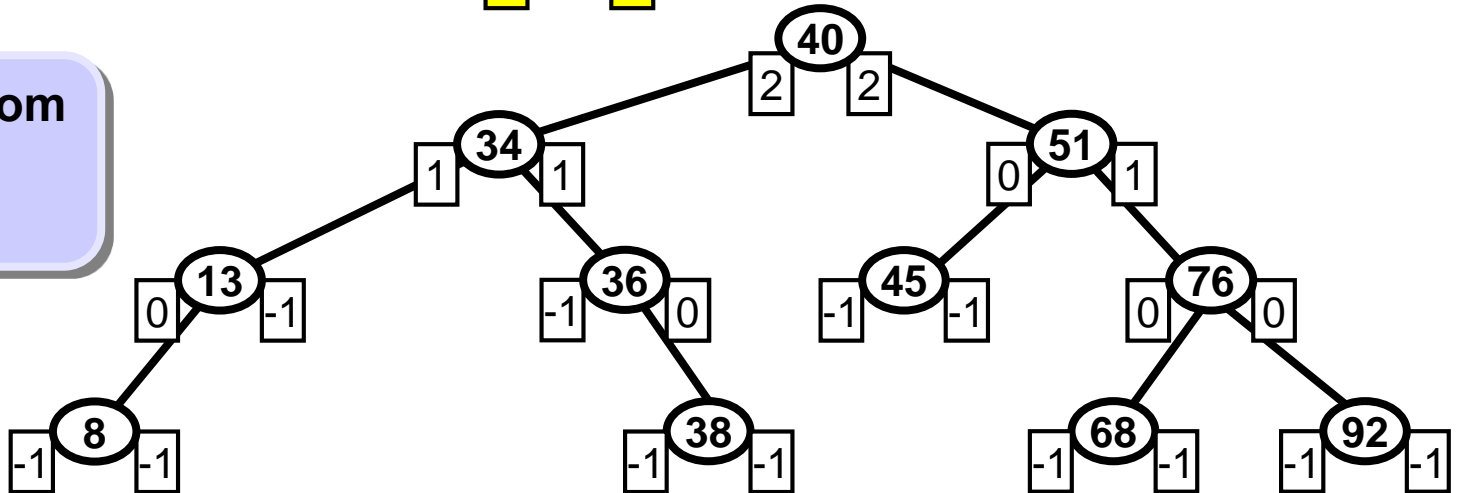


# Dvojitá LR rotace

Insert 38

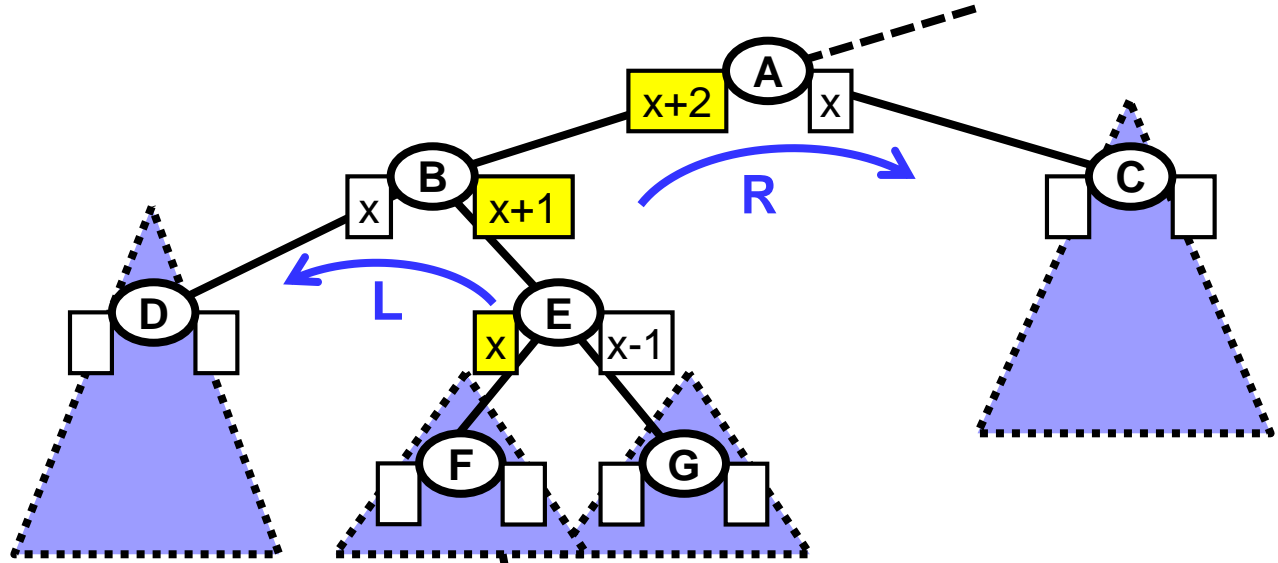


Vyvážený strom  
po dvojitě  
LR rotaci

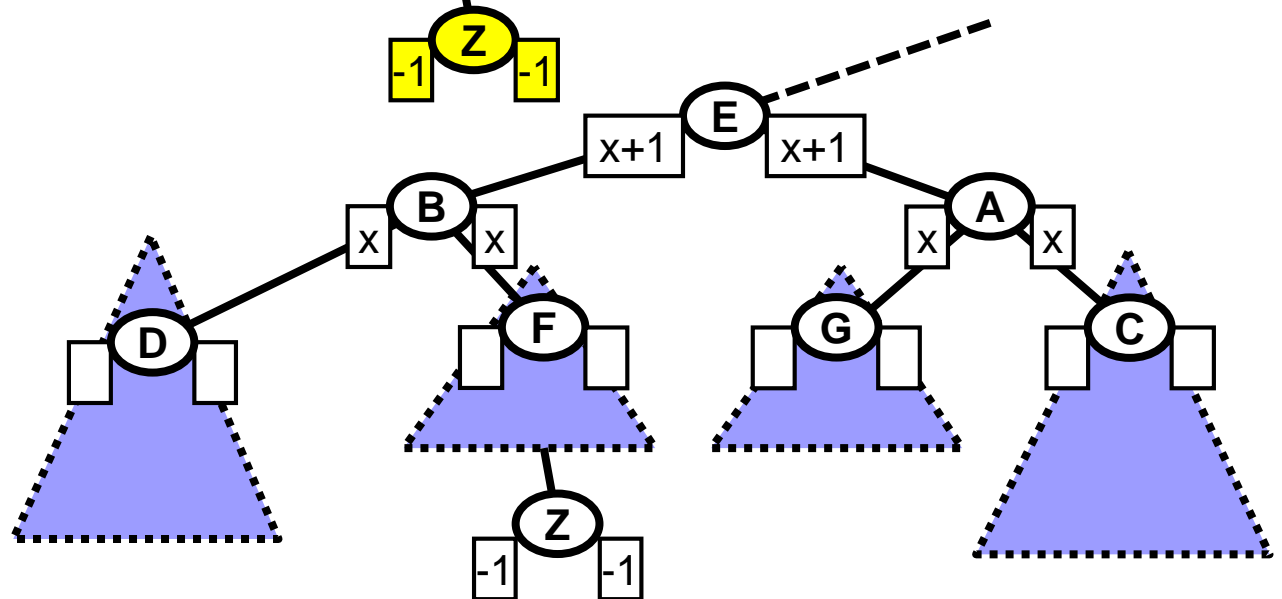


# Rotace LR obecně

Před



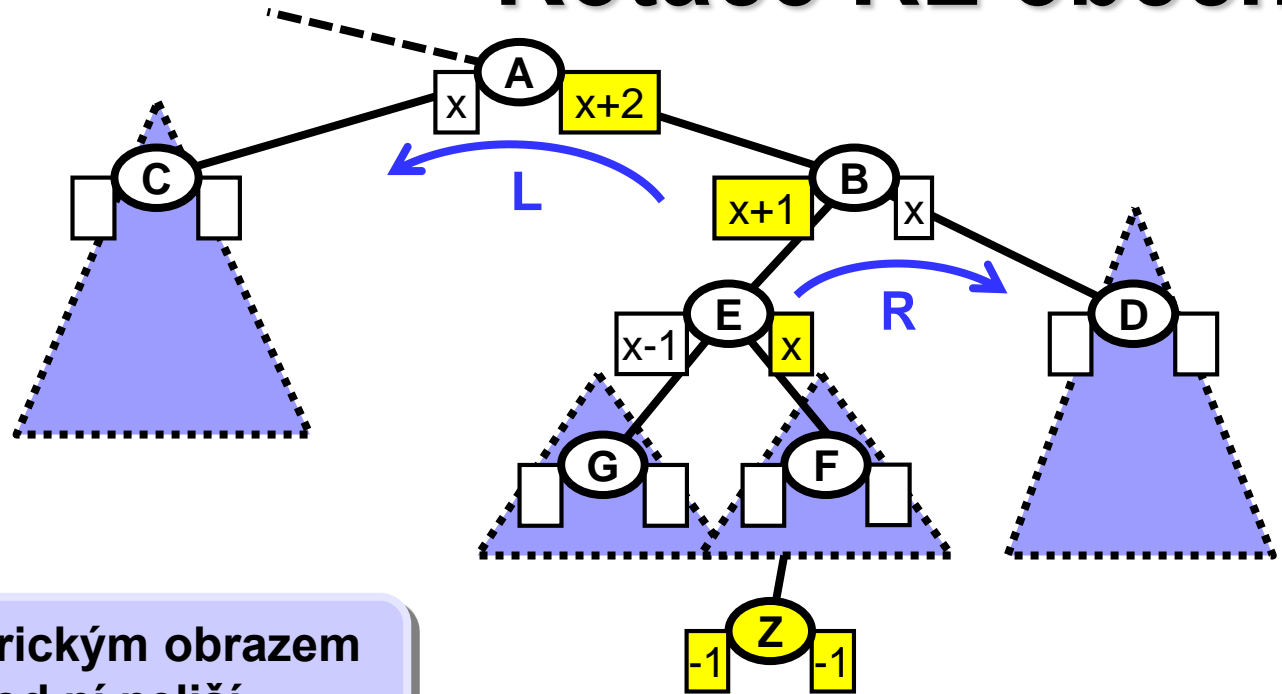
Po





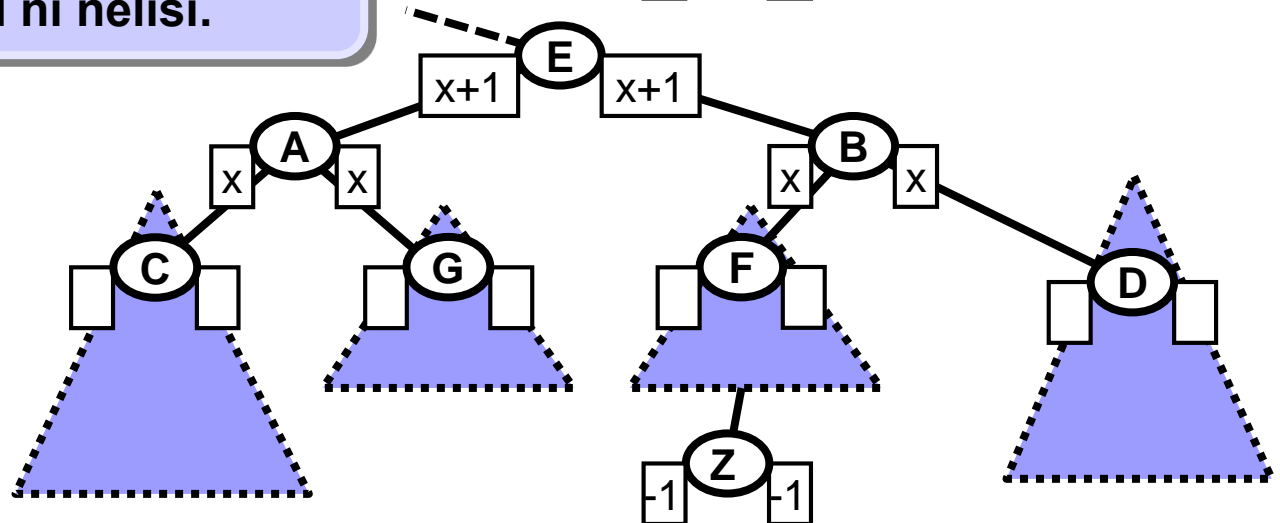
# Rotace RL obecně

Před



Rotace RL je symetrickým obrazem rotace LR, jinak se od ní neliší.

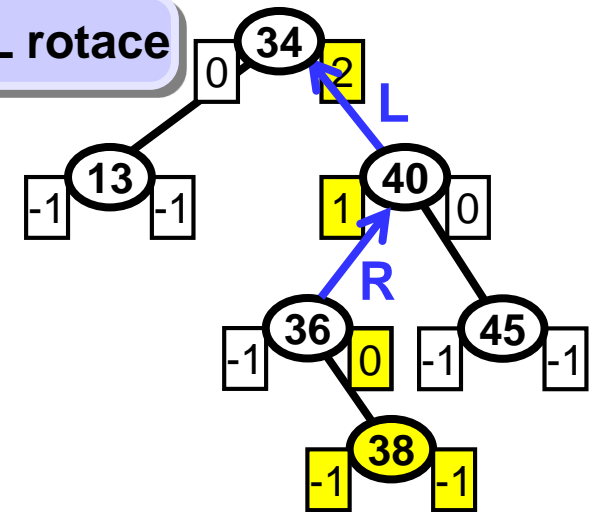
Po



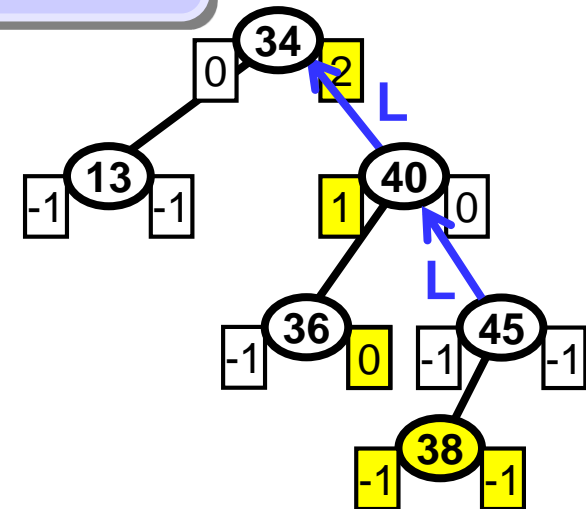
# Pravidla pro určení rotace

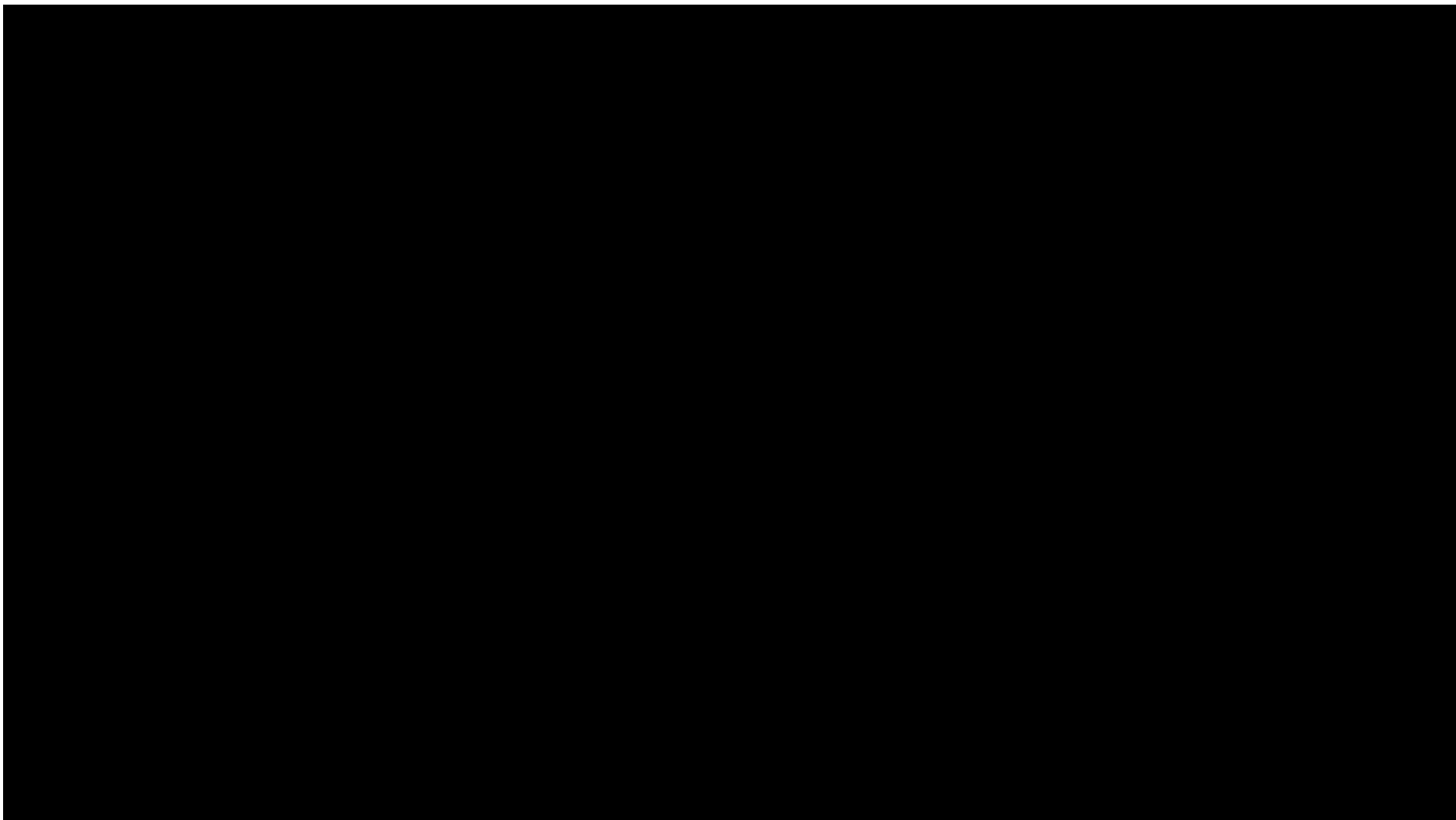
- Od přidaného uzlu postupujeme směrem ke kořeni a aktualizujeme hloubky podstromů v každém navštíveném uzlu.
- Když narazíme na rozvážený uzel, do kterého jsme bezprostředně došli:
  - dvěma hranami doprava nahoru, provedeme v tomto uzlu **R** rotaci.
  - dvěma hranami doleva nahoru, provedeme v tomto uzlu **L** rotaci.
  - hranami doleva a pak doprava nahoru, provedeme v tomto uzlu **LR** rotaci.
  - hranami doprava a pak doleva nahoru, provedeme v tomto uzlu **RL** rotaci.
- Po provedení jedné rotace po operaci Insert je AVL strom opět vyvážený.

RL rotace



L rotace

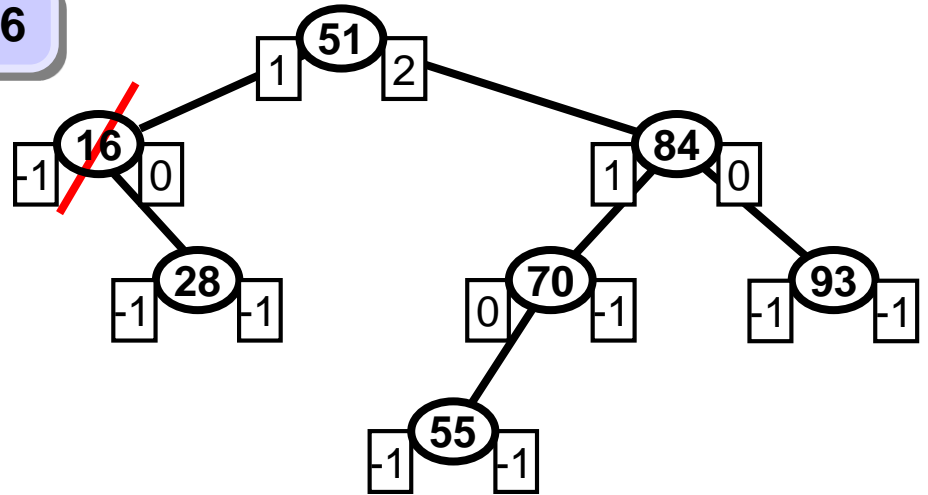




[https://youtu.be/LS\\_U6g-Fsts](https://youtu.be/LS_U6g-Fsts)

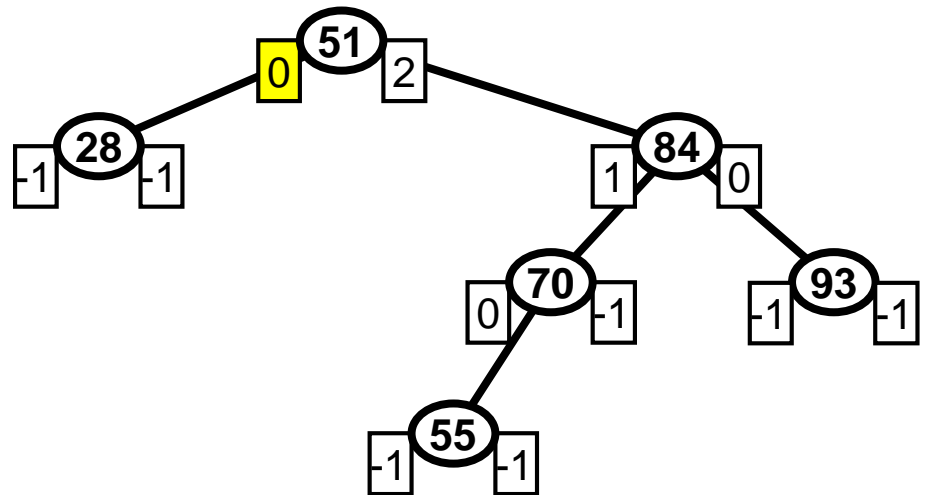
# Operace Delete v AVL

Delete 16



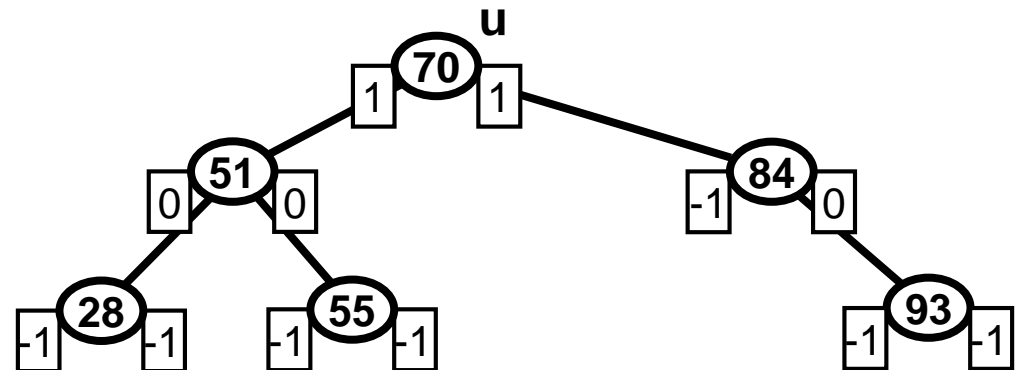
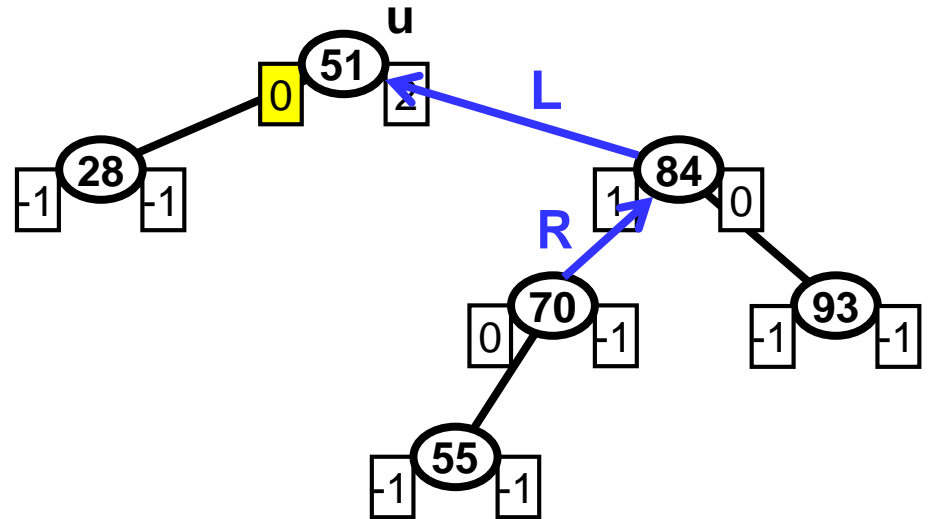
- Delete proběhne standardně jako v obyčejném BVS.

- Poté postupujeme od místa smazání nahoru ke kořeni a aktualizujeme výšky podstromů v každém uzlu.
- Při rozvážení aplikujeme rotaci podobně jako při vkládání.



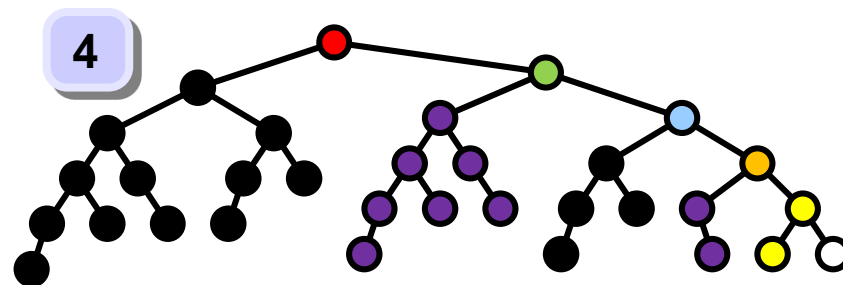
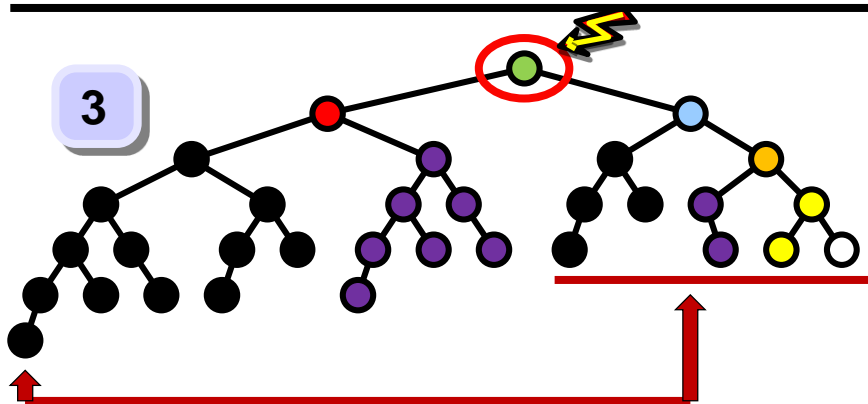
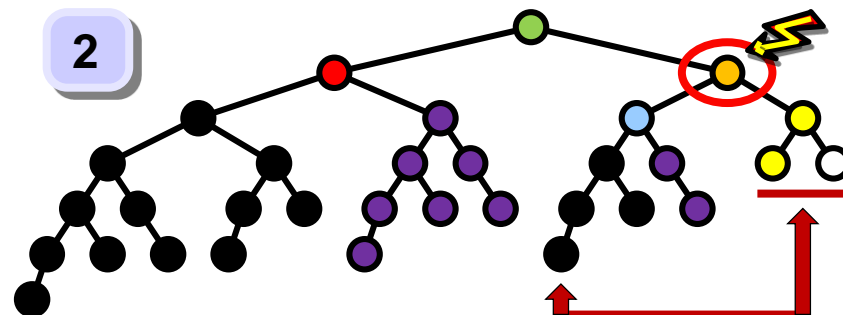
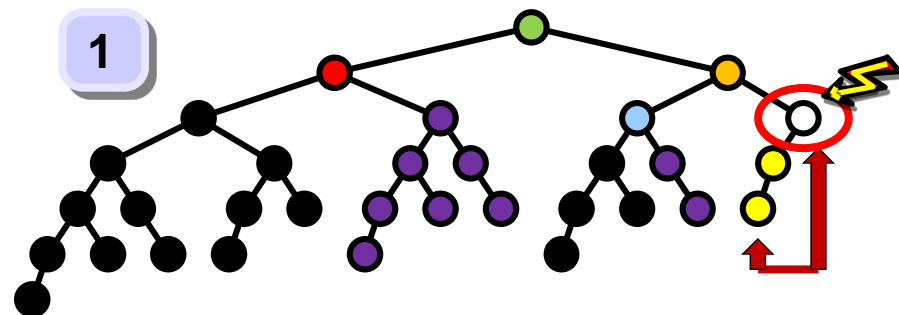
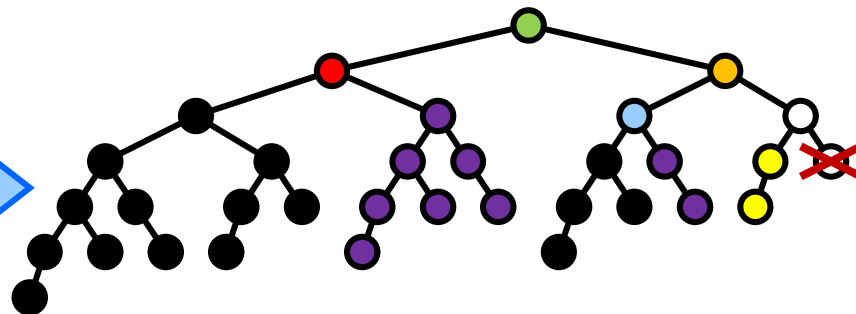
# Operace Delete v AVL

- V rozváženém uzlu  $u$  se podíváme na první dvě hrany cesty z  $u$  do nejhlubšího uzlu (při více takovýchto uzlech máme na výběr).
- Směr těchto dvou hran určí typ rotace podobně jako u operace Insert.



# Opakové rotace po operaci Delete

Příklad: Delete v AVL stromu s  $n = 33$  uzly.



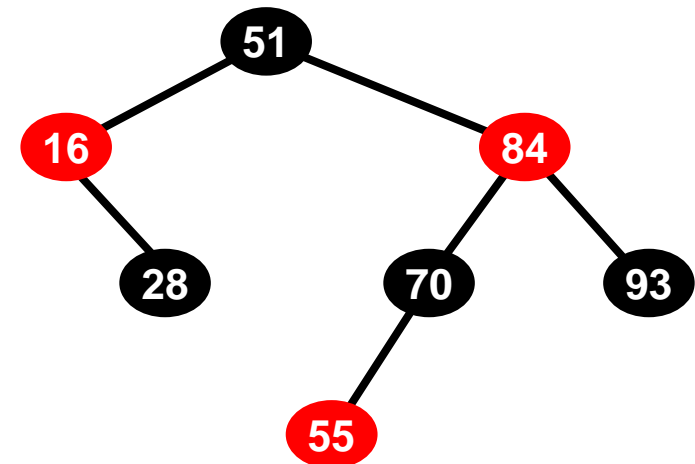
Vyvážení obecně po  $O(\log n)$  rotacích.

# Shrnutí

| Operace | BVS s $n$ uzly | AVL s $n$ uzly   |
|---------|----------------|------------------|
| Find    | $\Theta(n)$    | $\Theta(\log n)$ |
| Insert  | $\Theta(n)$    | $\Theta(\log n)$ |
| Delete  | $\Theta(n)$    | $\Theta(\log n)$ |

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

- Existuje i jiný druh samovyvažovacího BVS, tzv. červeno-černý strom (implementace: `java.util.TreeSet`).
- AVL je výhodnější, pokud operace Find jsou čtenější než operace Insert a Delete.





# Audience Q&A Session

① Start presenting to display the audience questions on this slide.



# B-strom

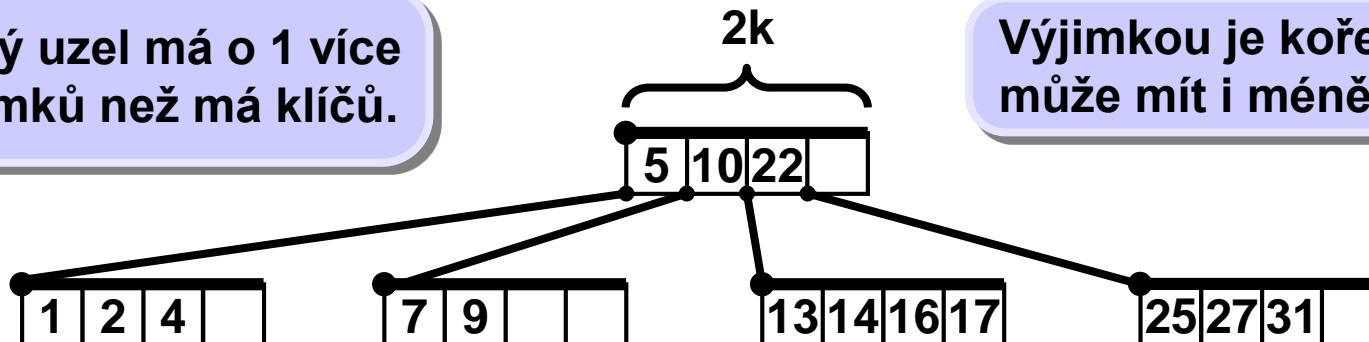
- Autoři: Bayer & McCreight (1972)
- Původ názvu: **b**alanced, **B**ayer, **B**oeing
- Využití: databázové systémy, souborové systémy

B-strom řádu  $k$

Každý uzel obsahuje minimálně  $k$  a maximálně  $2k$  vzestupně uspořádaných klíčů.

Každý uzel má o 1 více potomků než má klíčů.

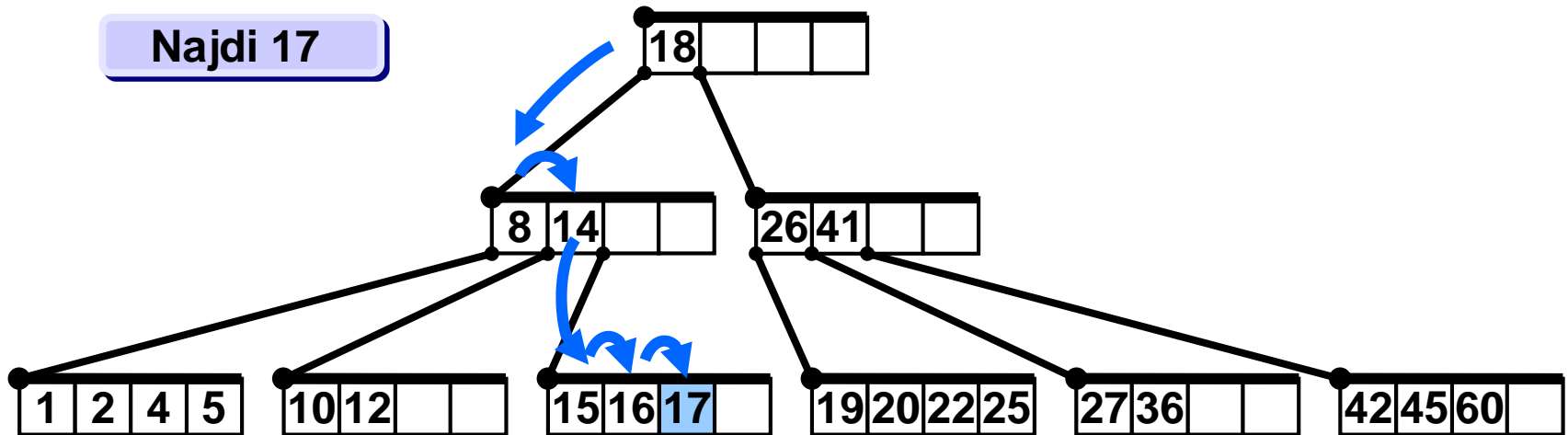
Výjimkou je kořen, který může mít i méně klíčů.



Všechny cesty z kořene do listu jsou stejně dlouhé.

# Find v B-stromu

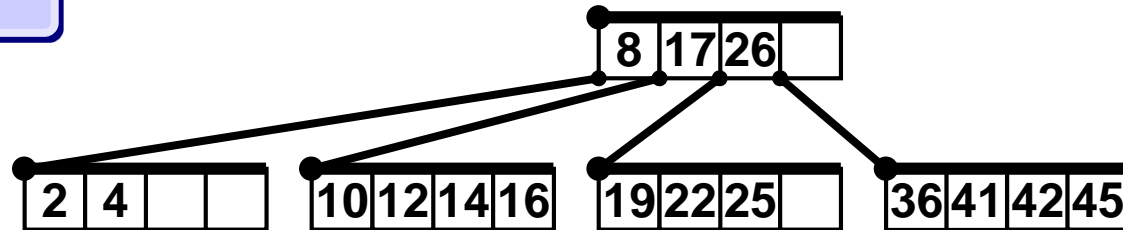
Najdi 17



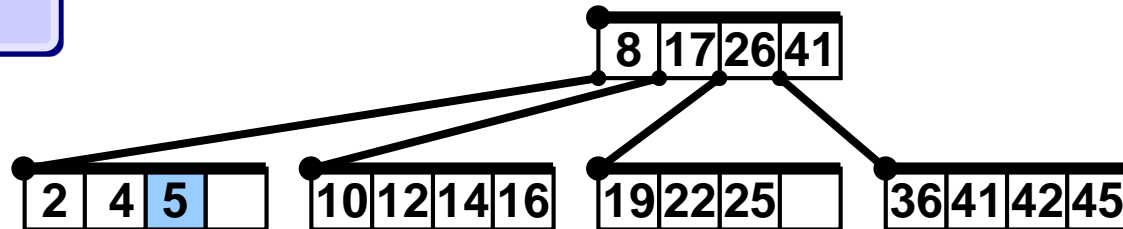
V uzlu se vyhledává sekvenčně  
nebo půlením intervalu.

# Insert v B-stromu

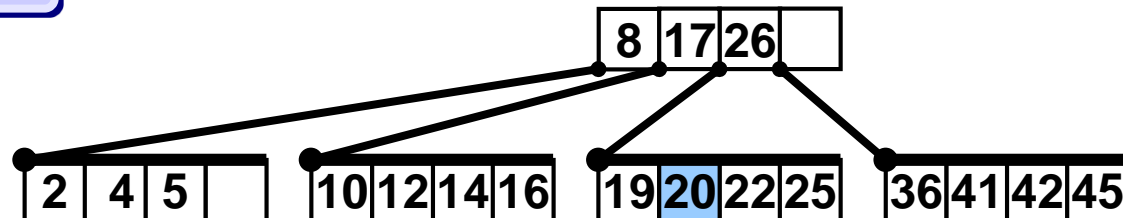
B-strom



Vlož 5

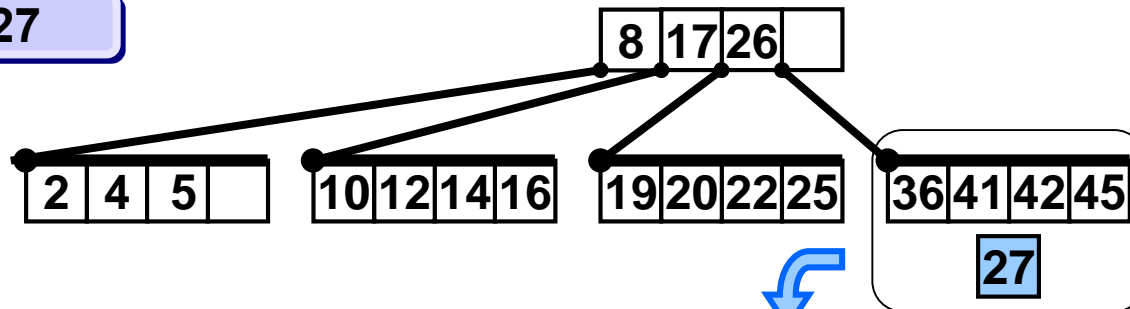


Vlož 20

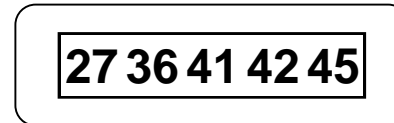


# Insert v B-stromu

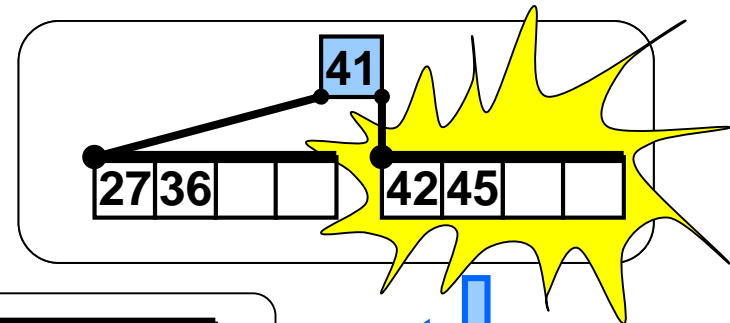
Vlož 27



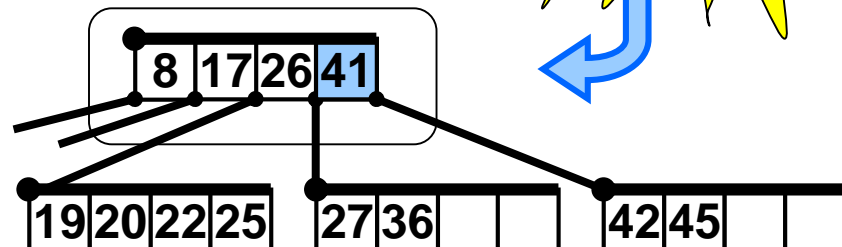
Seřad' mimo strom.



Vyber medián,  
vyvoř nový uzel,  
přesuň do něj hodnoty  
větší než medián.



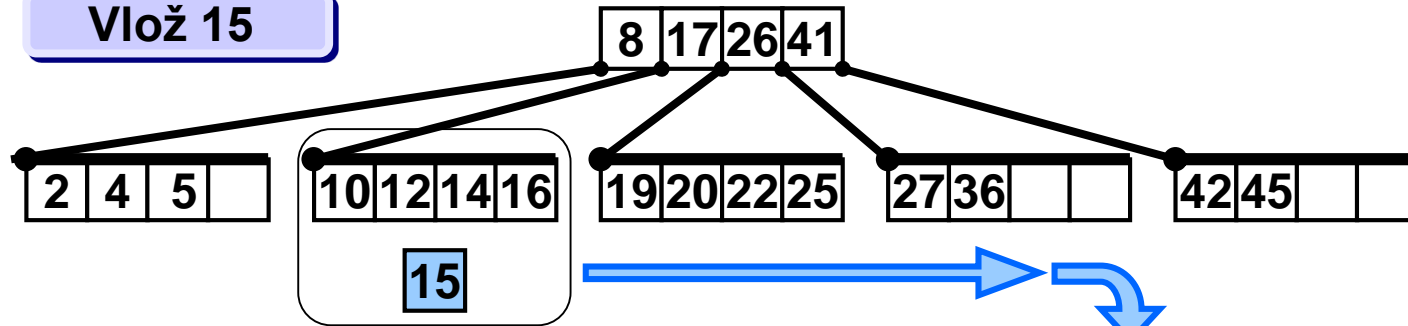
Medián zkus vložit  
do rodiče.



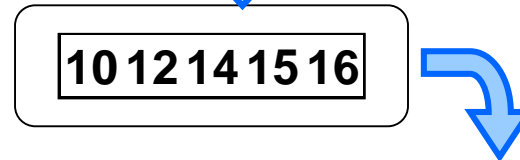
Zdařilo se.

# Insert v B-stromu

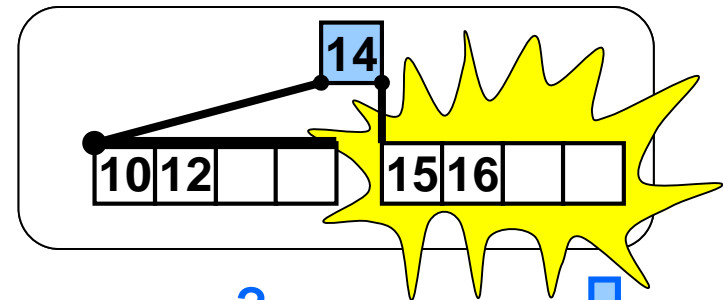
Vlož 15



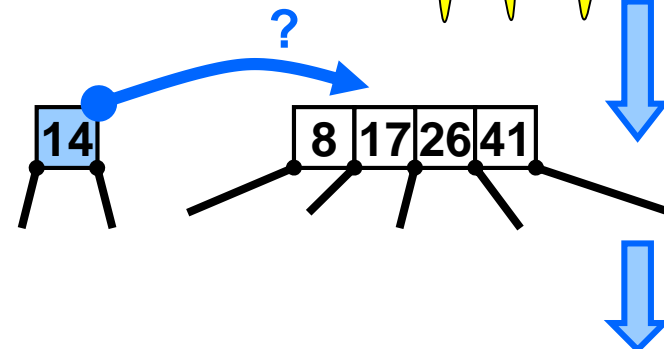
Seřad' mimo strom.



Vyber medián,  
vyvoř nový uzel,  
přesuň do něj hodnoty  
větší než medián.



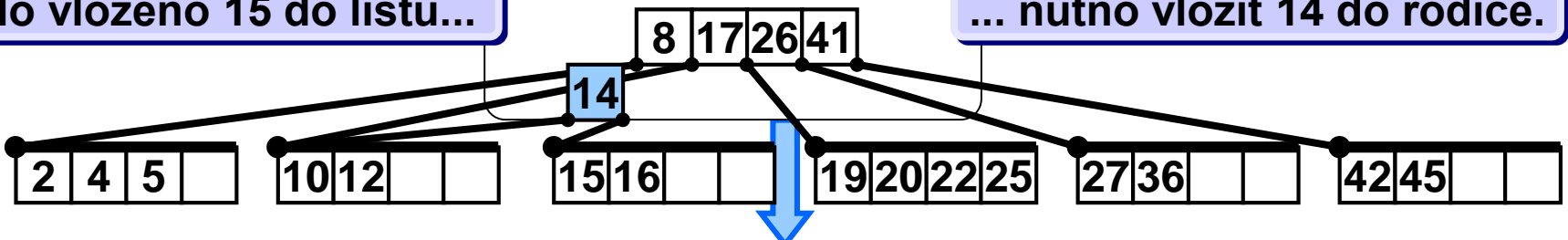
Medián zkus vložit  
do rodiče.



# Insert v B-stromu

Bylo vloženo 15 do listu...

... nutno vložit 14 do rodiče.

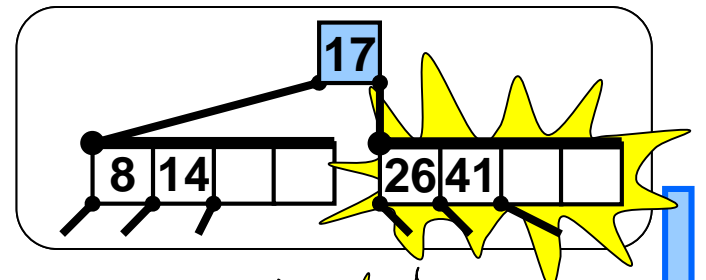


Rodič je zaplněn – Analogický další postup směrem ke kořeni.

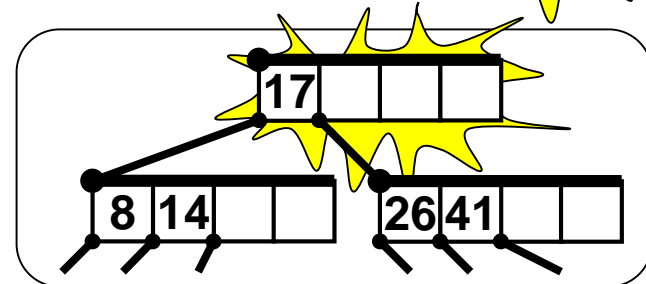
Seřad' mimo strom.

8 14 17 26 41

Vyber medián,  
vyvoř nový uzel,  
přesuň do něj hodnoty  
větší než medián.

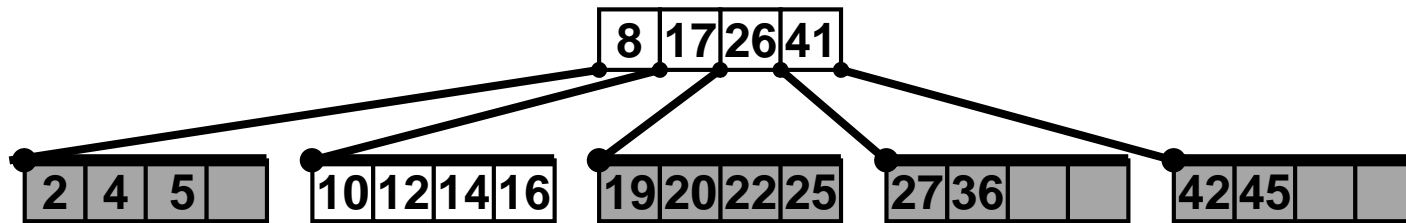


Medián nelze vložit do  
rodiče, žádný rodič není,  
tedy se zřídí nový kořen.

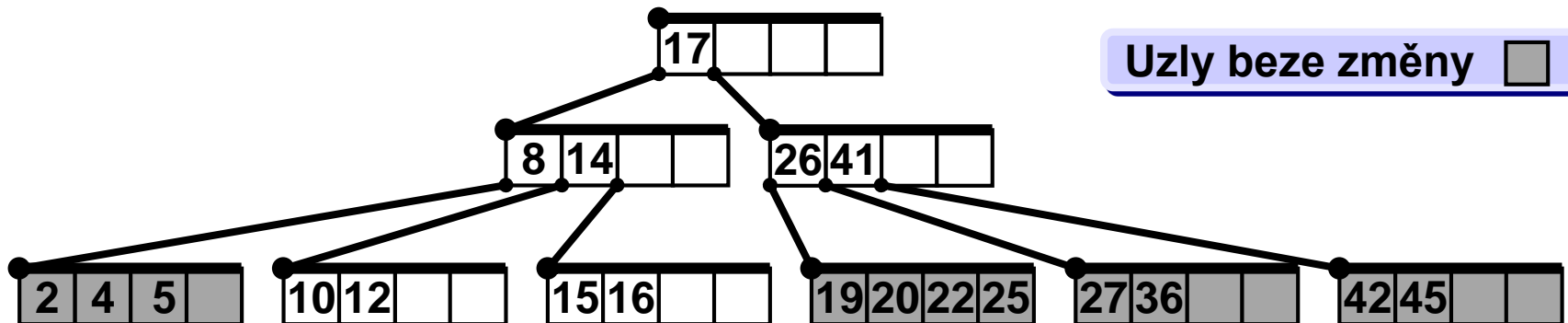


# Insert v B-stromu

Rekapitulace - vlož 15



Vlož 15

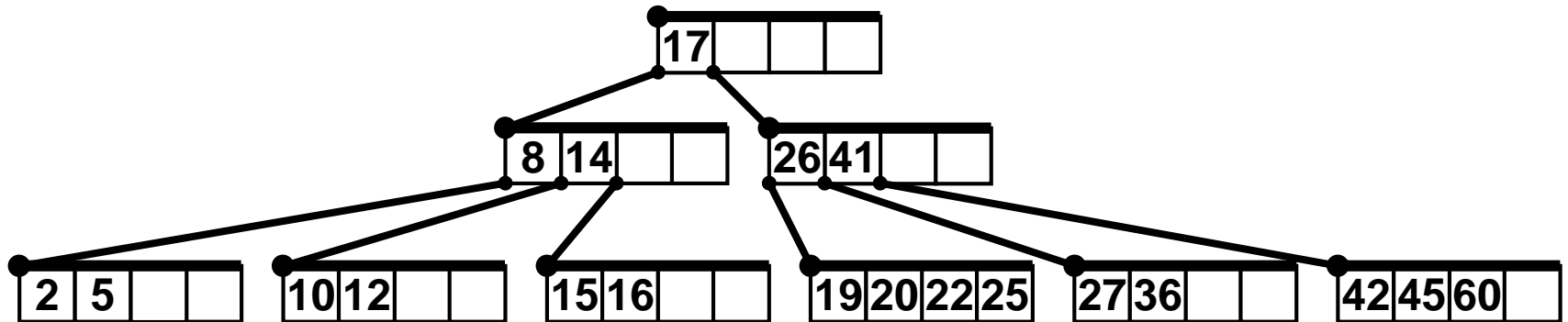
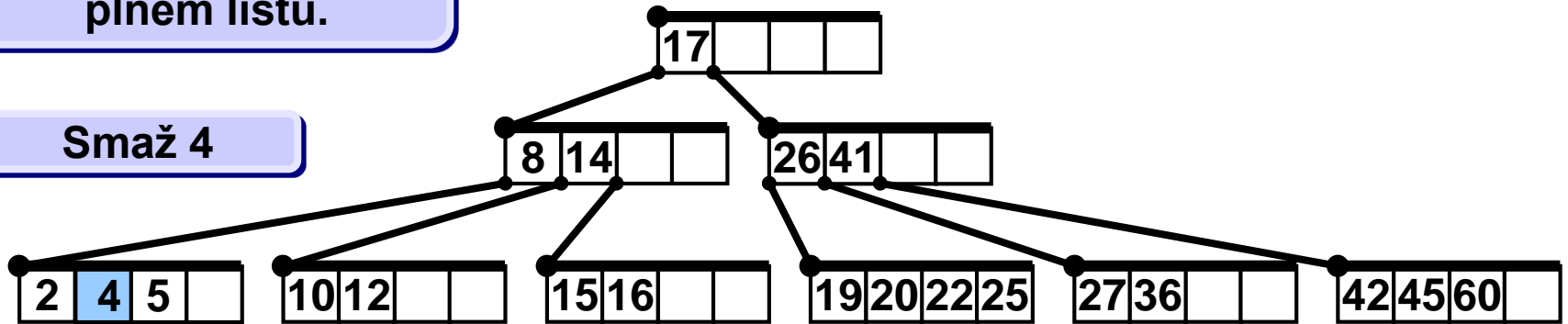


V každém patře přibyl jeden uzel, kromě toho přibyl nový kořen, strom ale roste směrem "vzhůru", zůstává ideálně vyvážený.

# Delete v B-stromu

Mazání v dostatečně plném listu.

Smaž 4

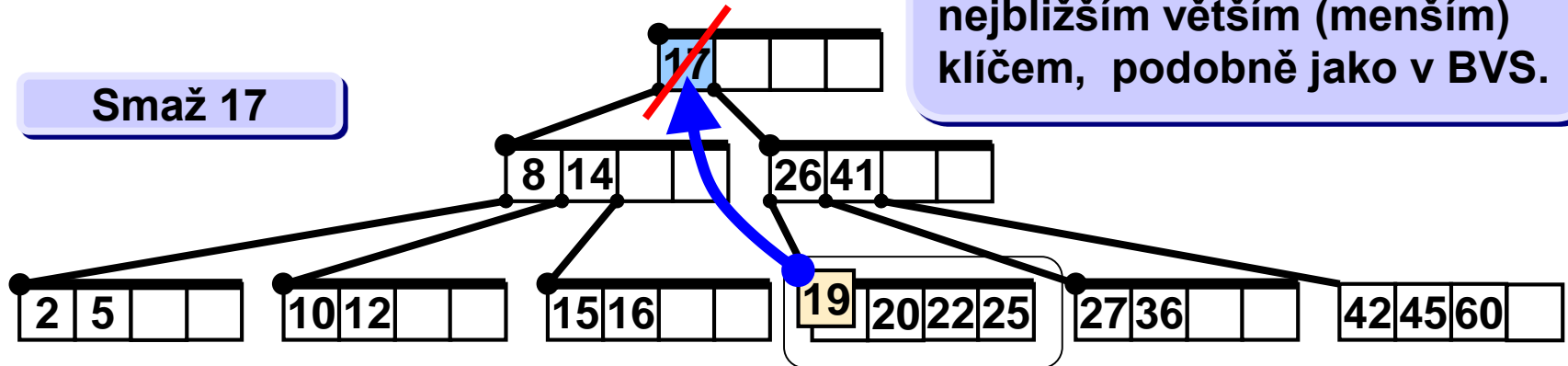




# Delete v B-stromu

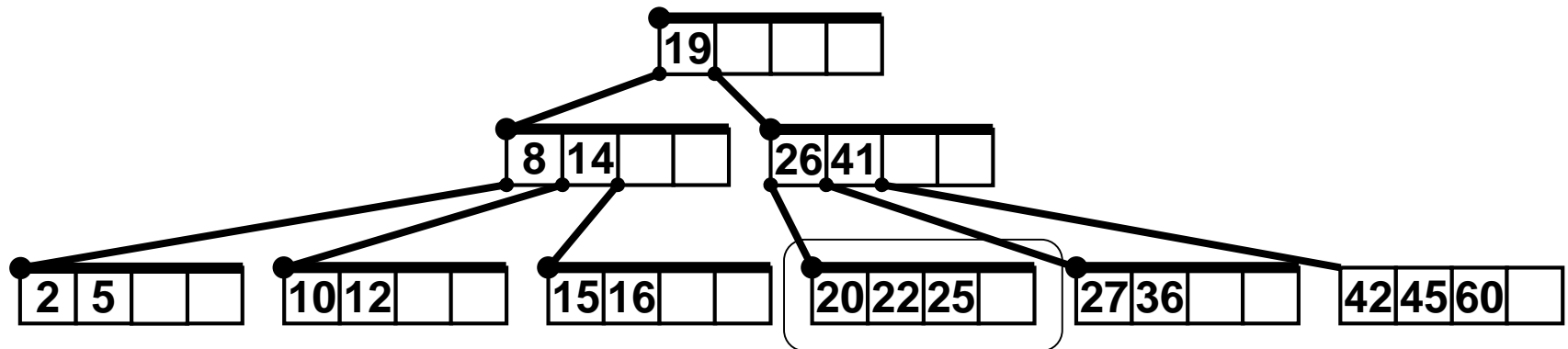
Mazání ve vnitřním uzlu

Smaž 17



Smazaný klíč se nahradí  
nejbližším větším (menším)  
klíčem, podobně jako v BVS.

Nejbližší větší (menší) klíč je vždy v B-stromu v listu,  
má-li tento list dostatečný počet klíčů, jsme hotovi.

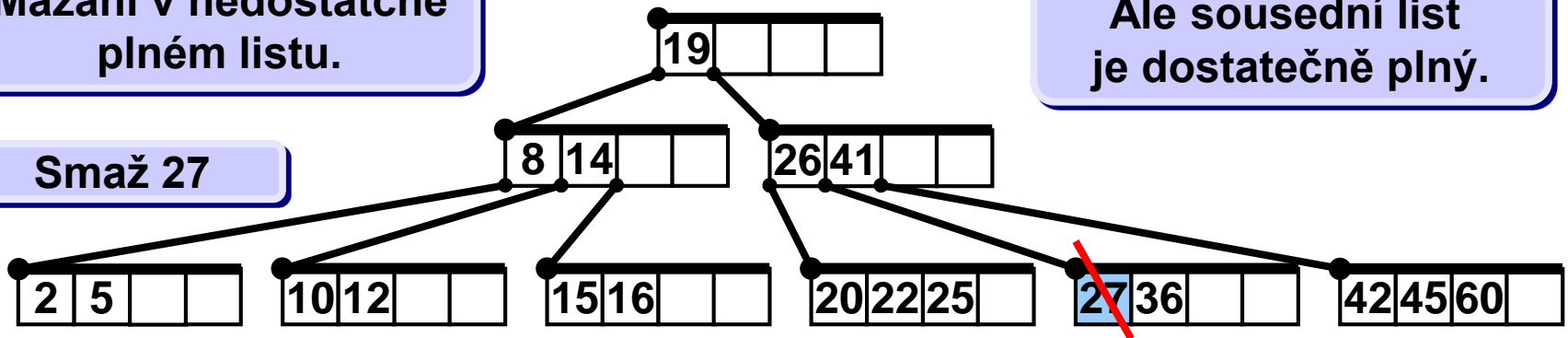


# Delete v B-stromu

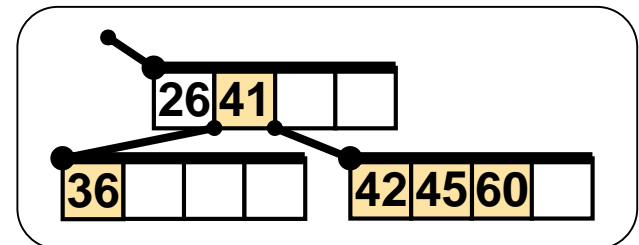
Mazání v nedostatečně plném listu.

Ale sousední list je dostatečně plný.

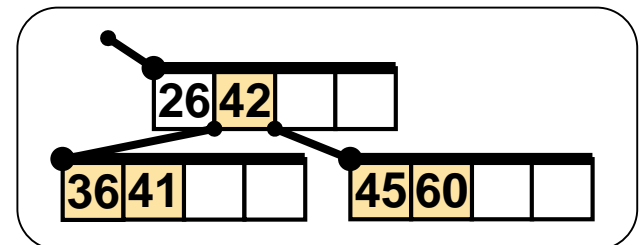
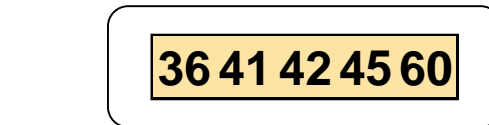
Smaž 27



Sjednot' klíče s klíči v sousedním listu a s dělicím klíčem v rodiči a seřad'.

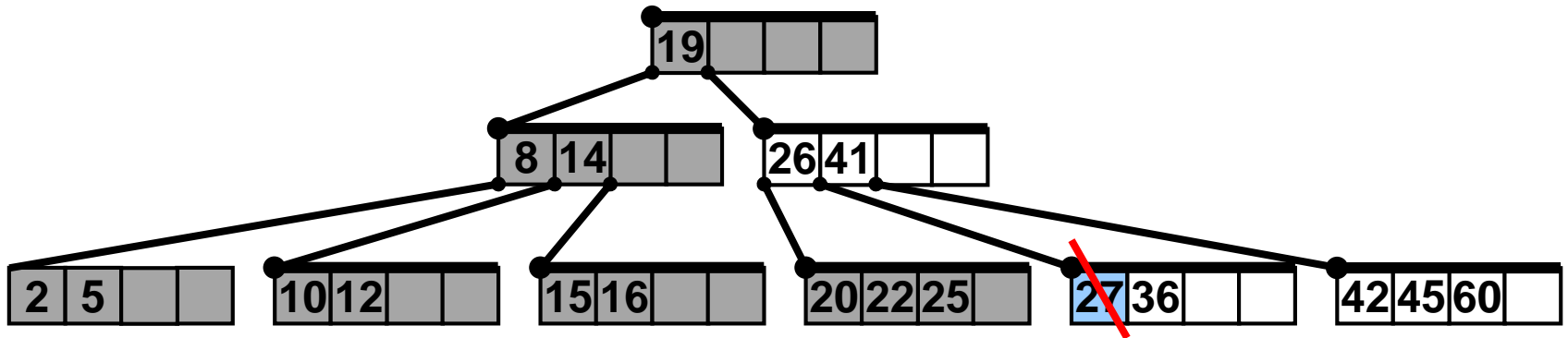


Medián sjednocení vlož na místo původně dělicího klíče, menší a větší klíče než medián rozděl do levého a pravého listu.



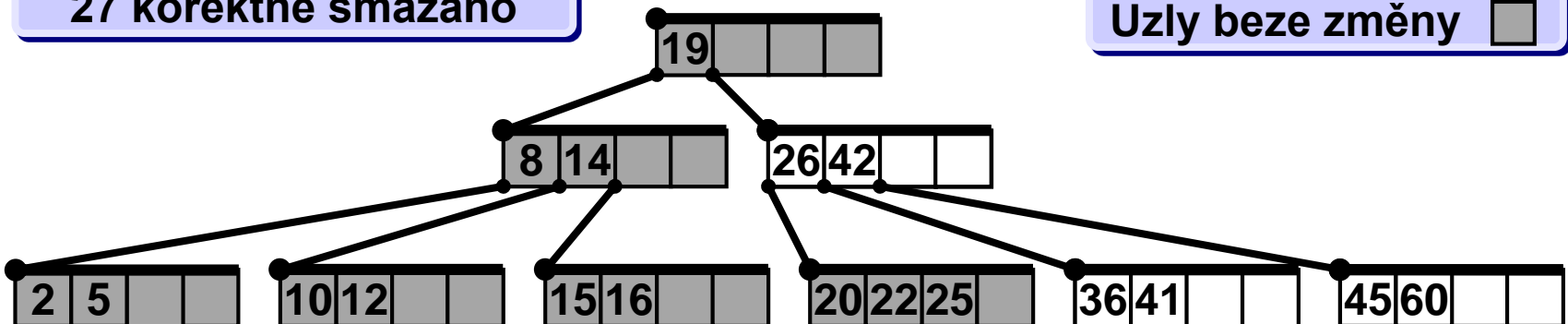
# Delete v B-stromu

Rekapitulace - smaž 27



27 korektně smazáno

Uzly beze změny

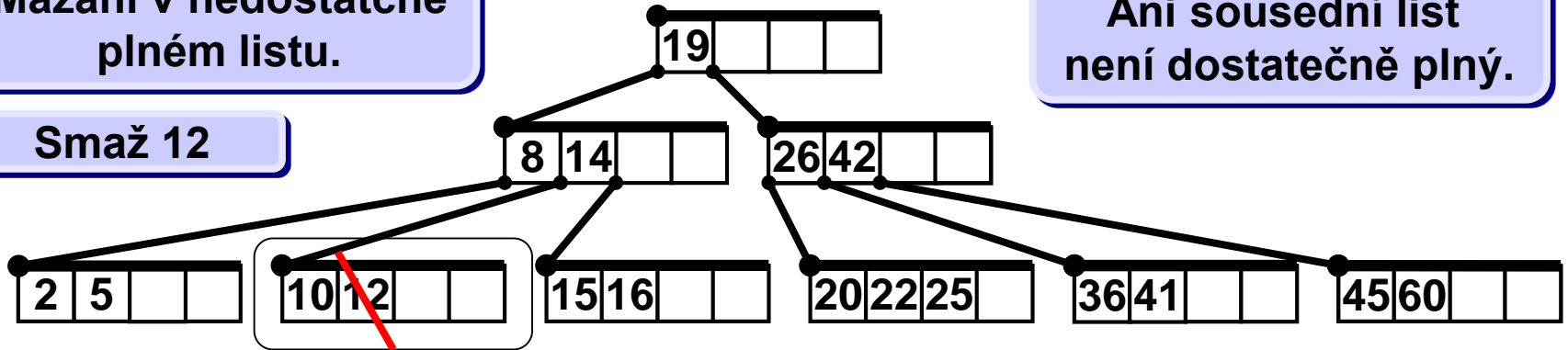


# Delete v B-stromu

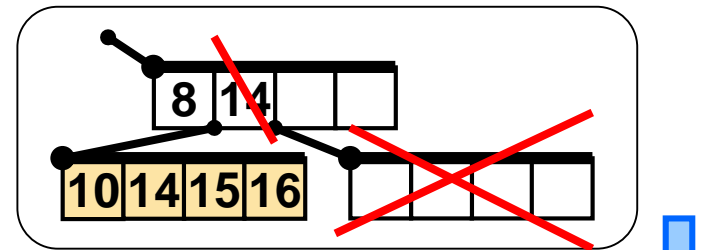
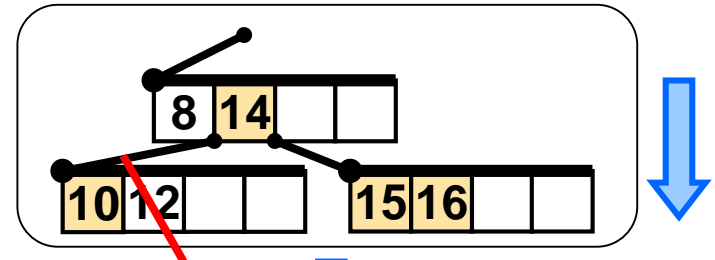
Mazání v nedostatečně plném listu.

Smaž 12

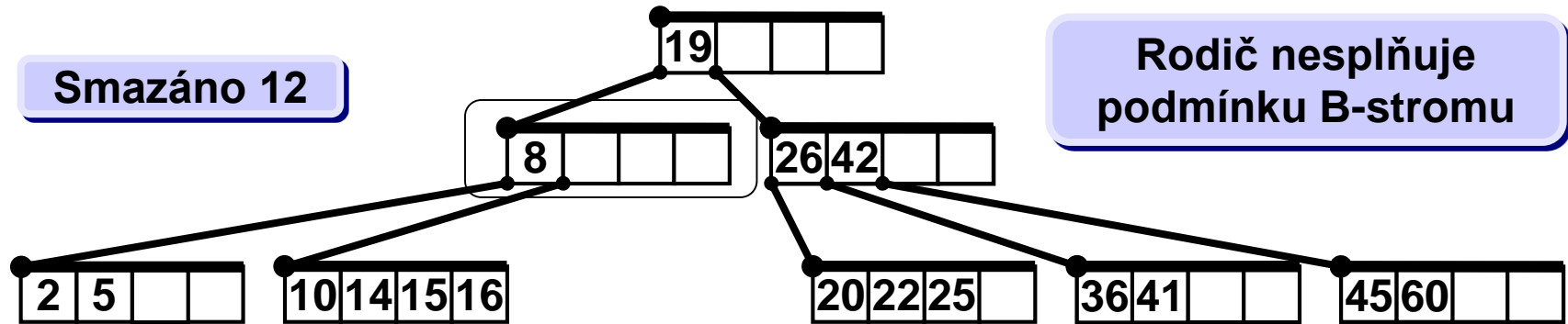
Ani sousední list není dostatečně plný.



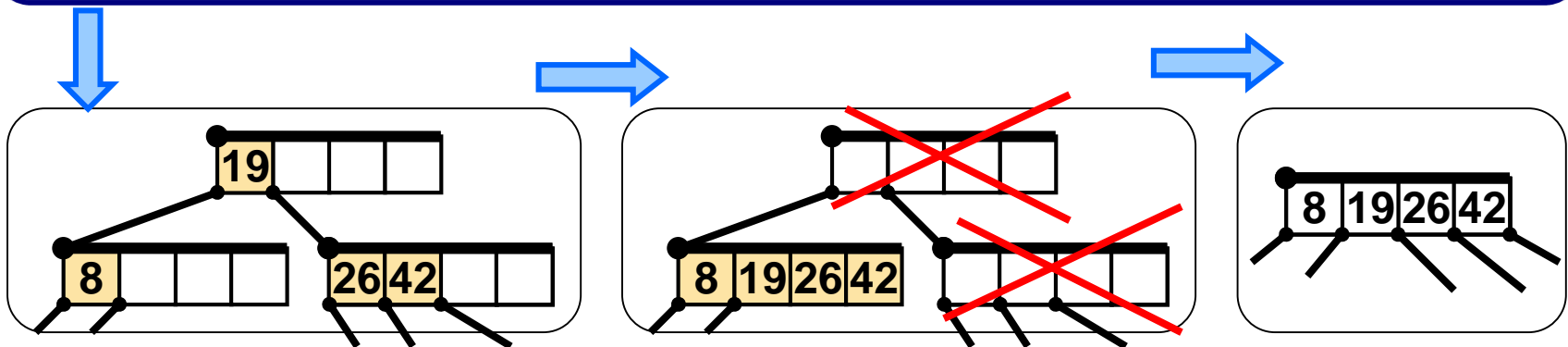
Sjednot' klíče s klíči v sousedním listu a s dělicím klíčem v rodiči a seřad'. Vše vlož do původního listu, sousední list smaž, dělicí klíč v rodiči také smaž.



# Delete v B-stromu

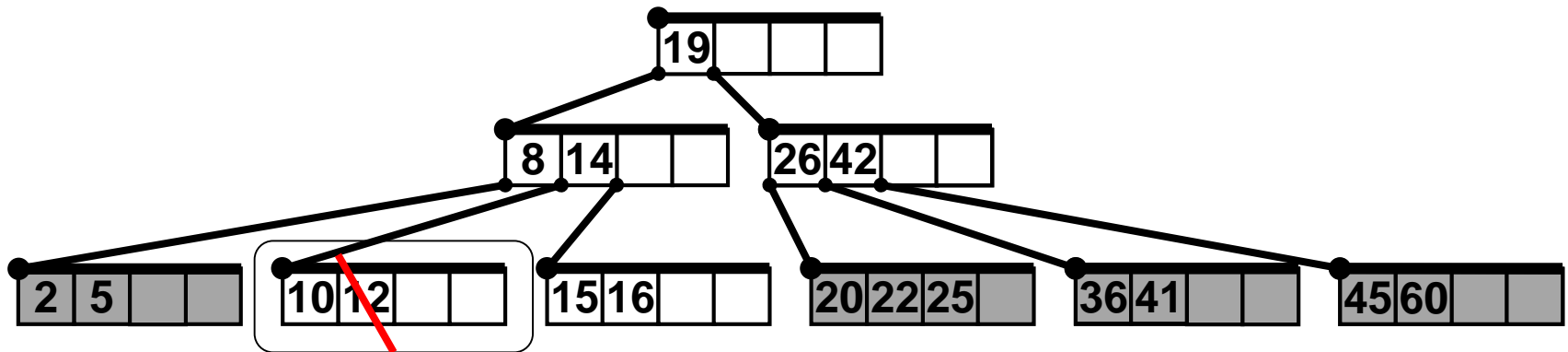


Rodič, který poskytl klíč potomku, není dostatečně plný.  
Aplikujeme na něj (a případně iterativně na jeho rodiče) tentýž postup  
spojení klíčů a sousedních uzlů a přesun dělicího prvku z rodiče.

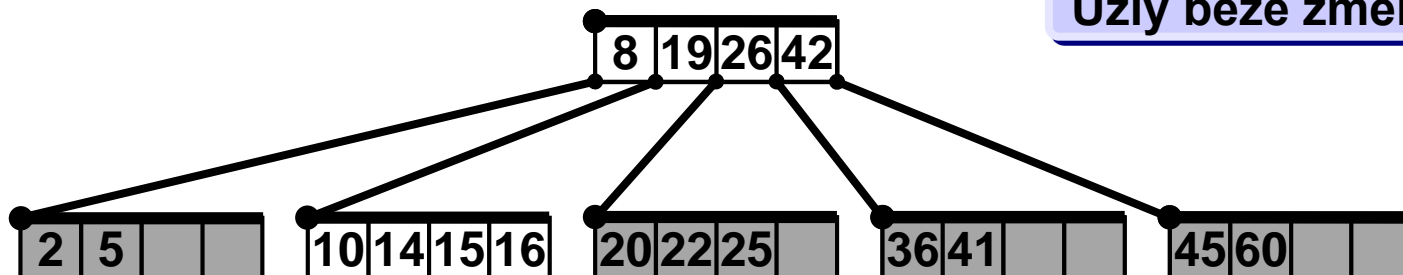


# Delete v B-stromu

Rekapitulace - smaž 12



Smazáno 12 a strom byl adekvátně restrukturován.



Uzly beze změny

# Shrnutí

| Operace | B-strom řádu $k$ s $n$ uzly |
|---------|-----------------------------|
| Find    | $\Theta(k \cdot \log_k n)$  |
| Insert  | $\Theta(k \cdot \log_k n)$  |
| Delete  | $\Theta(k \cdot \log_k n)$  |



Jakou má Find asymptotickou složitost při použití binárního vyhledávání v uzlech?

Obecnější definice B-stromu:

- Je dána kapacita uzlů  $m \geq 2$ .
- Každý uzel má minimálně  $\lfloor \frac{m}{2} \rfloor$  klíčů (kromě kořene) a maximálně  $m$  klíčů.

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



# Audience Q&A Session

① Start presenting to display the audience questions on this slide.