

# VYHLEDÁVACÍ STROMY

## **AVL strom**

Operace Find, Insert, Delete  
Rotace L, R, LR, RL

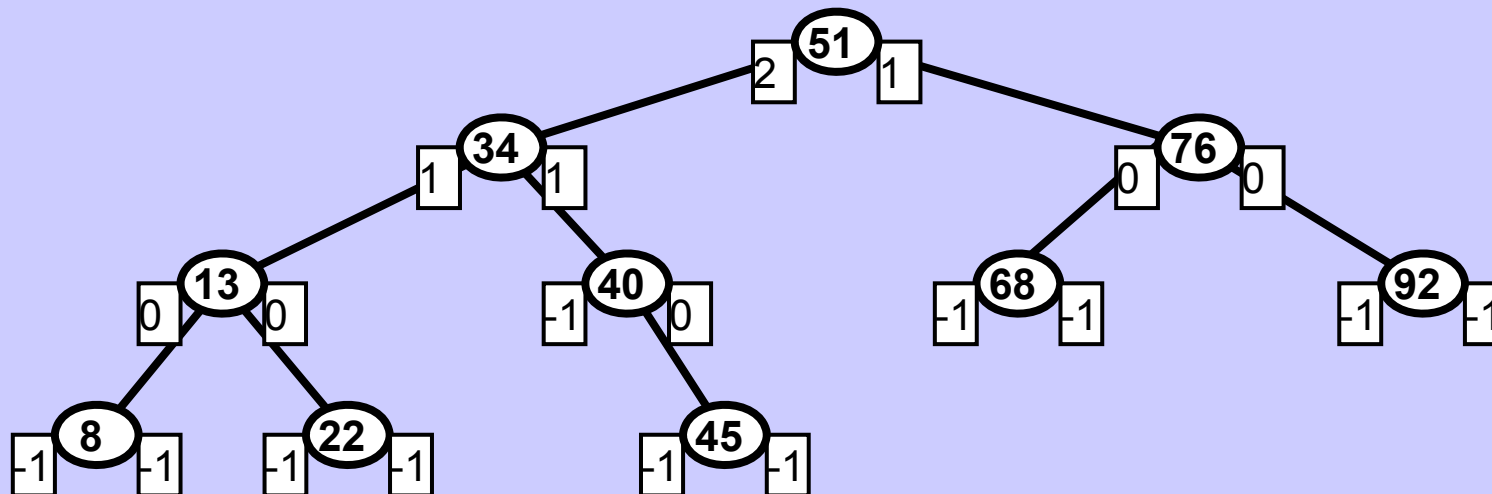
## **B-strom**

Operace Find, Insert, Delete

## AVL strom -- G.M. Adelson-Velskij & E.M. Landis, 1962

AVL strom je BVS s přidányi vlastnostmi, které jej udržují (téměř) vyvážený.

AVL má také operace Find, Insert, Delete.



Každý uzel registruje hloubku svého levého a pravého podstromu, hloubka prázdného stromu je -1.

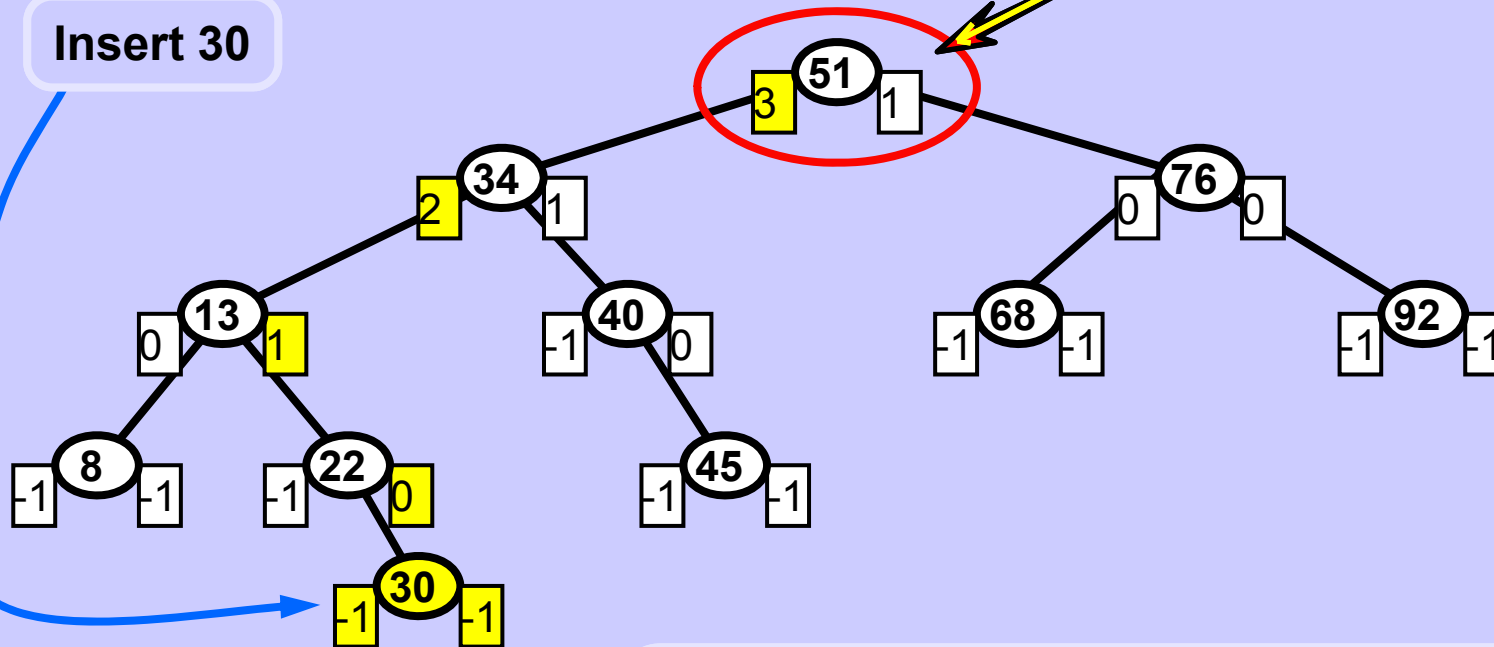
V každém uzlu je rozdíl výšek obou podstromů roven -1, 0, 1.

V tomto případě: Výška stromu = hloubka stromu

Vložení uzlu může způsobit rozvážení AVL stromu.

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

Insert 30

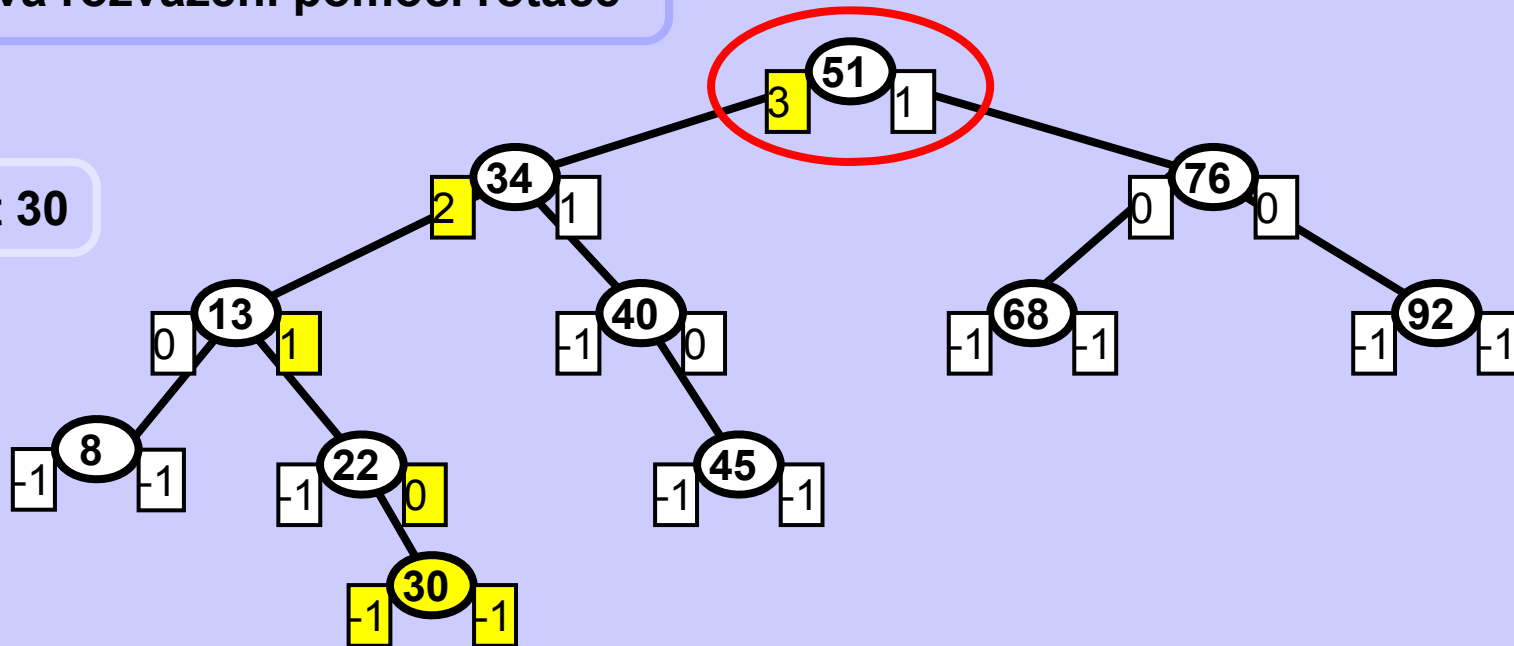


Změněné hloubky

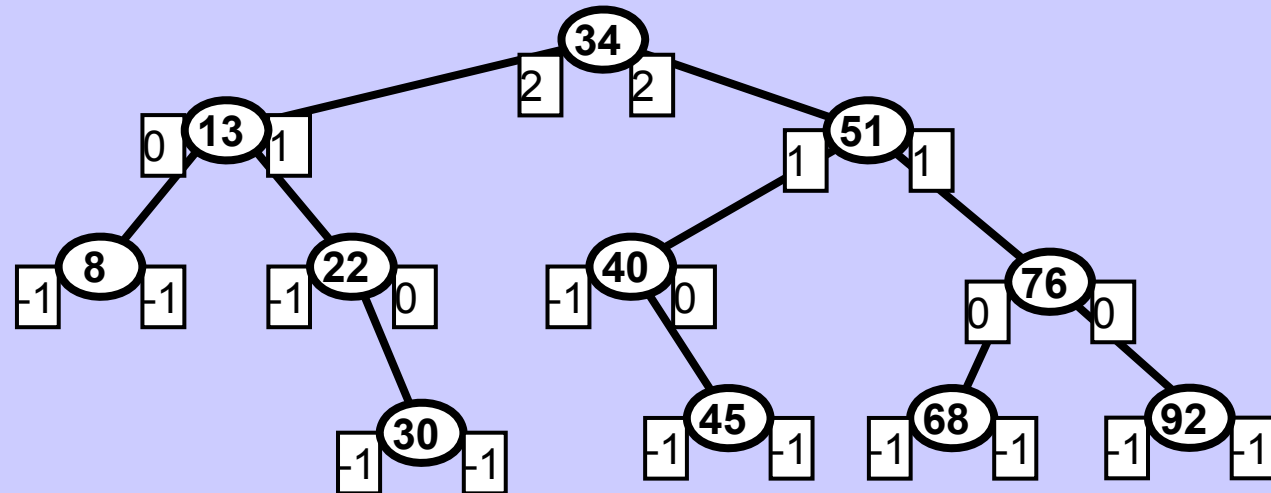
Levý podstrom uzlu 51 je příliš hluboký, strom přestal být AVL.

## Náprava rozvážení pomocí rotace

Insert 30

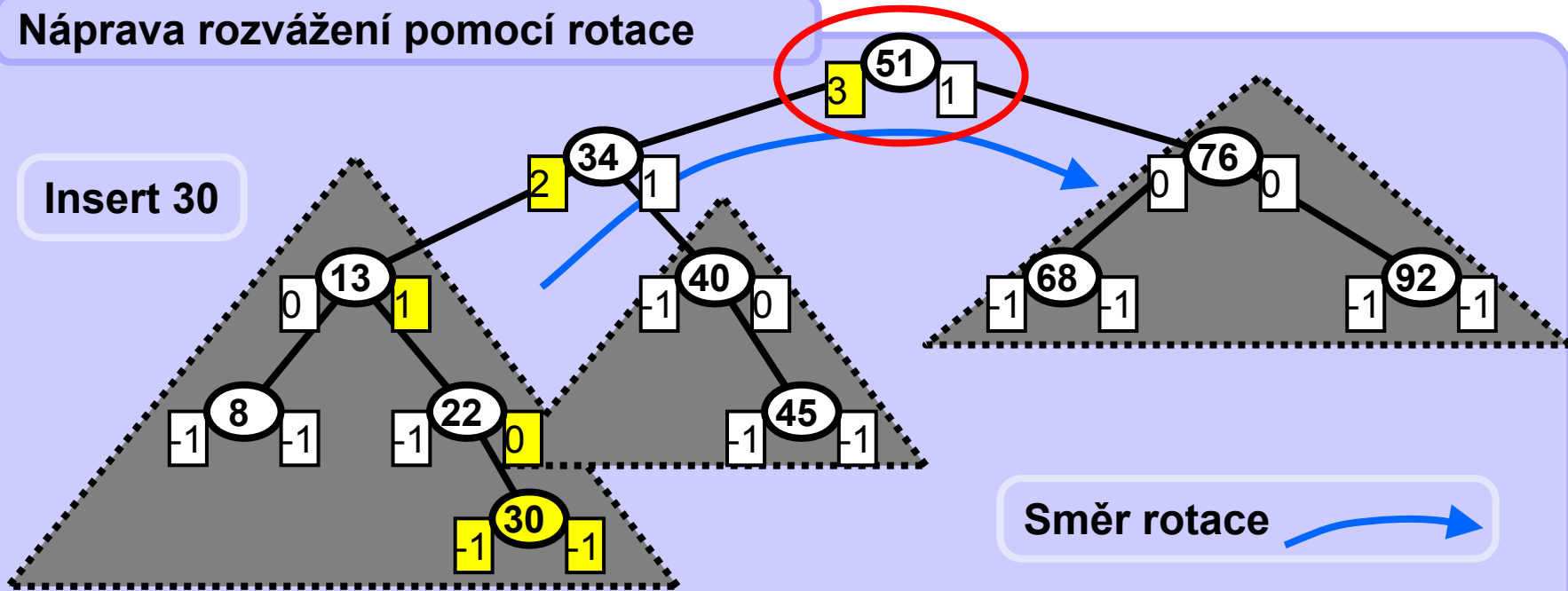


Vyvážený strom  
po pravé  
jednoduché rotaci,  
tzv. R rotaci



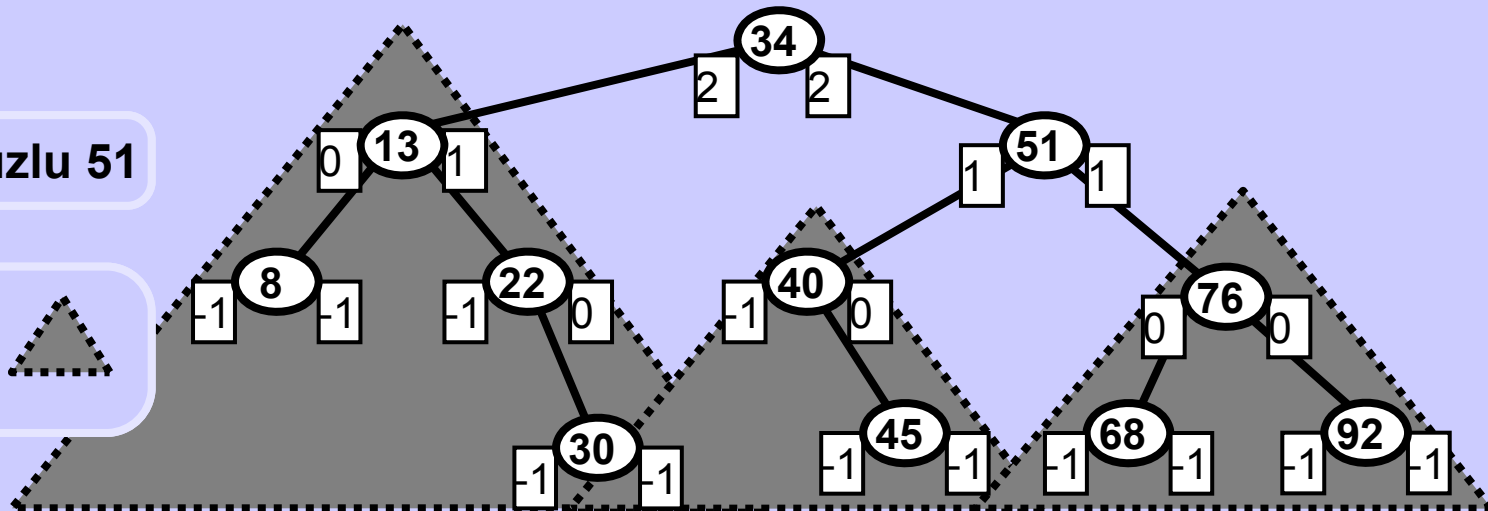
# Náprava rozvážení pomocí rotace

Insert 30



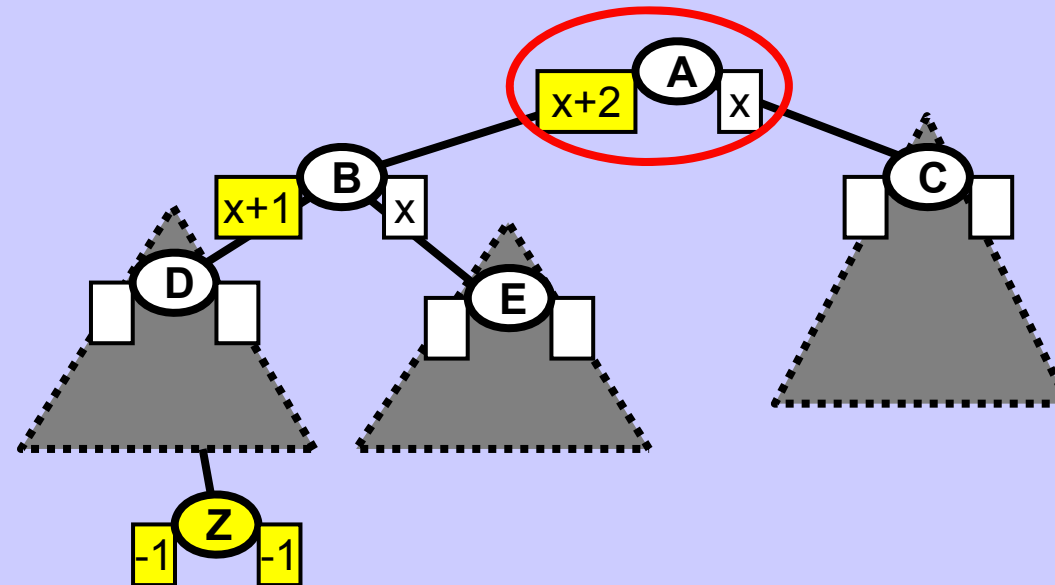
R rotace v uzlu 51

Podstromy  
beze změn

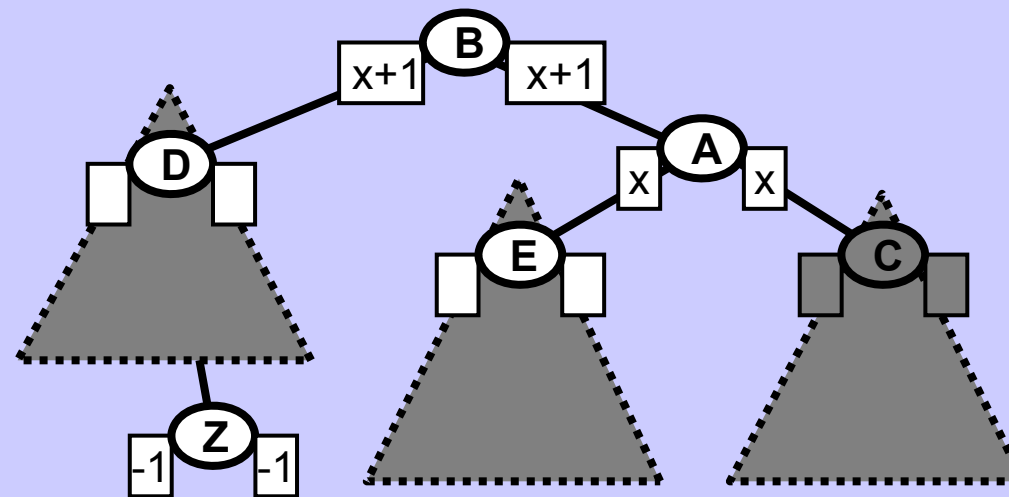


## Rotace R obecně

Před

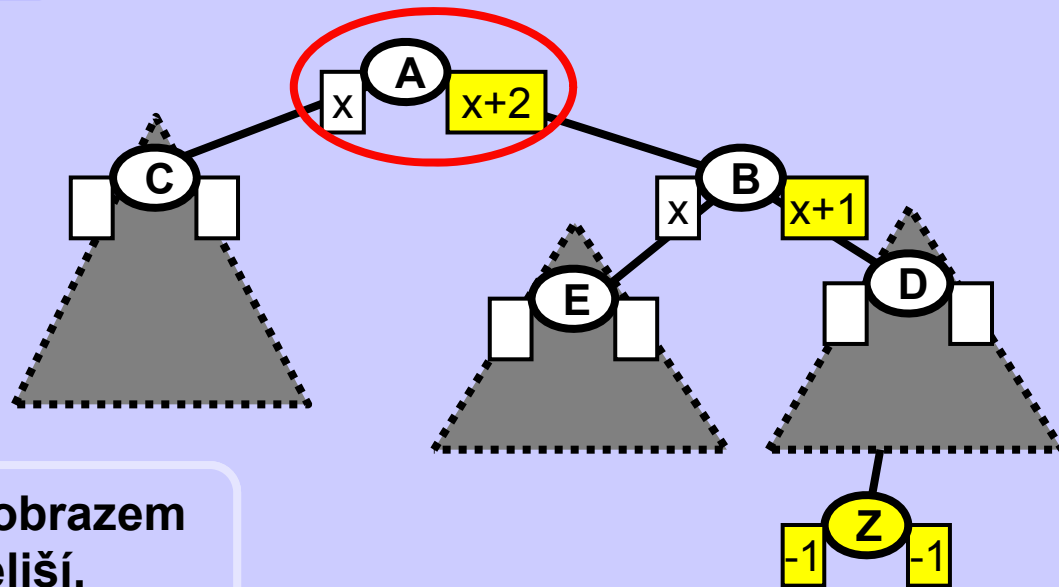


Po



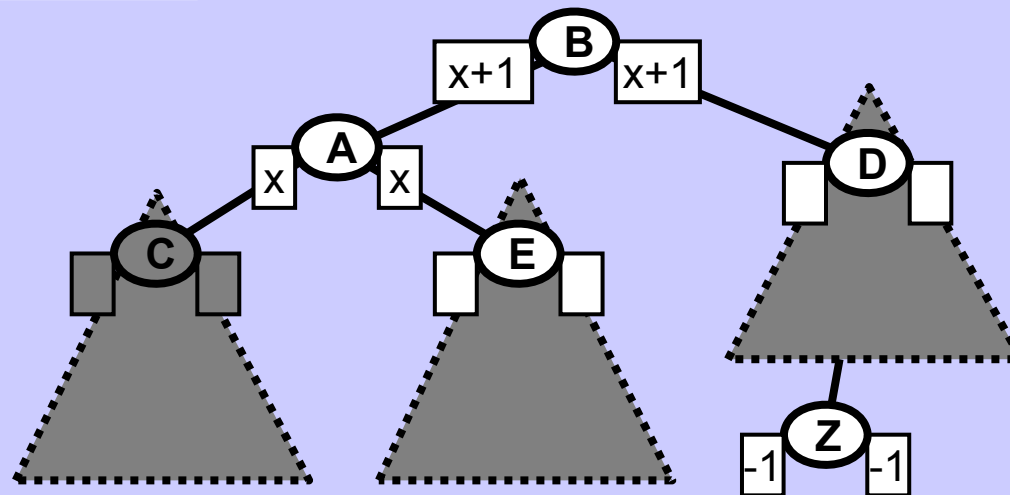
## Rotace L obecně

Před

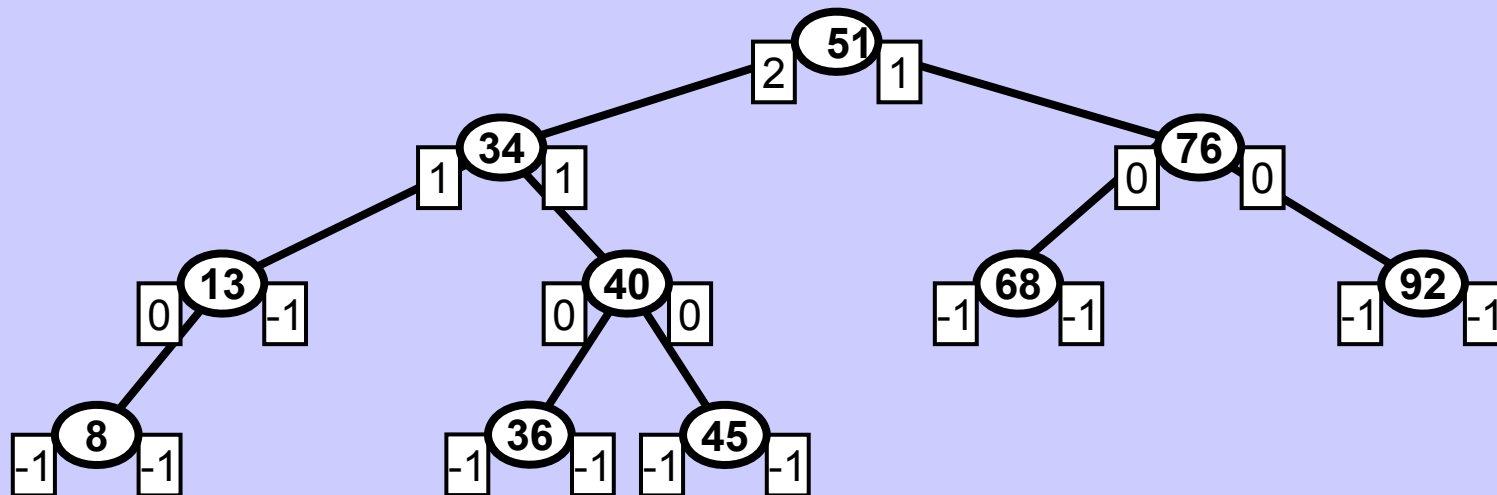


Rotace L je symetrickým obrazem rotace R, jinak se od ní neliší.

Po



## AVL strom



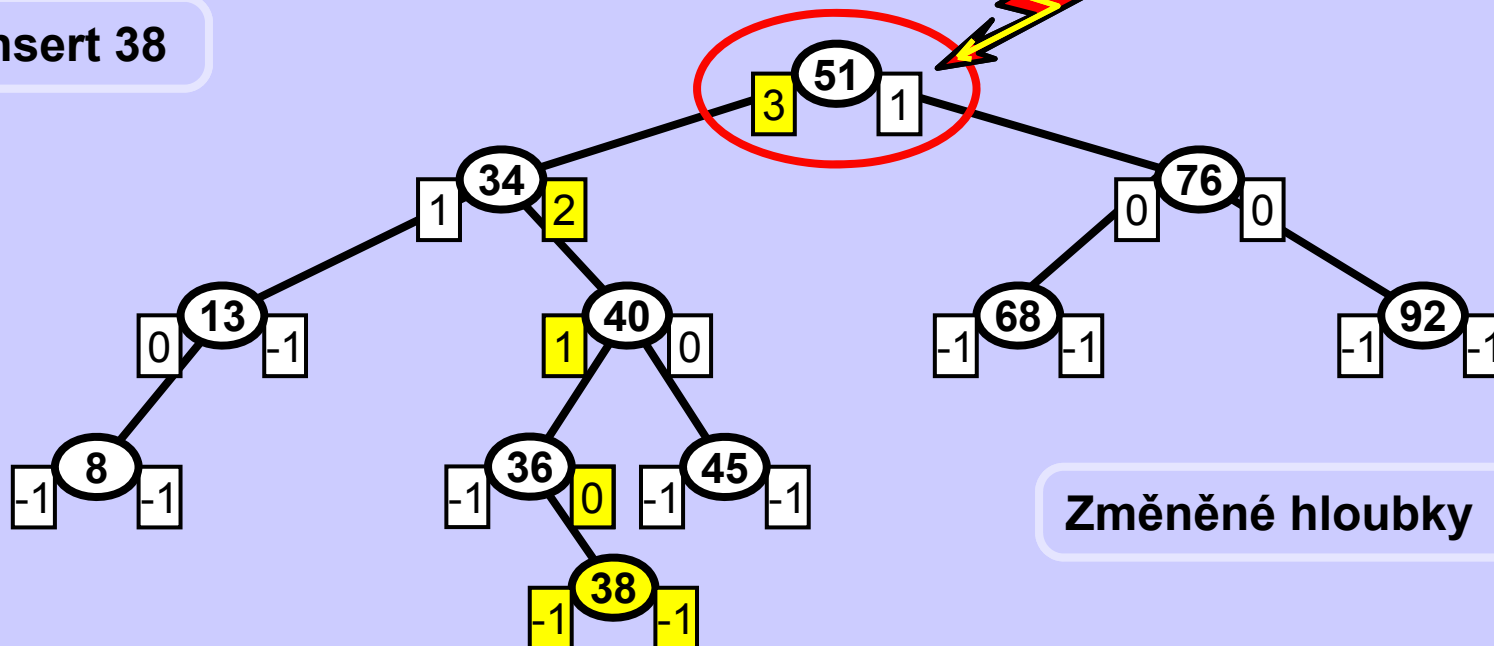
Strom pro demonstraci LR rotace



Vložení uzlu může způsobit rozvážení stromu.

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

Insert 38

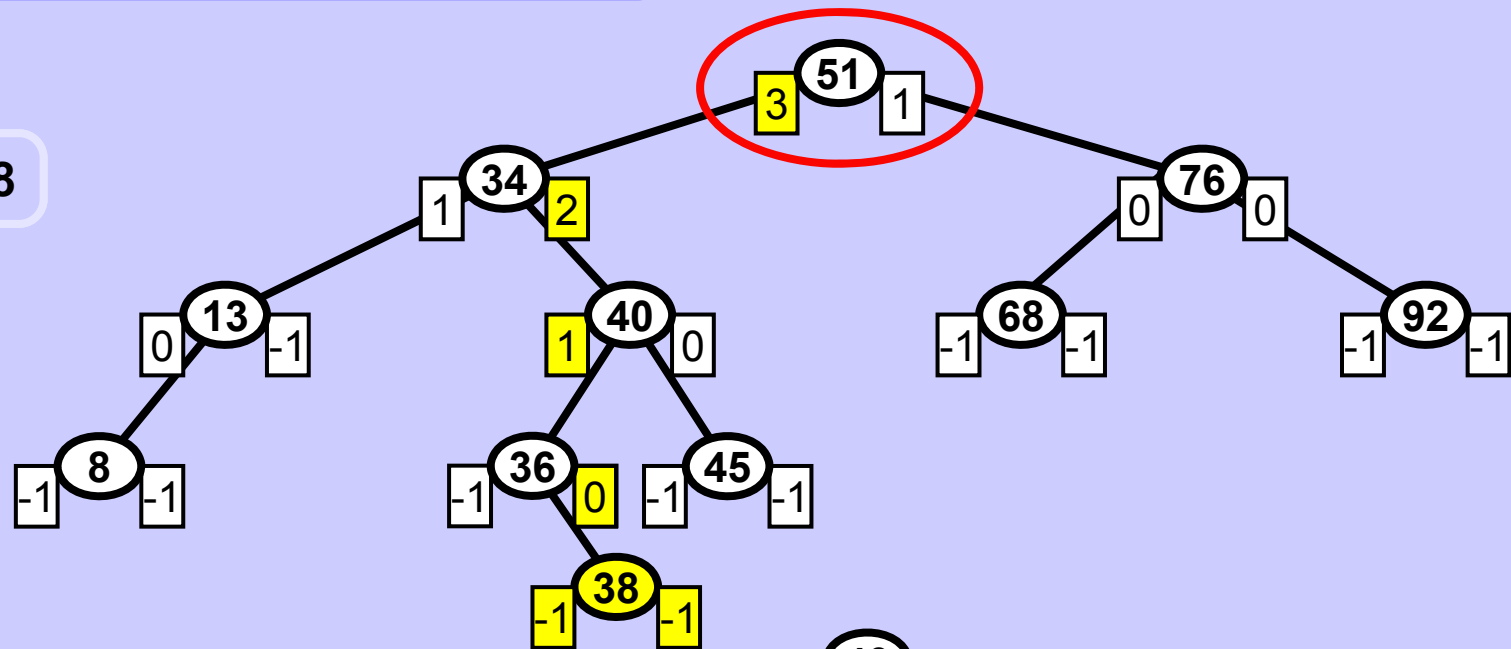


Levý podstrom uzlu 51 je příliš hluboký, strom přestal být AVL.

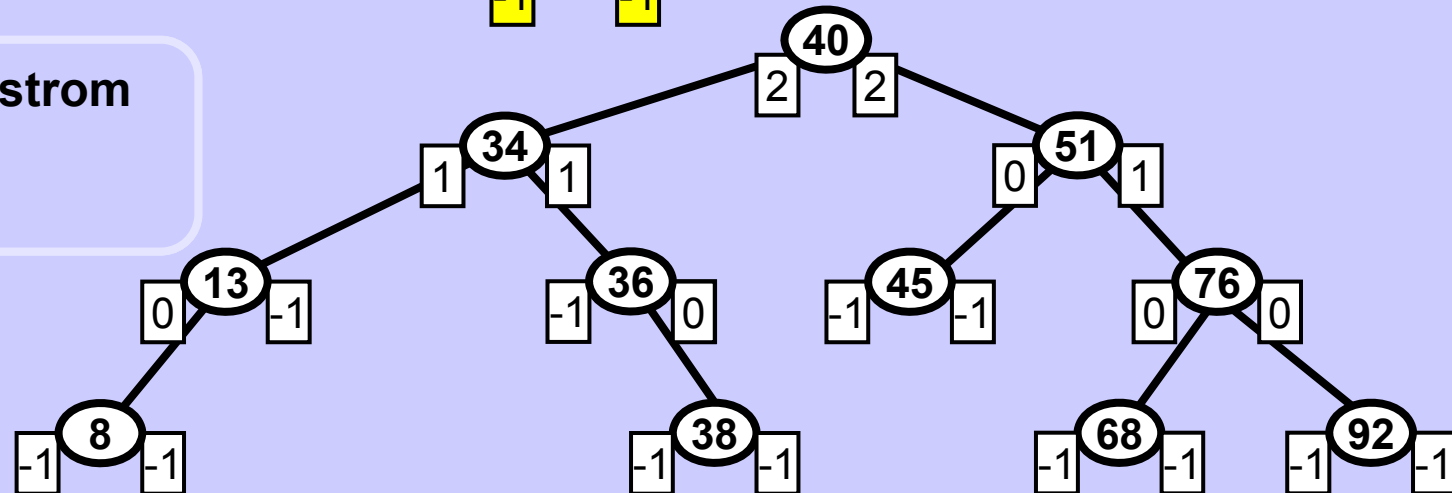
Použití rotace R by nepomohlo, příliš hlubokým by se stal původní pravý podstrom uzlu 34 díky tomu, že by se jeho hloubka vůbec nezměnila a hloubka levého podstromu 51 by klesla.

## Náprava rozvážení pomocí LRrotace

Insert 38

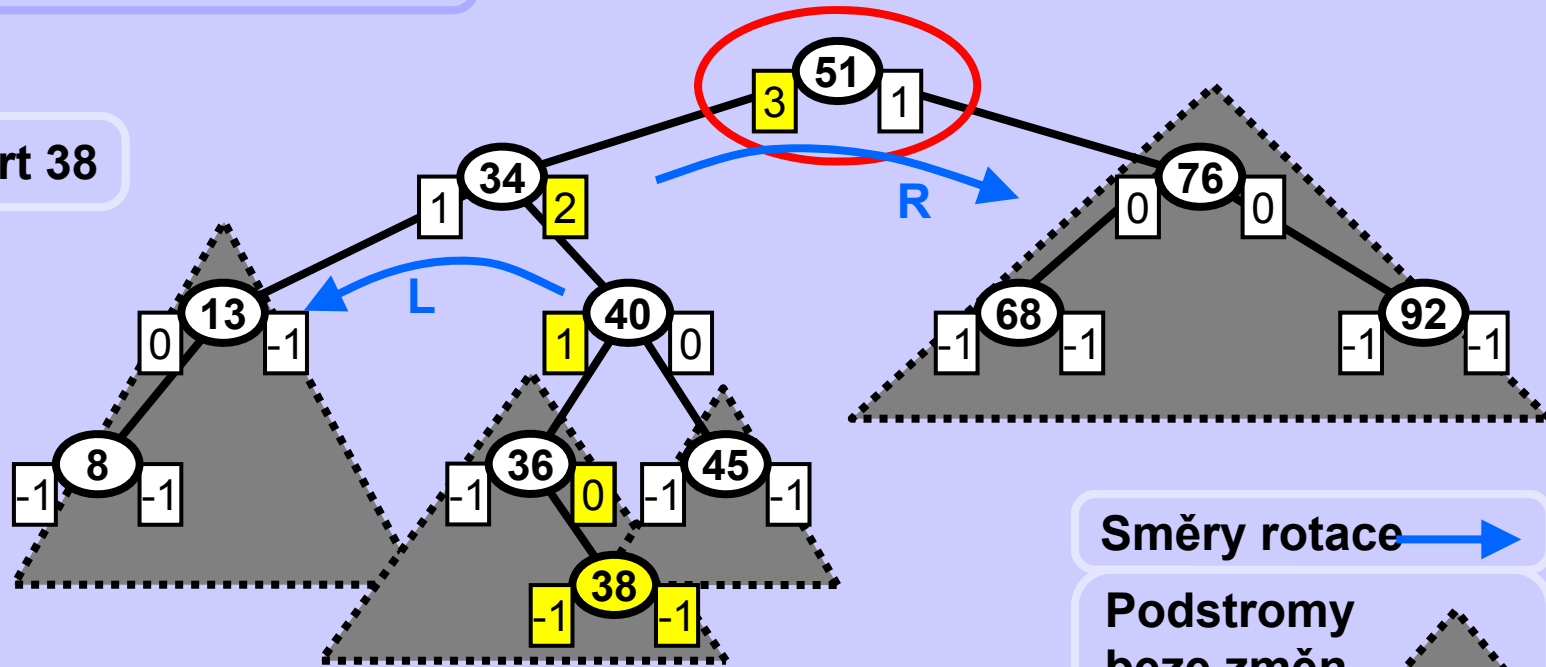


Vyvážený strom  
po dvojité  
LR rotaci

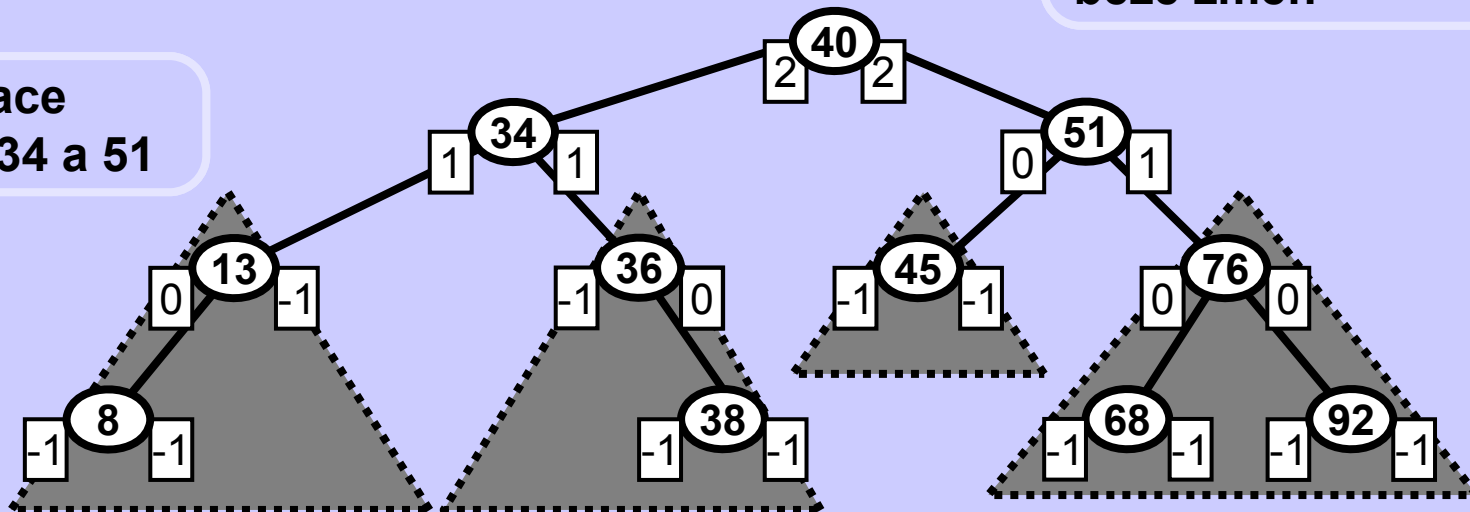


## Náprava rozvážení rotací

Insert 38

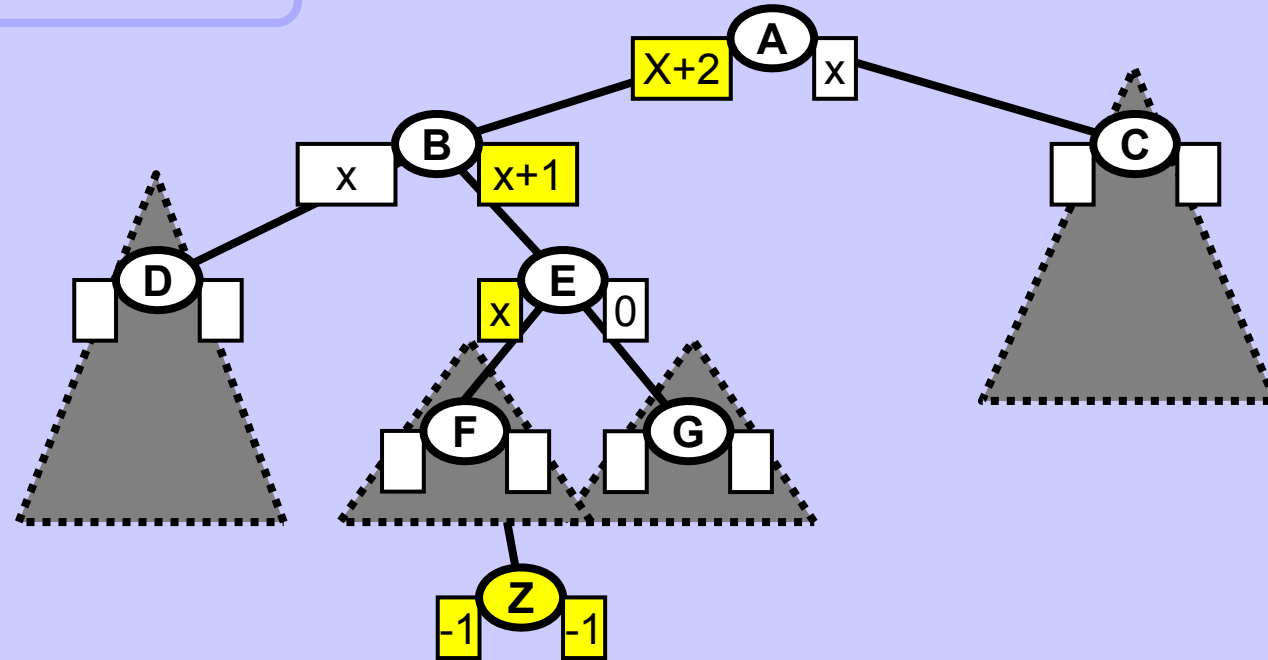


LR rotace  
v uzlu 34 a 51

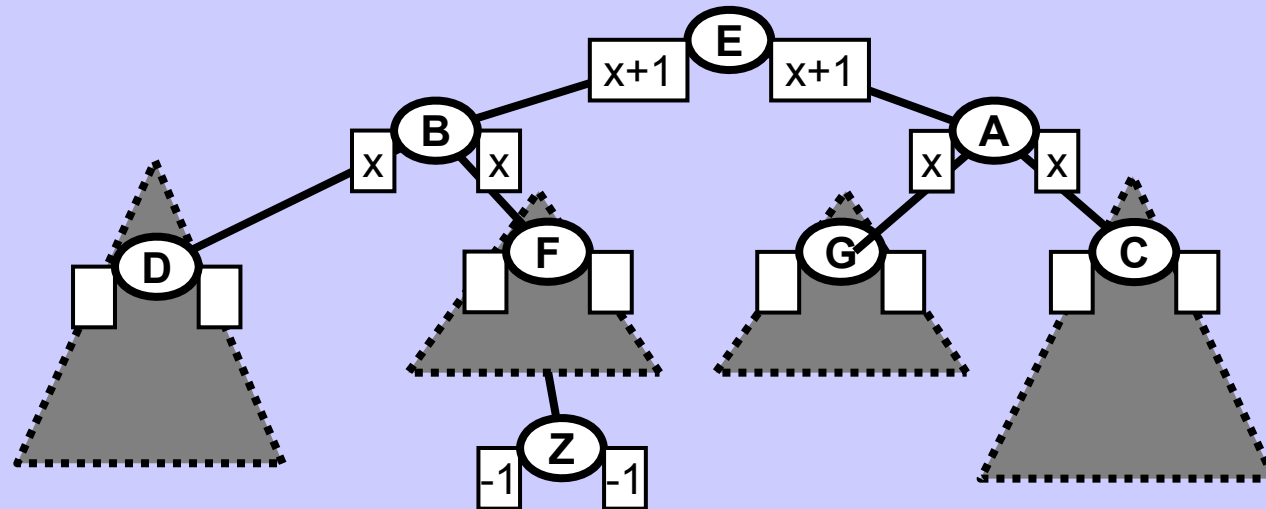


## Rotace LR obecně

Před

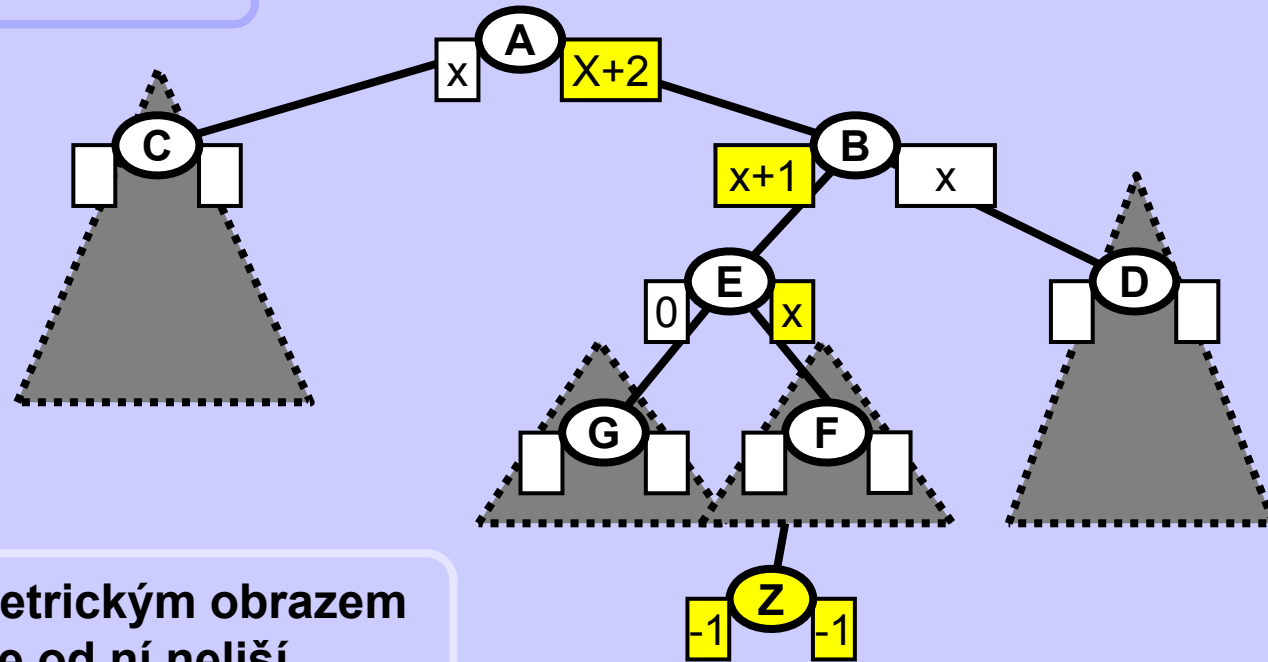


Po



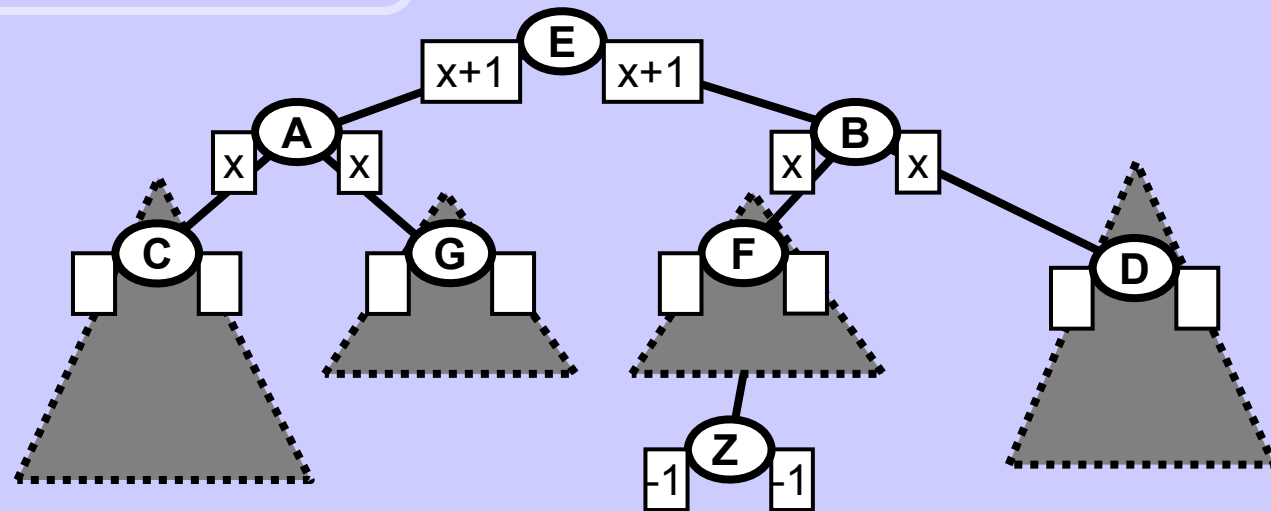
## Rotace RL obecně

Před



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

Po



## Pravidla pro aplikaci L, R, LR nebo RL rotací

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

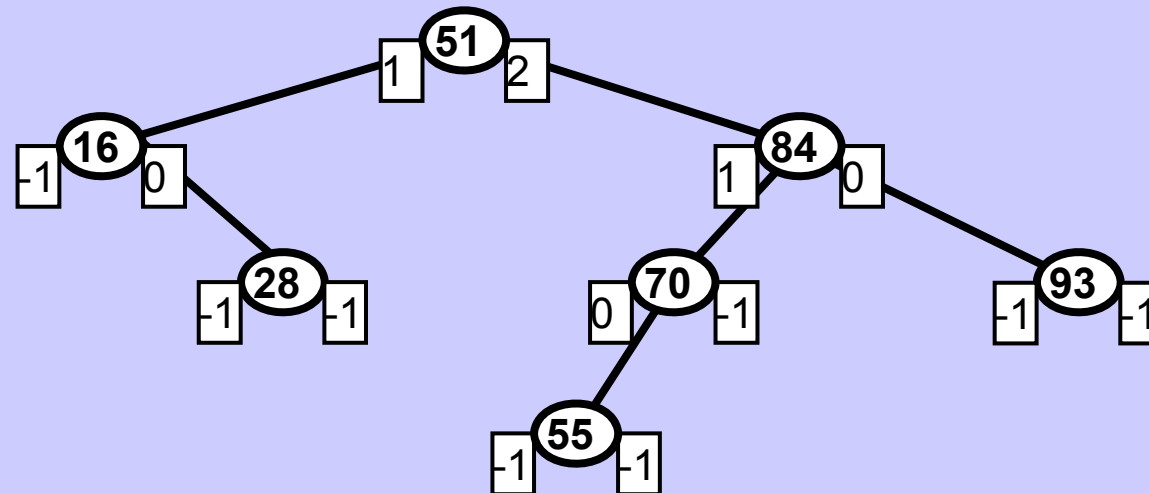
- \* dvěma hranami *doprava* nahoru provedeme v tomto uzlu R rotaci.
- \* dvěma hranami *doleva* nahoru, provedeme v tomto uzlu L rotaci.
- \* hranami *doleva a pak doprava* nahoru, provedeme v tomto uzlu LR rotaci.
- \* hranami *doprava a pak doleva* nahoru, provedeme v tomto uzlu RL rotaci.

Po provedení jedné rotace po operaci Insert je AVL strom opět vyvážen.

Po provedení jedné rotace po operaci Delete (viz dále) strom vyvážen být nemusí, je nutno kontrolovat a případně aktualizovat vyvážení až ke kořeni včetně.

## Delete v AVL stromu

## Strom pro demonstraci rotace po smazání uzlu



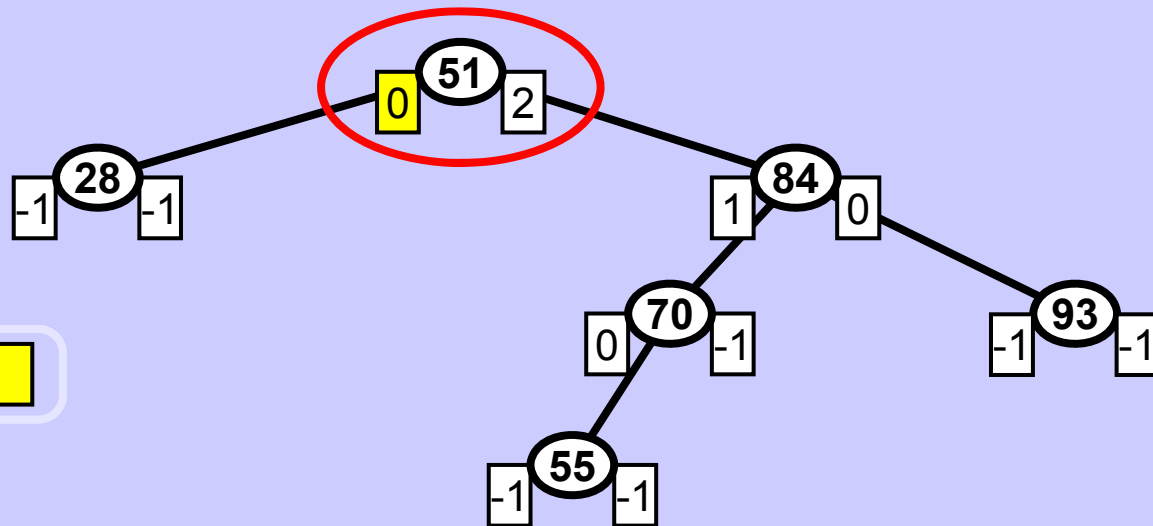
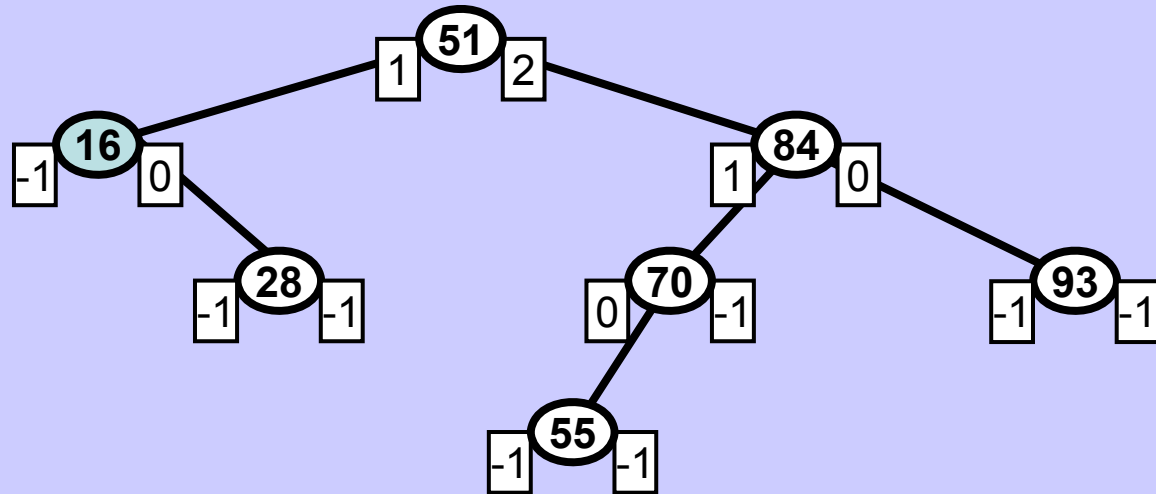
Delete 16

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.

## Delete v AVL stromu

Delete 16

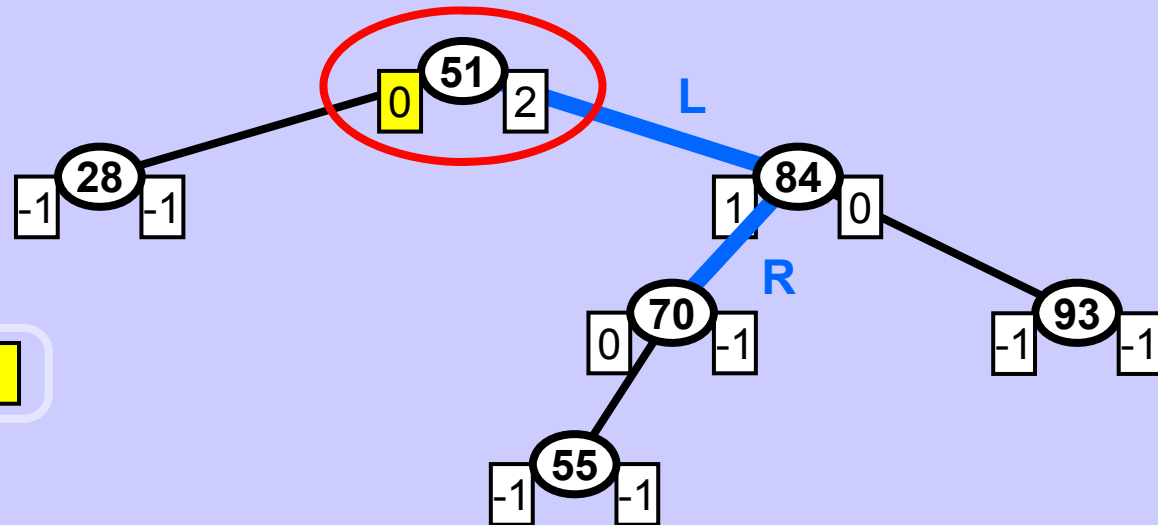


Změněné hloubky





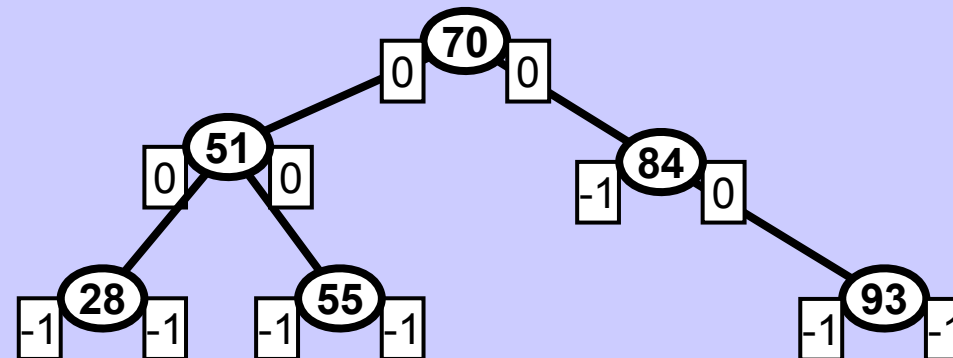
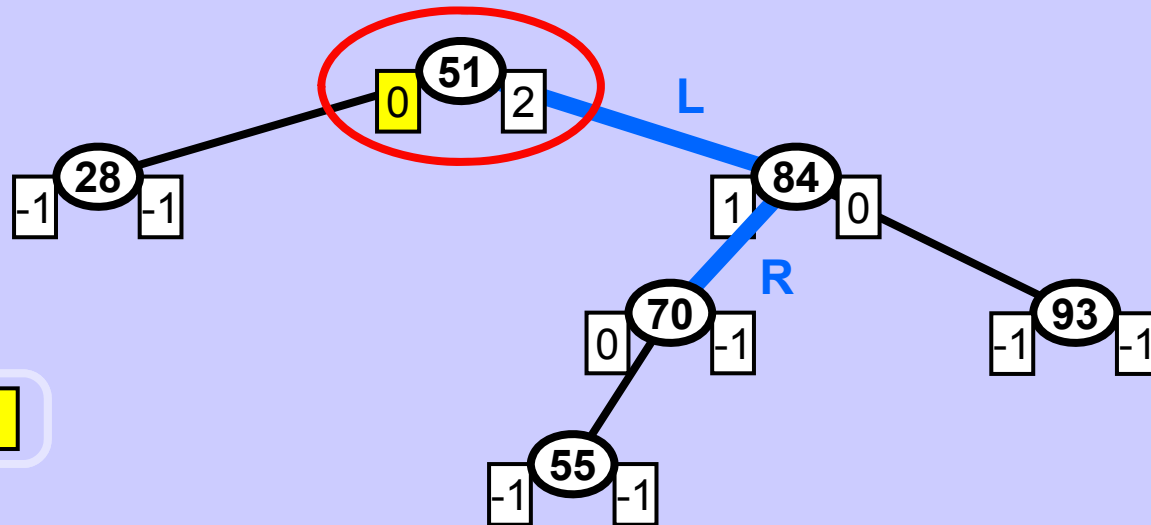
## Delete v AVL stromu

Změněné hloubky 

Z rozváženého uzlu 51 prozkoumáme kořen sousedního podstromu, než ze kterého jsme přišli, v tomto případě uzel 84. Má-li tento oba své podstromy stejně hluboké použijeme jednoduchou L nebo R rotaci. Má-li je různě hluboké (nejvýše se liší o 1), rozhodneme, zda použijeme L, R, LR, RL rotaci, jako kdyby rozvážení (uzel 51) vzniklo naopak přidáním uzlu do tohoto sousedního podstromu (s kořenem 84). V tomto případě použijeme RL.

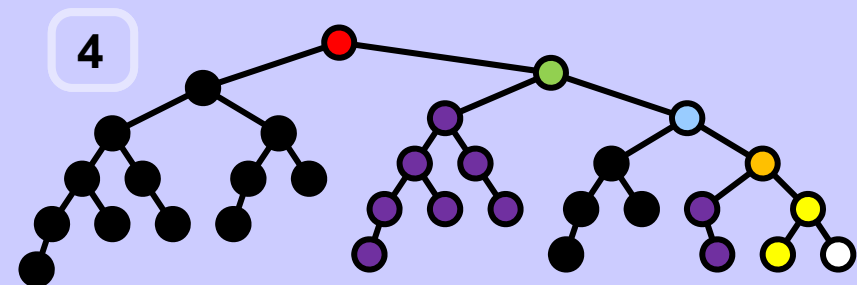
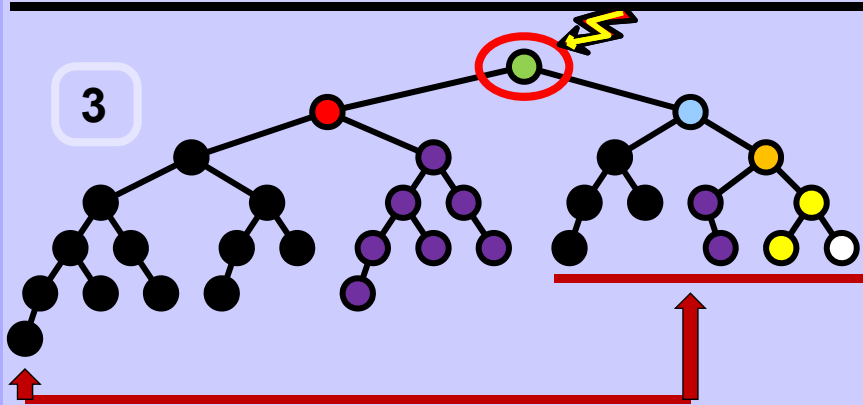
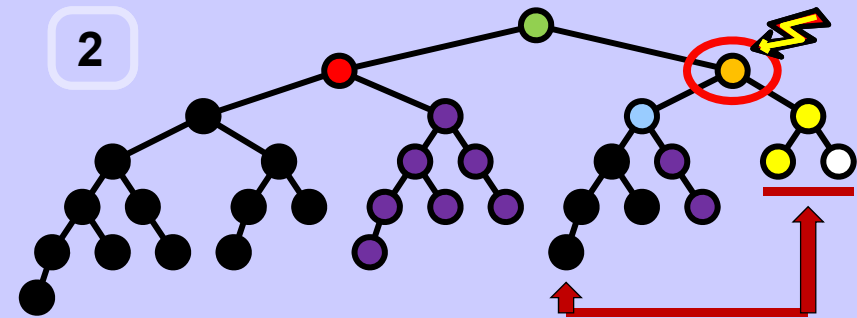
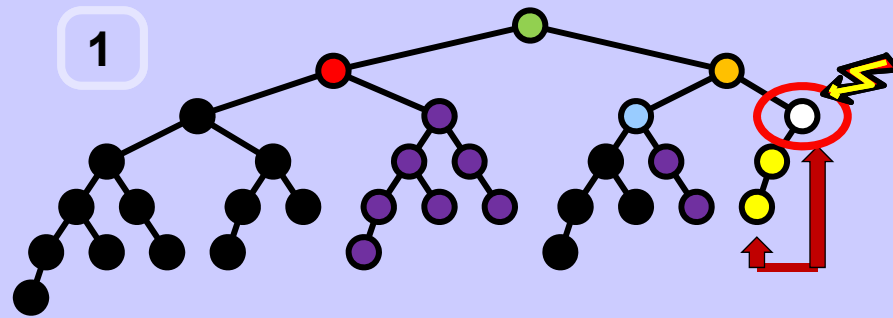
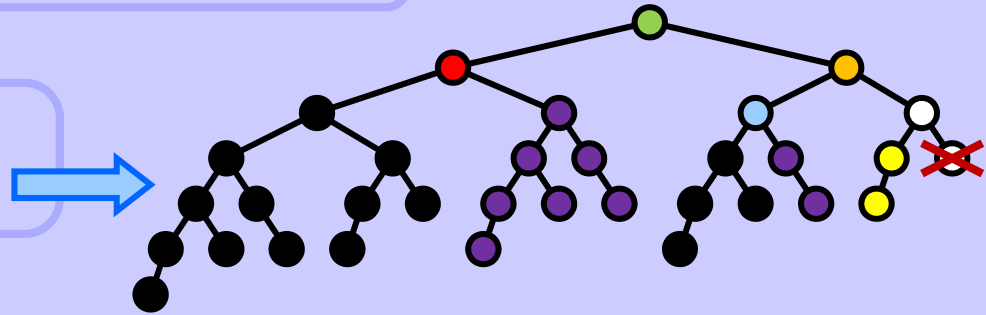
## Delete v AVL stromu

Delete 16

Změněné hloubky Po rotaci RL  
v uzlu 84 a 51

## Nutnost vícenásobných rotací po operaci Delete.

Příklad. Před operací Delete je AVL strom vyvážený.



Vyváženo.

## Implementace operací v AVL stromu

...

// homework...

## Asymptotické složitosti operací Find, Insert, Delete v BVS a AVL

Operace	BVS s n uzly		AVL strom s n uzly
	Vyvážený	Možná nevyvážený	Vyvážený
<b>Find</b>	$O(\log(n))$	$O(n)$	$O(\log(n))$
<b>Insert</b>	$\Theta(\log(n))$	$O(n)$	$\Theta(\log(n))$
<b>Delete</b>	$\Theta(\log(n))$	$O(n)$	$\Theta(\log(n))$

## B-strom -- Rudolf Bayer, Edward M. McCreight, 1972

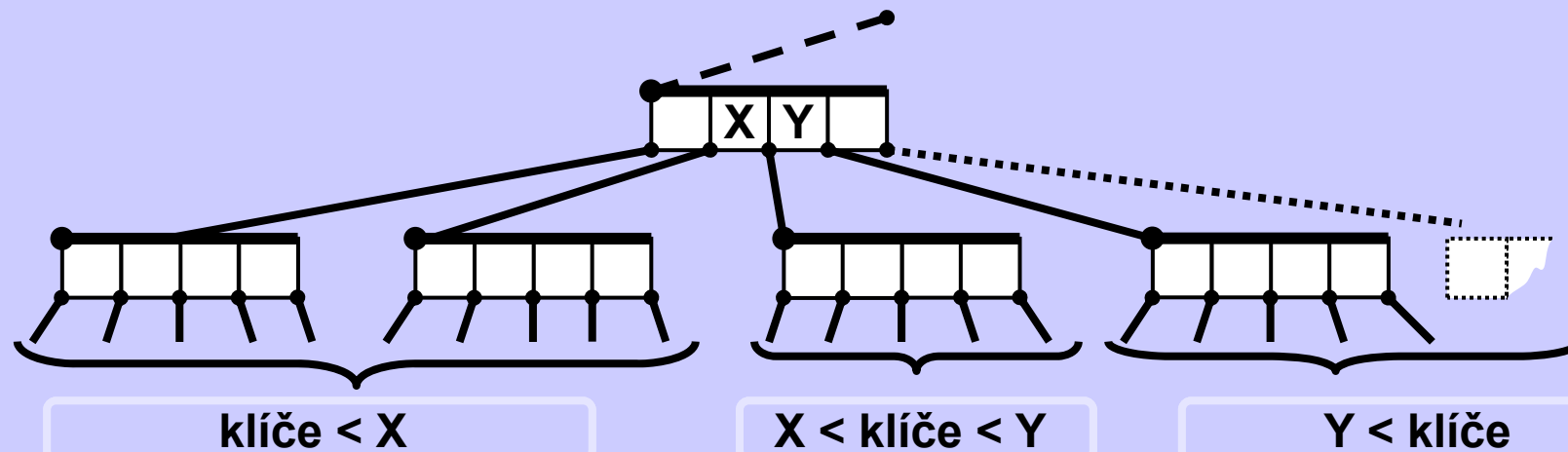
Všechny cesty z kořene do listu jsou stejně dlouhé  
tj. B-strom je ideálně vyvážený.

Klíče jsou v uzlu seřazené.

Fixní  $k > 1$  pro celý strom určuje velikost všech uzlů.

Každý uzel kromě kořene má nejméně  $k$  a nejvýše  $2k$  klíčů,  
každý vnitřní uzel tedy má nejméně  $k+1$  a nejvýše  $2k+1$  potomků.

Kořen může mít libovolný počet klíčů od 1 do  $k$ . Není-li zároveň listem,  
má alespoň 2 potomky.

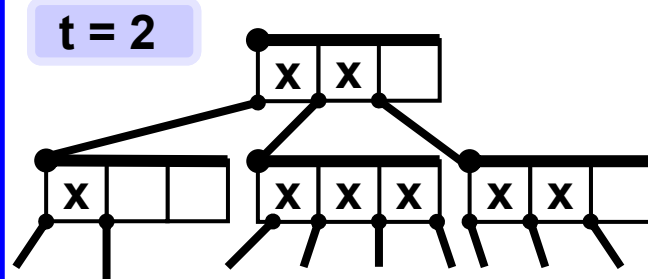


## B-strom -- alternativní specifikace

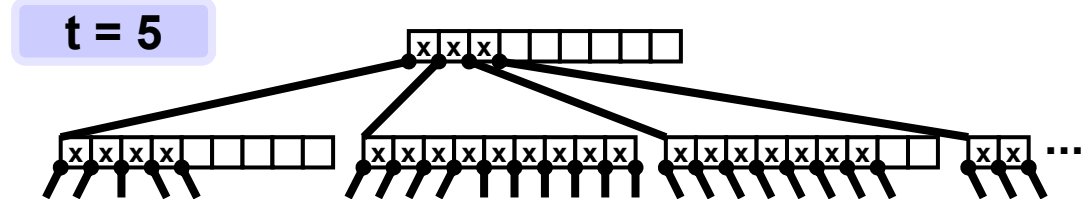
Cormen et al. 1990: Stupeň B-stromu:

Horní a dolní hranice počtu klíčů v uzlu je dána parametrem  $t \geq 2$ :

- Každý uzel různý od kořene obsahuje alespoň  $t-1$  klíčů.  
Tudíž každý vnitřní uzel různý od kořene má alespoň  $t$  potomků.
- Každý uzel obsahuje nejvýše  $2t-1$  klíčů.  
Tudíž každý vnitřní uzel má nejvýše  $2t$  potomků.



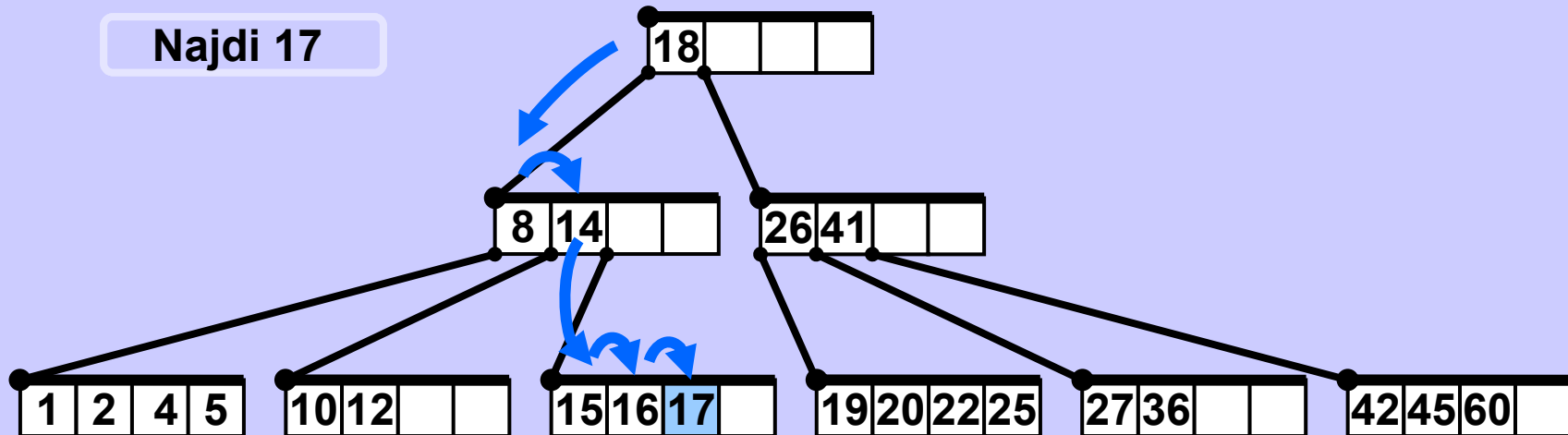
min klíčů = 1  
min potomků = 2  
max klíčů = 3  
max potomků = 4



min klíčů = 4  
min potomků = 5  
max klíčů = 9  
max potomků = 10

## B-strom -- Find

Najdi 17



V uzlu se vyhledává sekvenčně (lze také využít půlení intervalu apod).  
 Pokud uzel není listem a klíč v něm není,  
 hledání pokračuje v odpovídajícím potomku.  
 Pokud uzel je listem a klíč v něm není, nenalezeno.

## B-strom -- aktualizací strategie

### Vícefázová a jednofázová strategie

**1. Vícefázová strategie: “Řeš problém, až když vznikne”.**

Nejprve vlož nebo smaž aktuální klíč a teprve potom, pokud je to nutné, uprav strukturu stromu.

Může se stát, že pak bude nutno během úprav procházet strom zpět až ke kořeni.

**2. Jednofázová strategie: “Předcházej budoucím problémům”.**

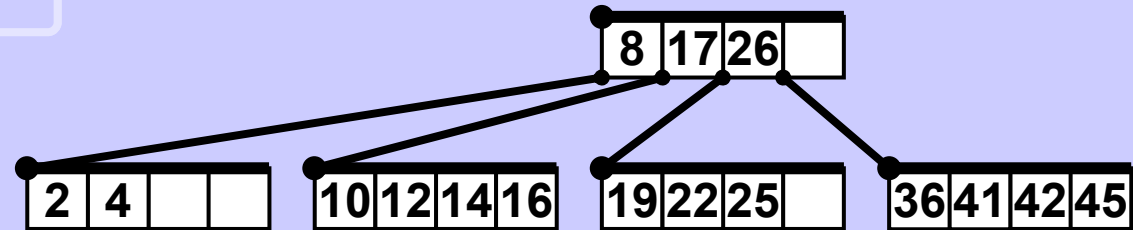
Postupuj od kořene na místo, kde má být klíč/uzel vložen/smazán a během postupu upravuj strukturu stromu tak, aby po vložení/smazání nebylo nutno vracet se směrem ke kořeni.



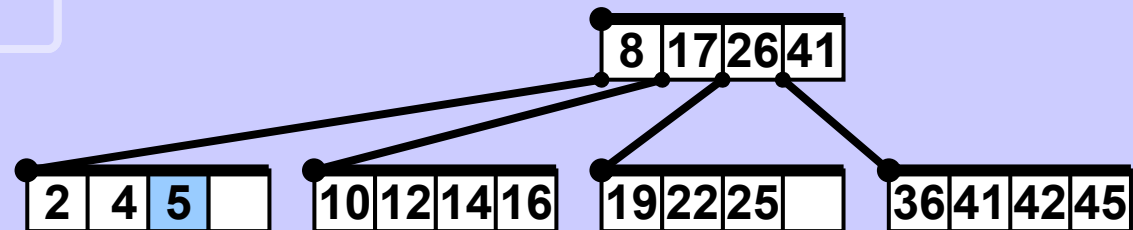
## B-strom -- Insert

## Vícefázová strategie

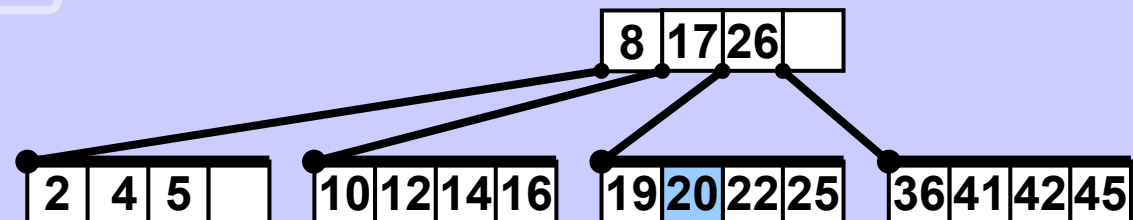
B-strom



Vlož 5



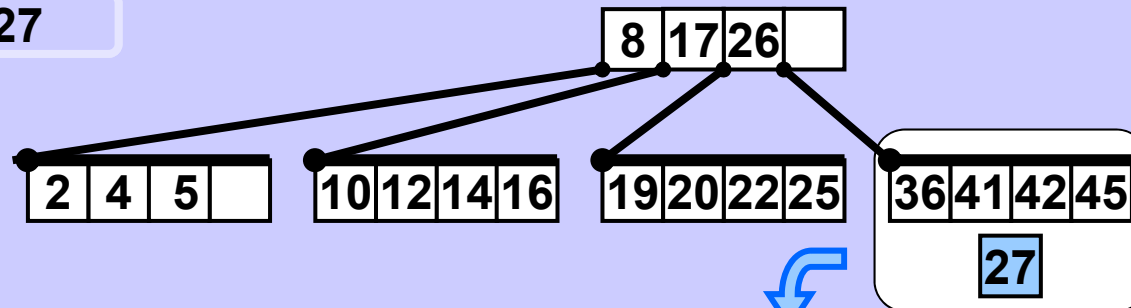
Vlož 20



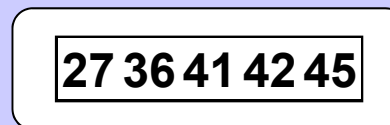
## B-strom -- Insert

## Vícefázová strategie

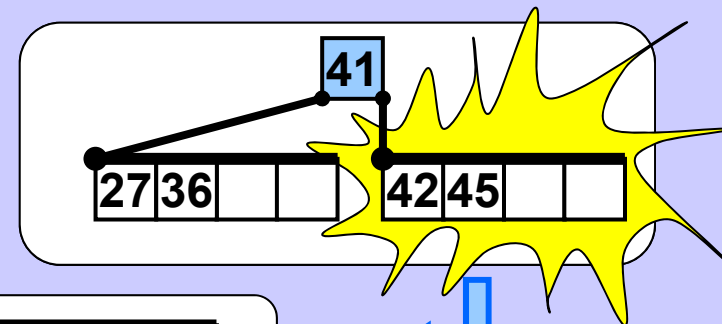
Vlož 27



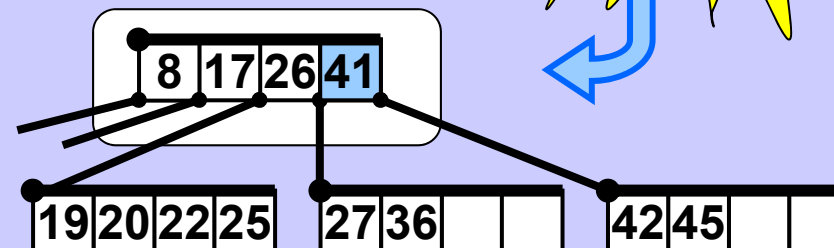
Seřad' mimo strom.



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



Medián zkus vložit  
do rodiče.

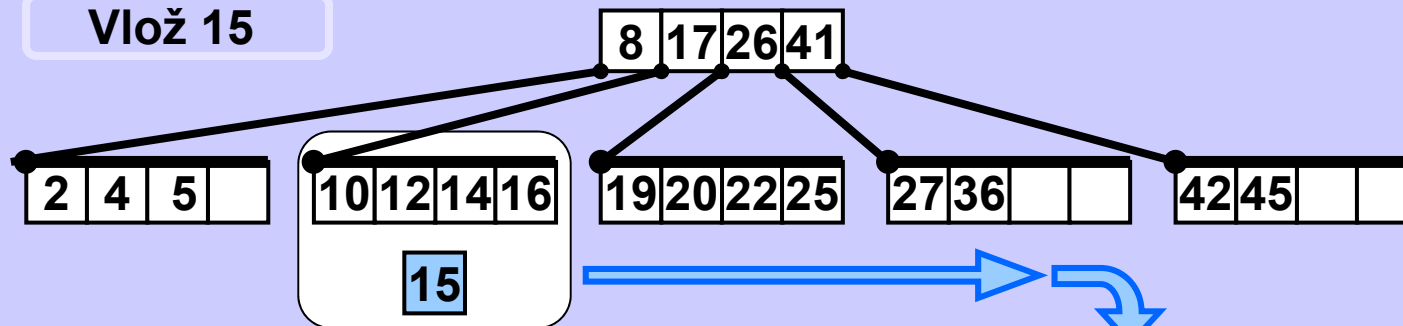


Zdařilo se.

## B-strom -- Insert

## Vícefázová strategie

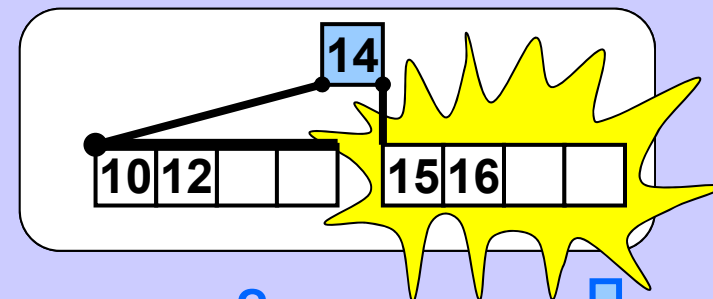
Vlož 15



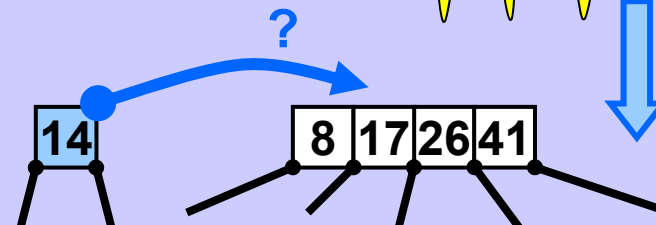
Seřad' mimo strom.

10 12 14 15 16

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



Medián zkus vložit  
do rodiče.

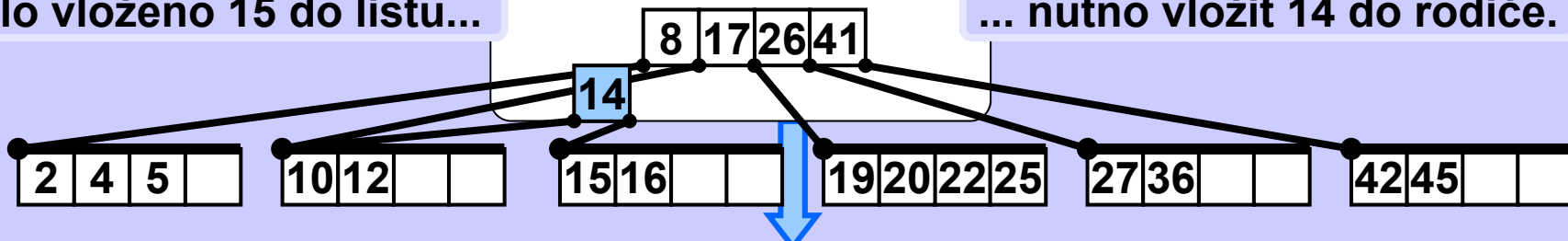


## B-strom -- Insert

## Vícefázová strategie

Bylo vloženo 15 do listu...

... nutno vložit 14 do rodiče.

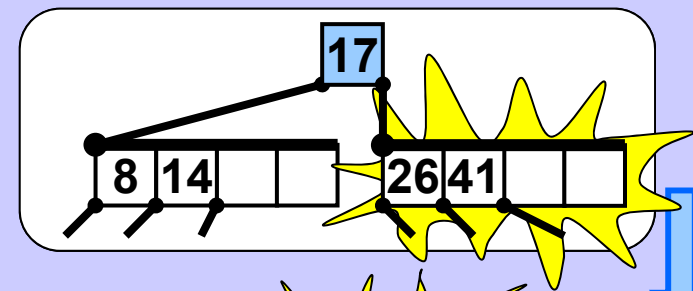


Rodič je zaplněn – Analogický další postup směrem ke kořeni.

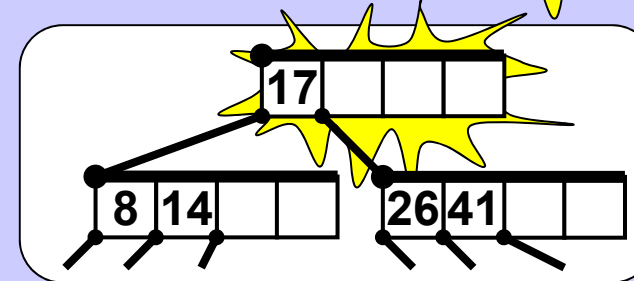
Seřad' mimo strom.

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.

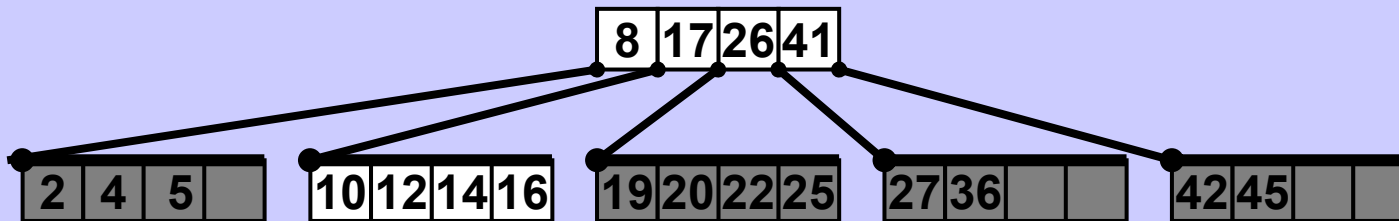


## B-strom -- Insert

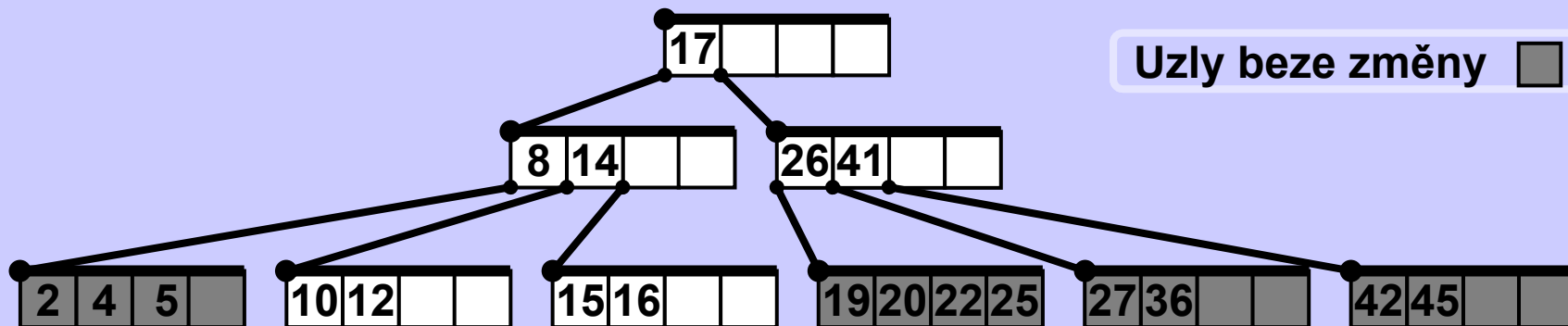
## Vícefázová strategie



Rekapitulace - vlož 15



Vlož 15



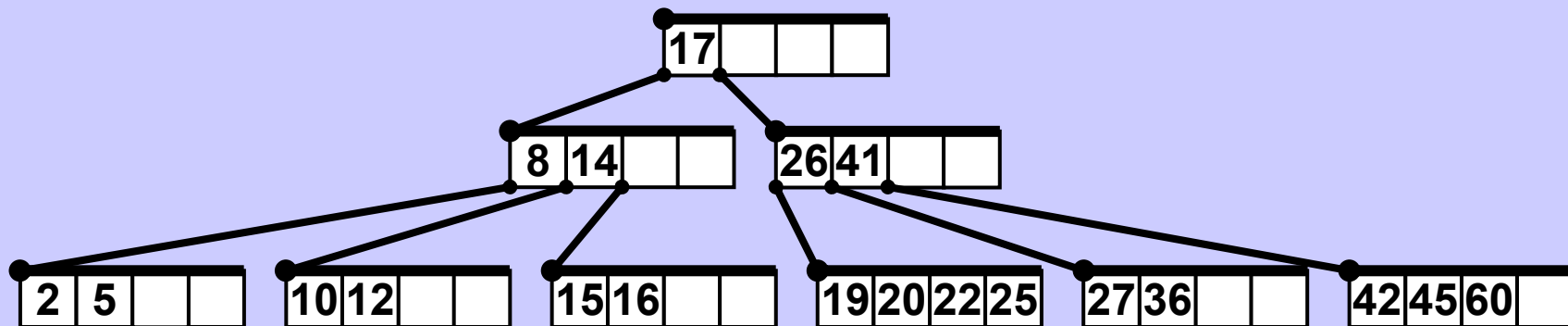
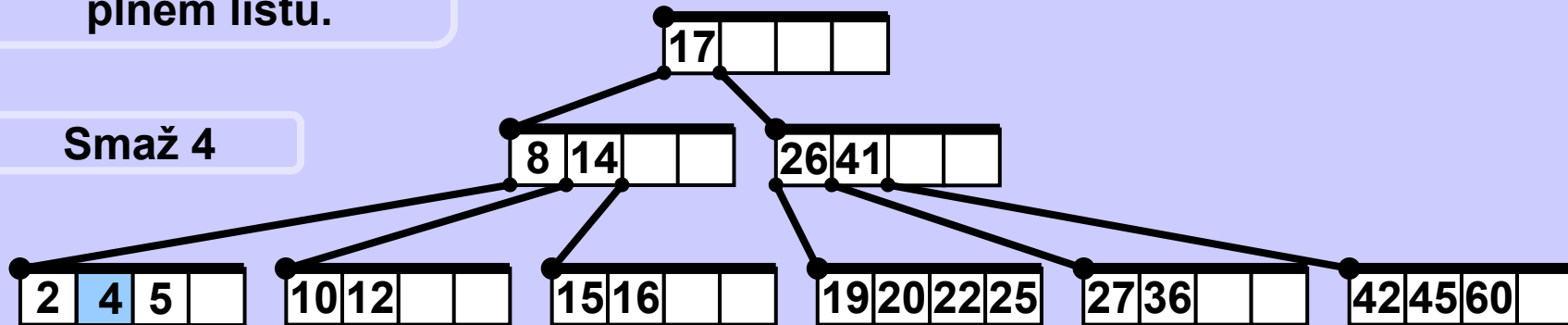
V každém patře přibyl jeden uzel, kromě toho přibyl nový kořen, strom ale roste směrem "vzhůru", zůstává ideálně vyvážený.

## B-strom -- Delete

## Vícefázová strategie

Mazání v dostatečně  
plném listu.

Smaž 4

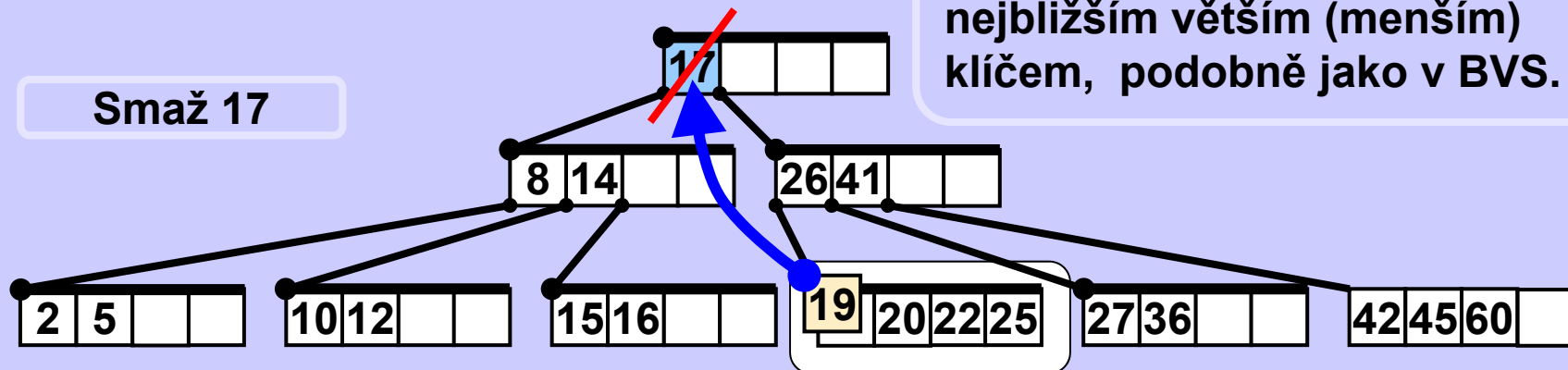


## B-strom -- Delete

## Vícefázová strategie

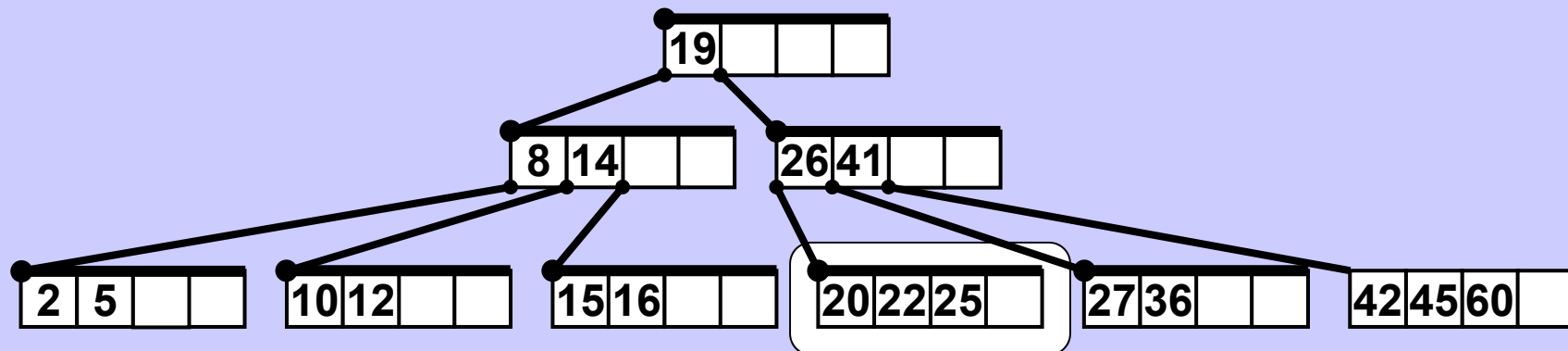
## Mazání ve vnitřním uzlu

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.



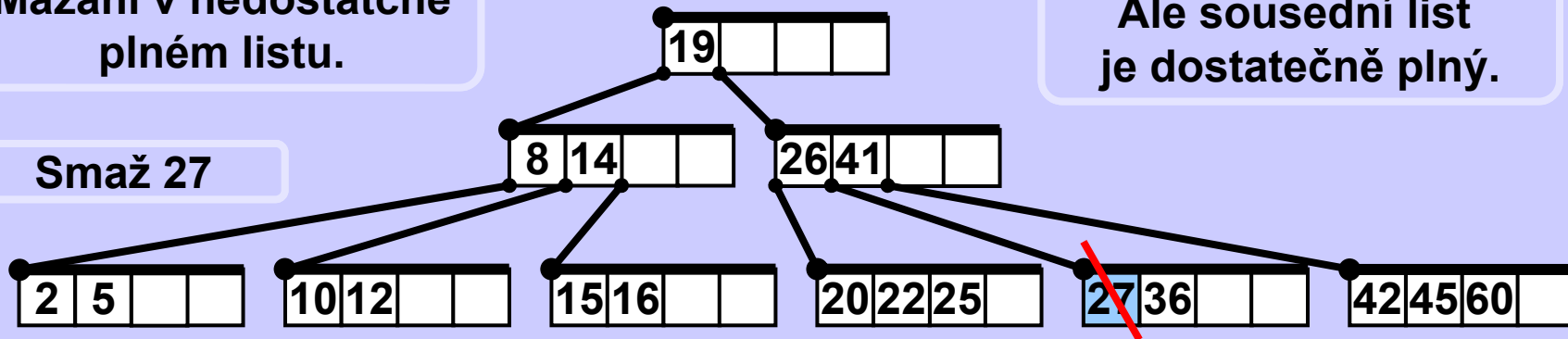
## B-strom -- Delete

## Vícefázová strategie

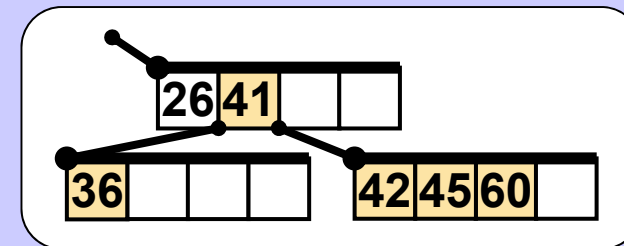
Mazání v nedostatečně plném listu.

Ale sousední list je dostatečně plný.

Smaž 27

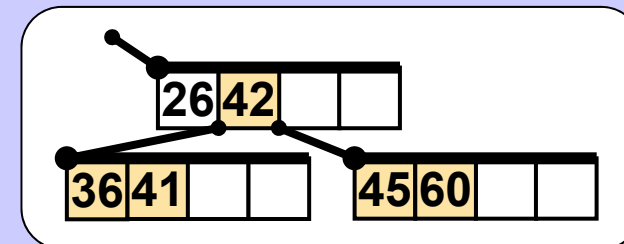


Sjednot' klíče s klíči v sousedním listu a s dělicím klíčem v rodiči a seřad'.



36 41 42 45 60

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

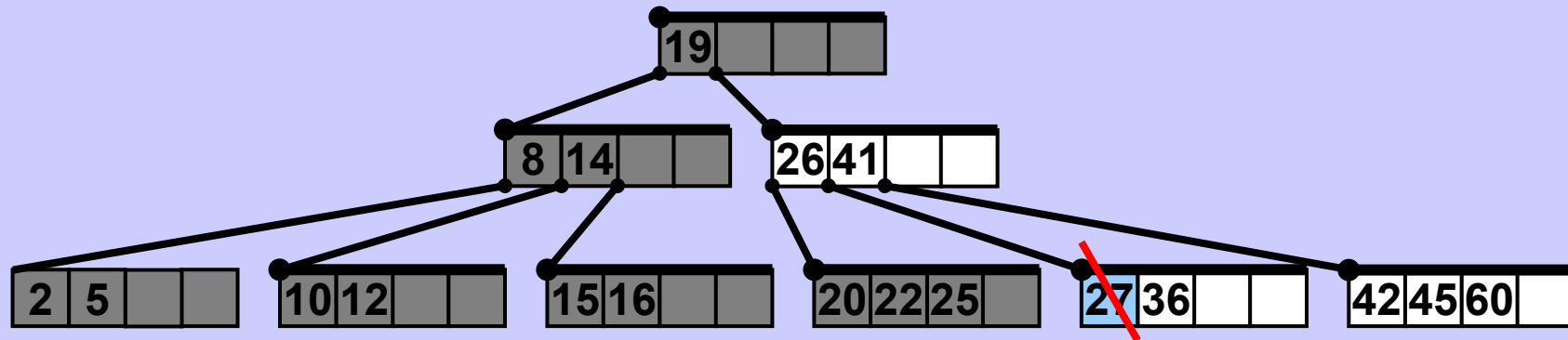




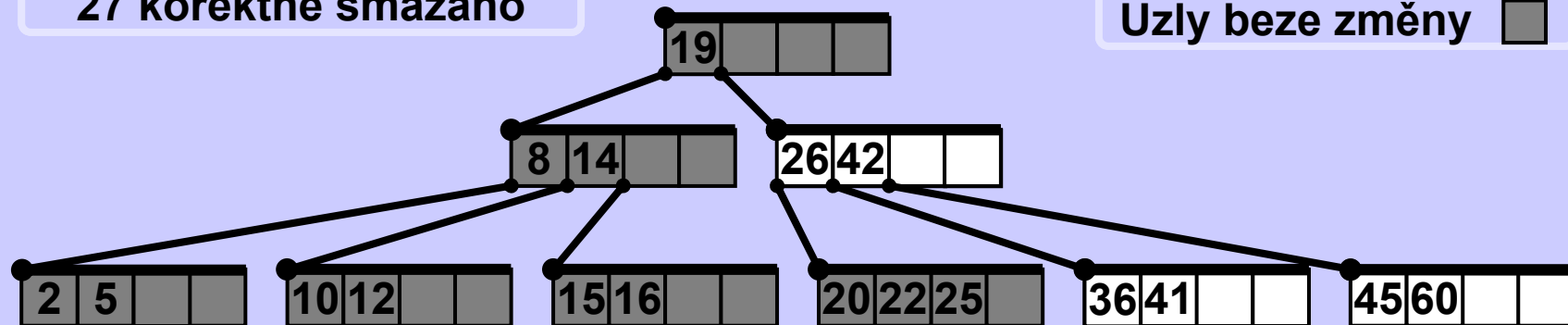
## B-strom -- Delete

## Vícefázová strategie

Rekapitulace - smaž 27



27 korektně smazáno

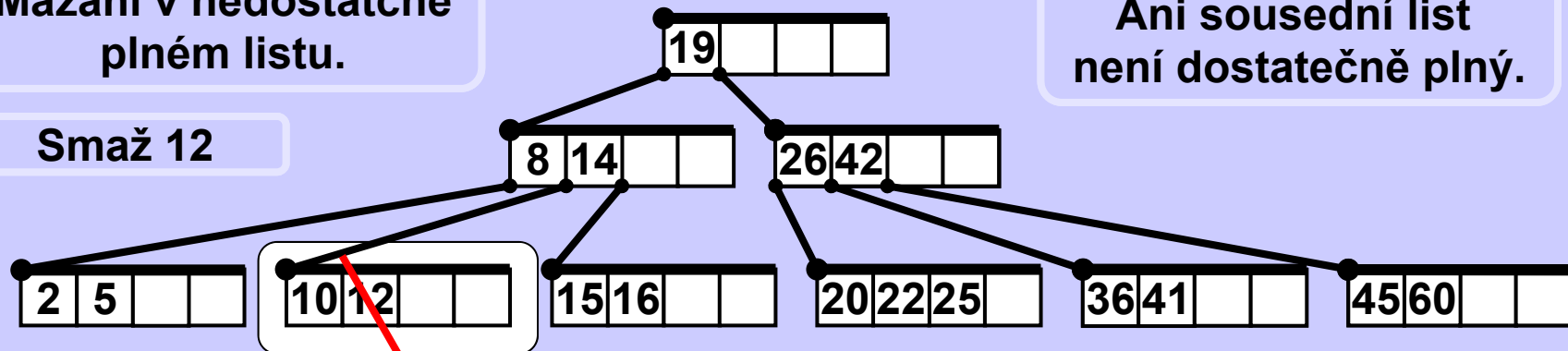


## B-strom -- Delete

## Vícefázová strategie

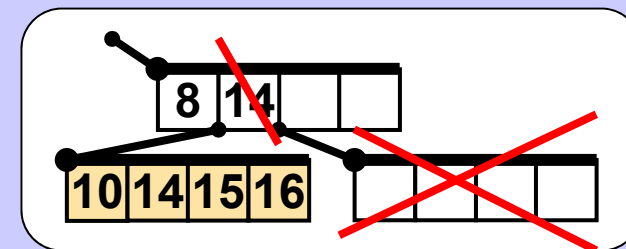
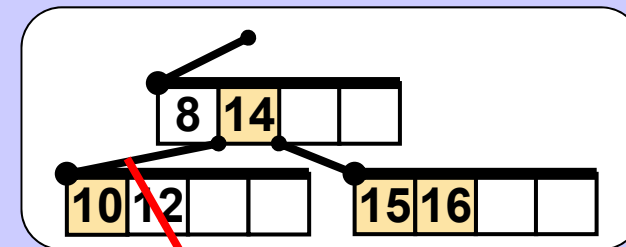
Mazání v nedostatečně plném listu.

Smaž 12



Ani sousední list není dostatečně plný.

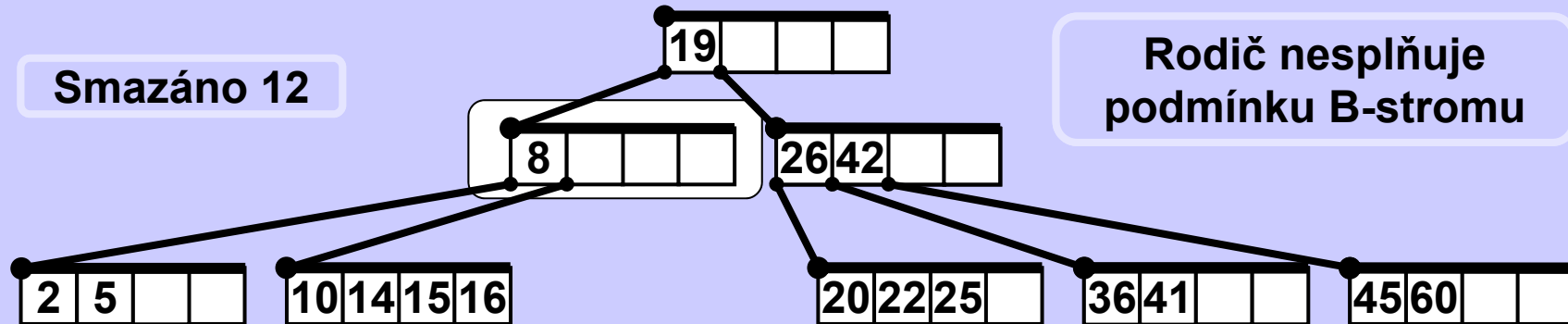
Sjednot' klíče s klíči v sousedním listu a s dělicím klíčem v rodiči a seřad'. Vše vlož do původního listu, sousední list smaž, dělicí klíč v rodiči také smaž.



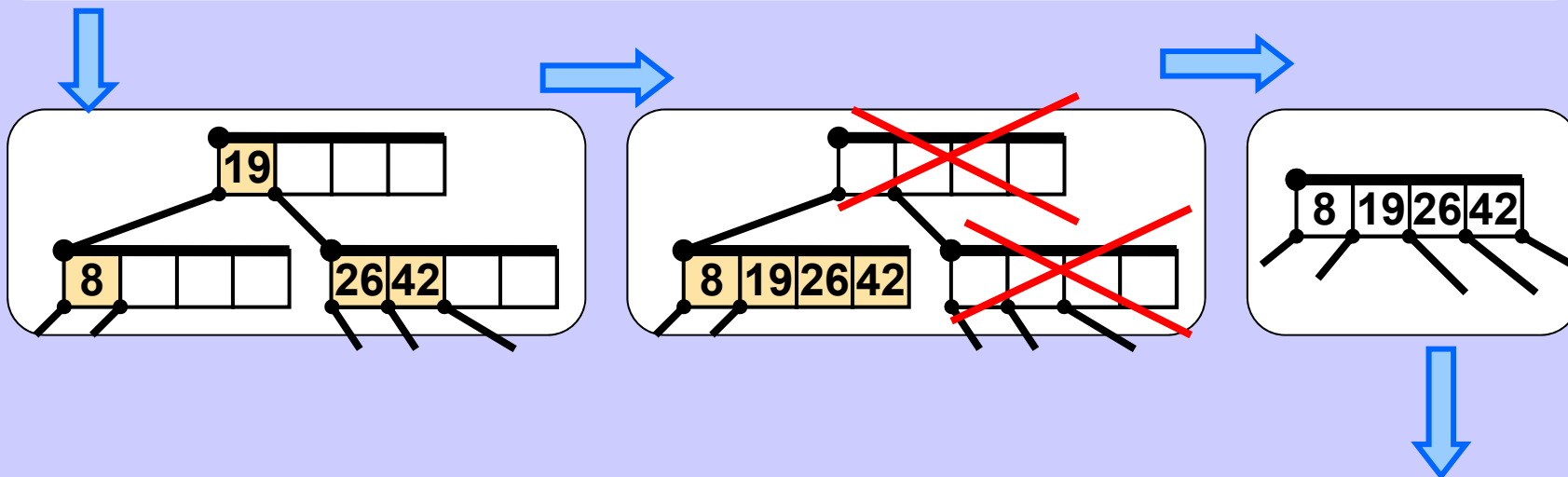
## B-strom -- Delete

## Vícefázová strategie

Smazáno 12



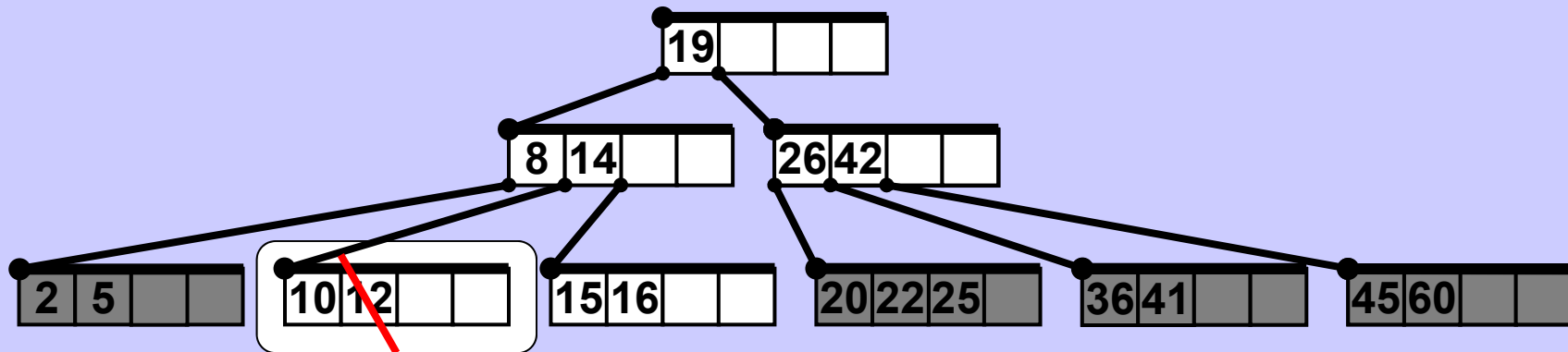
Rodič, který poskytl klíč potomku, není dostatečně plný.  
 Aplikujeme na něj (a případně iterativně na jeho rodiče) tentýž postup  
 spojení klíčů a sousedních uzlů a přesun dělicího prvku z rodiče.



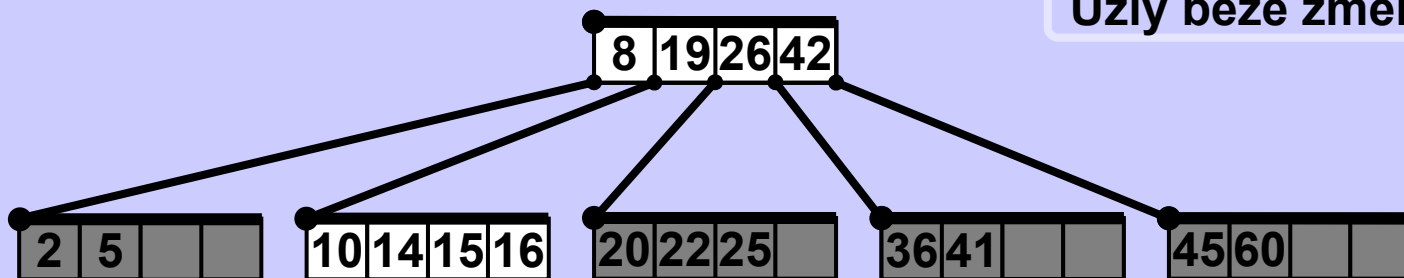
## B-strom -- Delete

## Vícefázová strategie

Rekapitulace - smaž 12



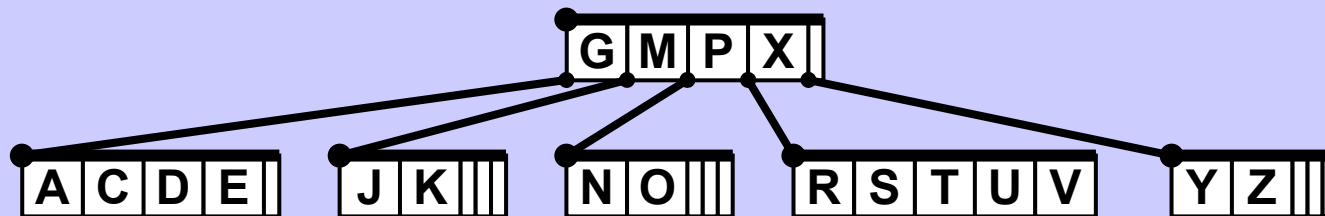
Smazáno 12 a strom byl adekvátně restrukturován.

Uzly beze změny

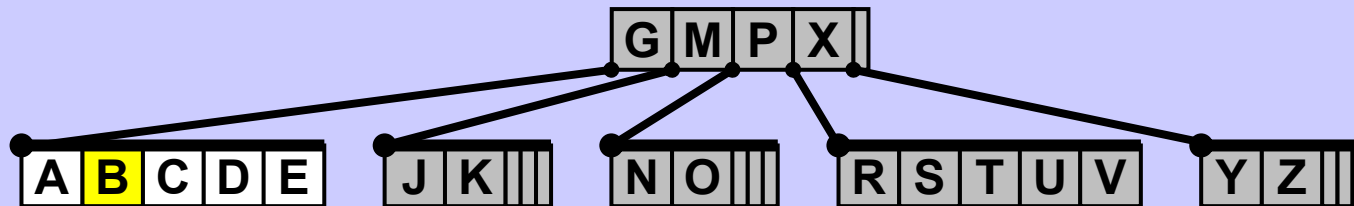
## B-strom -- Insert

## Jednofázová strategie

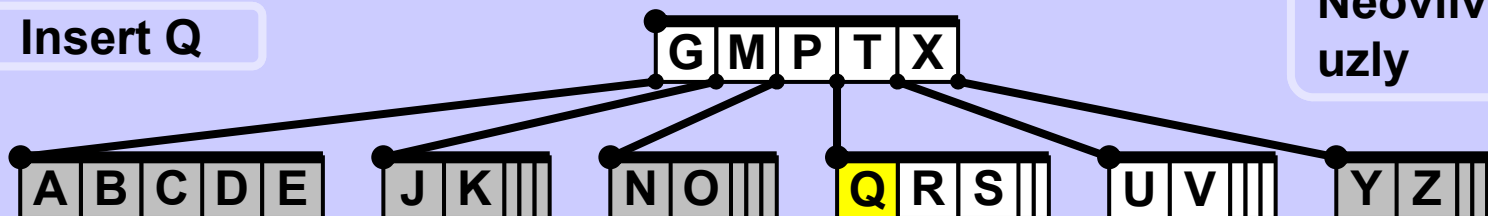
Příklad: Cormen et al. 1990:  $t = 3$ , min stupeň 3, max stupeň = 6,  
min klíčů v uzlu (kromě kořene) = 2, max klíčů v uzlu = 5.



Insert B



Insert Q



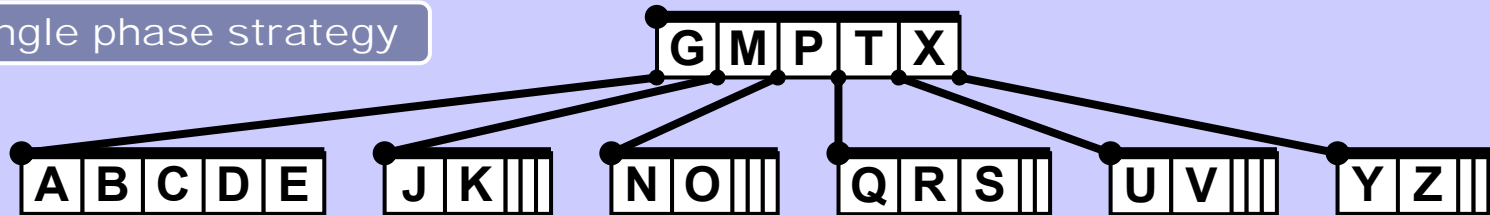
Neovlivněné  
uzly



## B-strom -- Insert

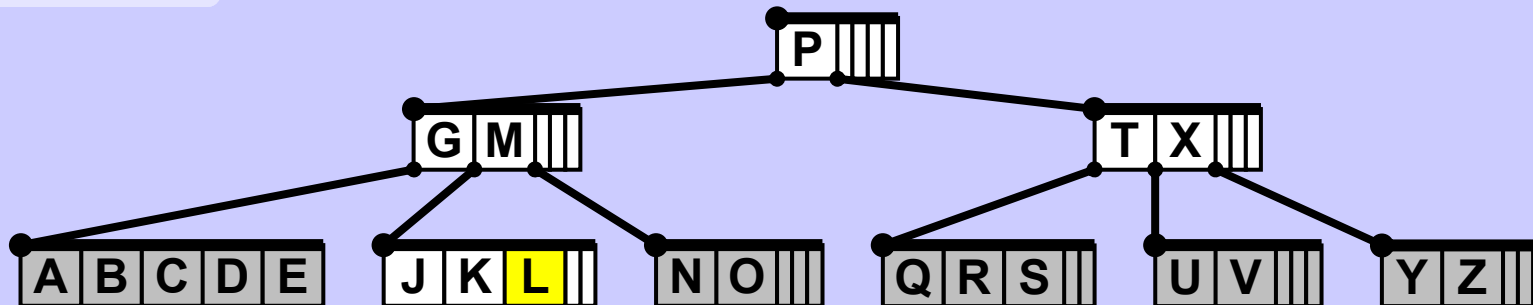
## Jednofázová strategie

Single phase strategy

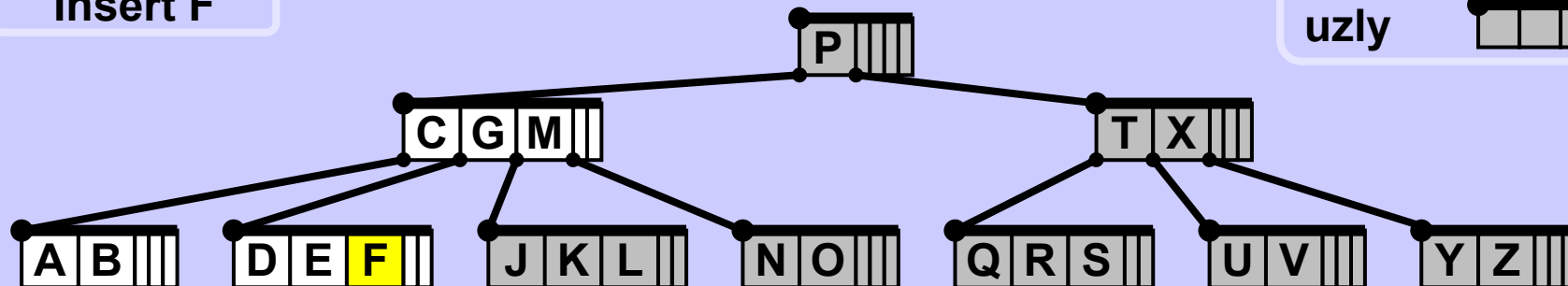


Rozděl kořen, protože je plný,  
a pokračuj směrem k listu vložit L.

Insert L



Insert F

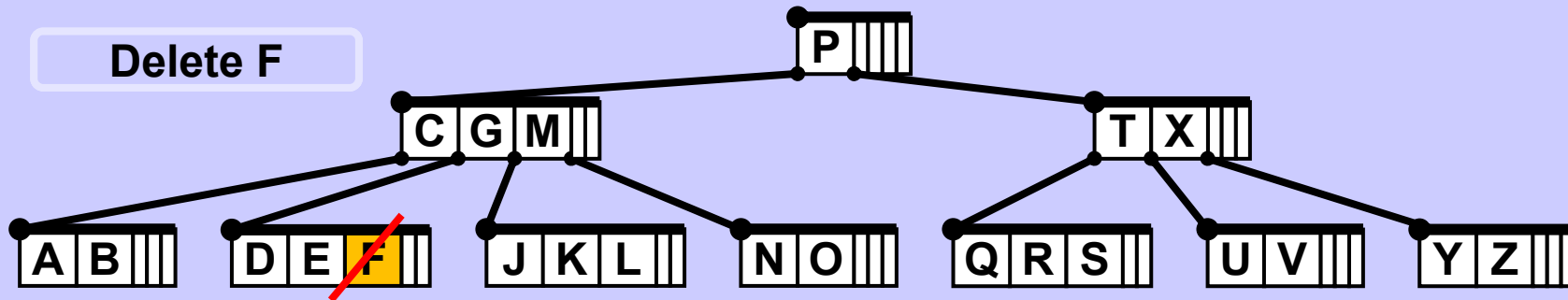


Neovlivněné  
uzly

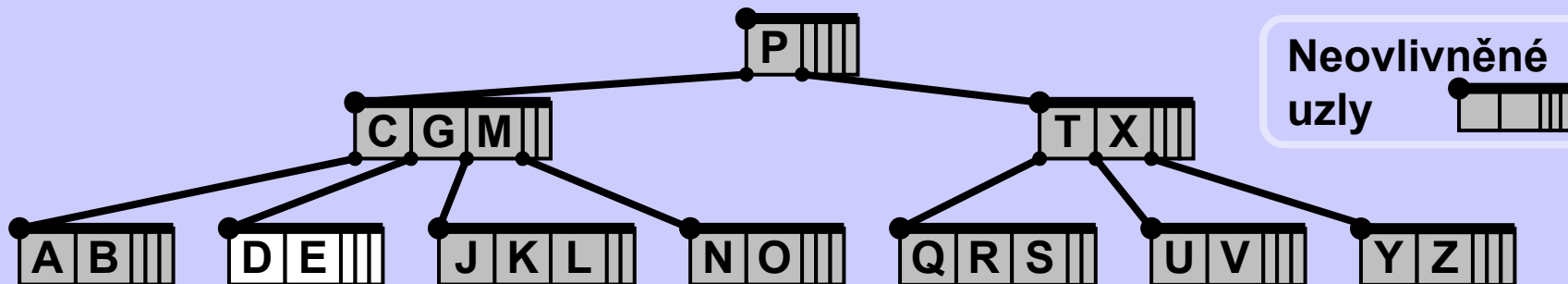
## B-strom -- Delete

## Jednofázová strategie

Delete F



1. Když je klíč  $k$  v listu  $X$ , smaž  $k$  v  $X$ .

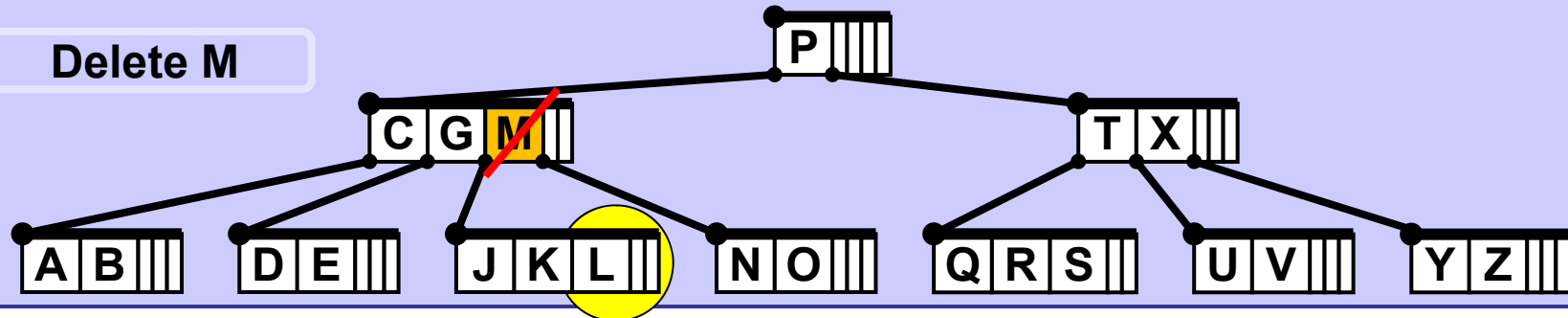


Další případy 2. a 3. zaručují, že smazání vždy proběhne v uzlu s alespoň  $t$  klíči, takže po smazání klíče  $k$  nebude nutno list spojovat se sousedy a opravovat strom ve vnitřních uzlech směrem zpět ke kořeni.

## B-strom -- Delete

## Jednofázová strategie

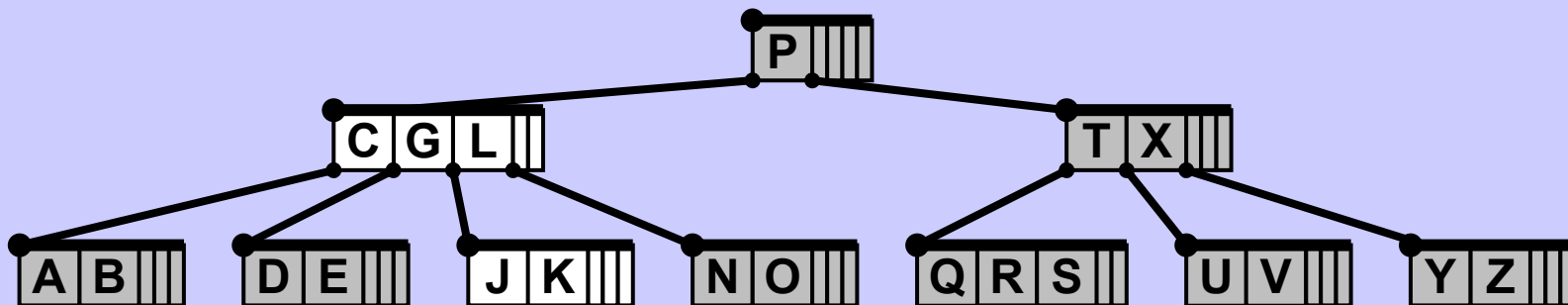
Delete M



2. Když je klíč  $k$  ve vnitřním uzlu  $X$ , udělej:

2a. Když má potomek  $Y$  vlevo od klíče  $k$  v uzlu  $X$  alespoň  $t$  klíčů, najdi maximální klíč  $k_p$  v podstromu s kořenem  $Y$ . Nahradi v  $X$  klíč  $k$  klíčem  $k_p$  and rekurzivně pokračuj mazat  $k_p$  v podstromu s kořenem  $Y$ .

(b. Pokud má  $Y$  méně než  $t$  klíčů, pak zkoumej potomka  $Z$  vpravo od klíče  $k$  v  $X$  a pokud  $Z$  má alespoň  $t$  klíčů, postupuj analogicky (s hledáním minima) jako ve 2a.

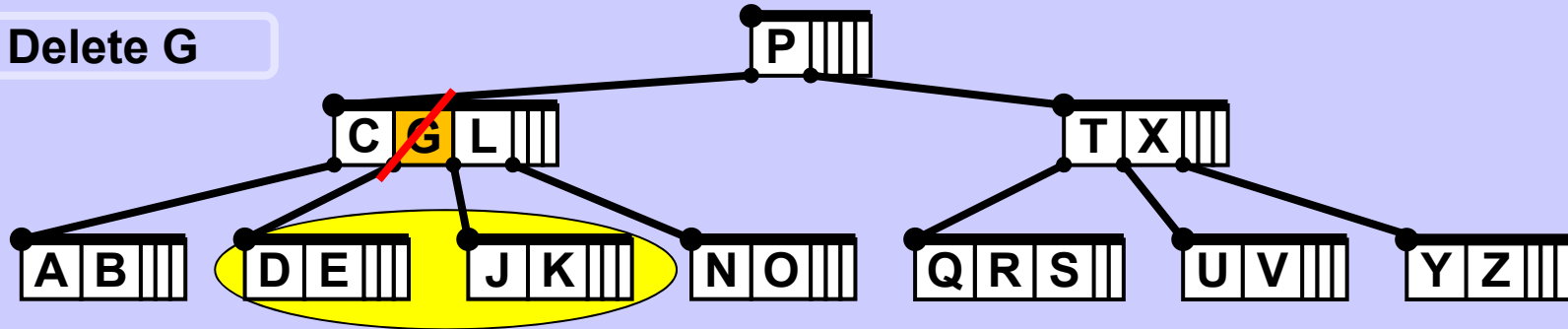




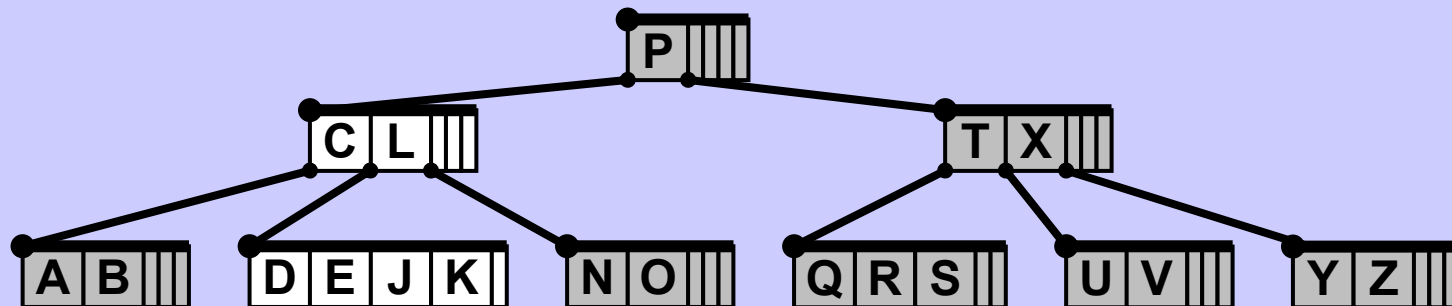
## B-strom -- Delete

## Jednofázová strategie

Delete G



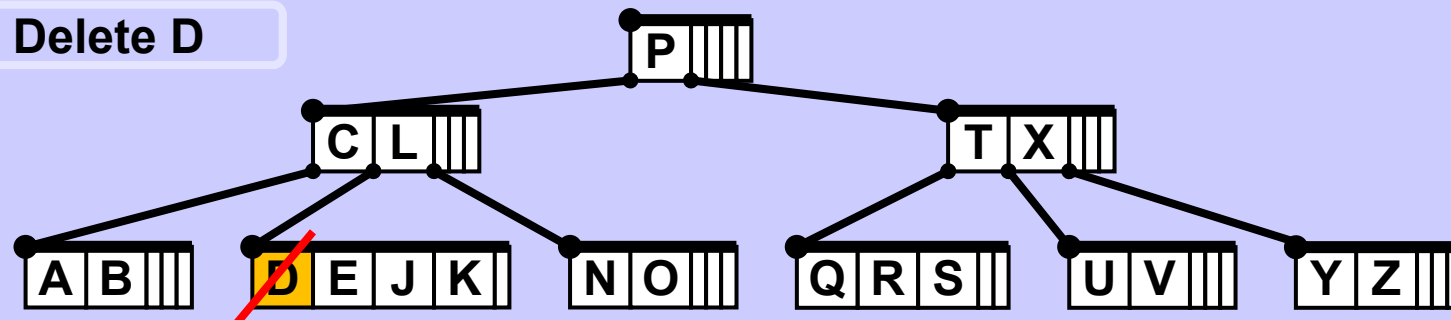
2c. Jinak, t.j. když **Y** a **Z** mají jen  $t-1$  klíčů, přesuň  $k$  a klíče uzlu **Z** do uzlu **Y**. Tím v **X** ubude klíč  $k$  a ukazatel na **Z**. Uzel **Y** nově obsahuje  $2t-1$  klíčů. Fyzicky smaž **Z** and pokračuj rekurzivně smazáním  $k$  v **Y**.



## B-strom -- Delete

## Jednofázová strategie

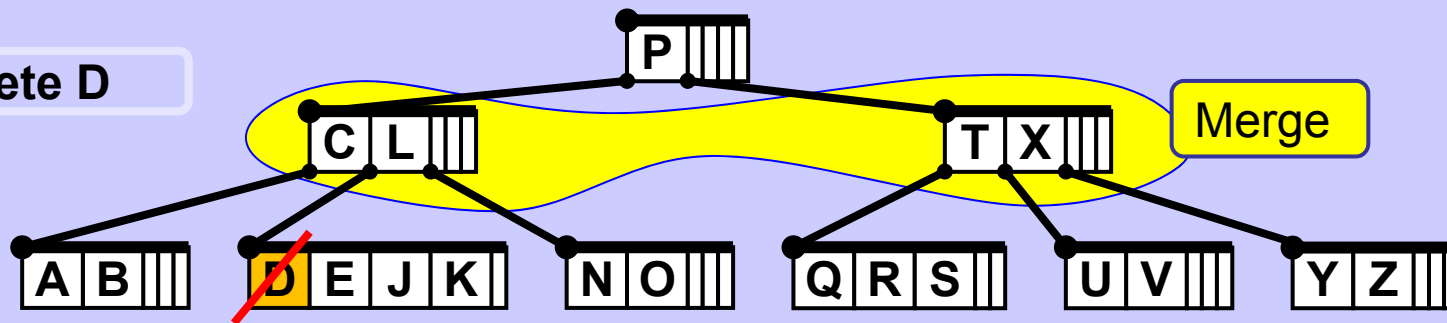
3. Když vnitřní uzel  $X$  neobsahuje klíč  $k$ , urči toho potomka  $X.c$  uzlu  $X$ , který je kořenem podstromu, v němž leží  $k$ , pokud se  $k$  ve stromu vůbec vyskytuje. Pokud má  $X.c$  pouze  $t-1$  klíčů, proved' podle okolností krok **3a** nebo **3b**, aby bylo zaručeno, že z uzlu  $X$  bude operace mazání pokračovat v takovém jeho potomku, který má alespoň  $t$  klíčů. Poté se přesuň do tohoto potomku a pokračuj rekurzivně.



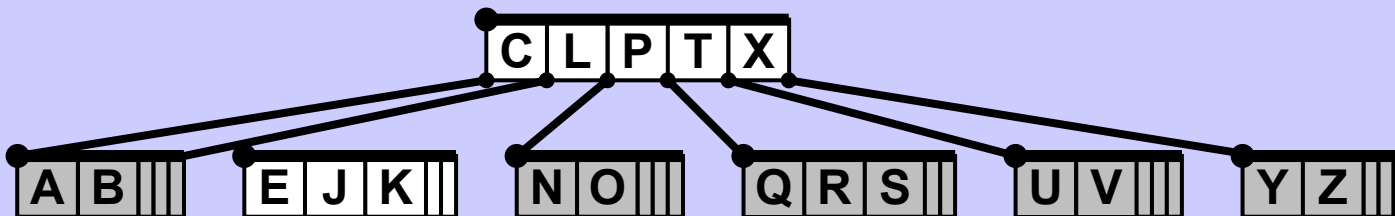
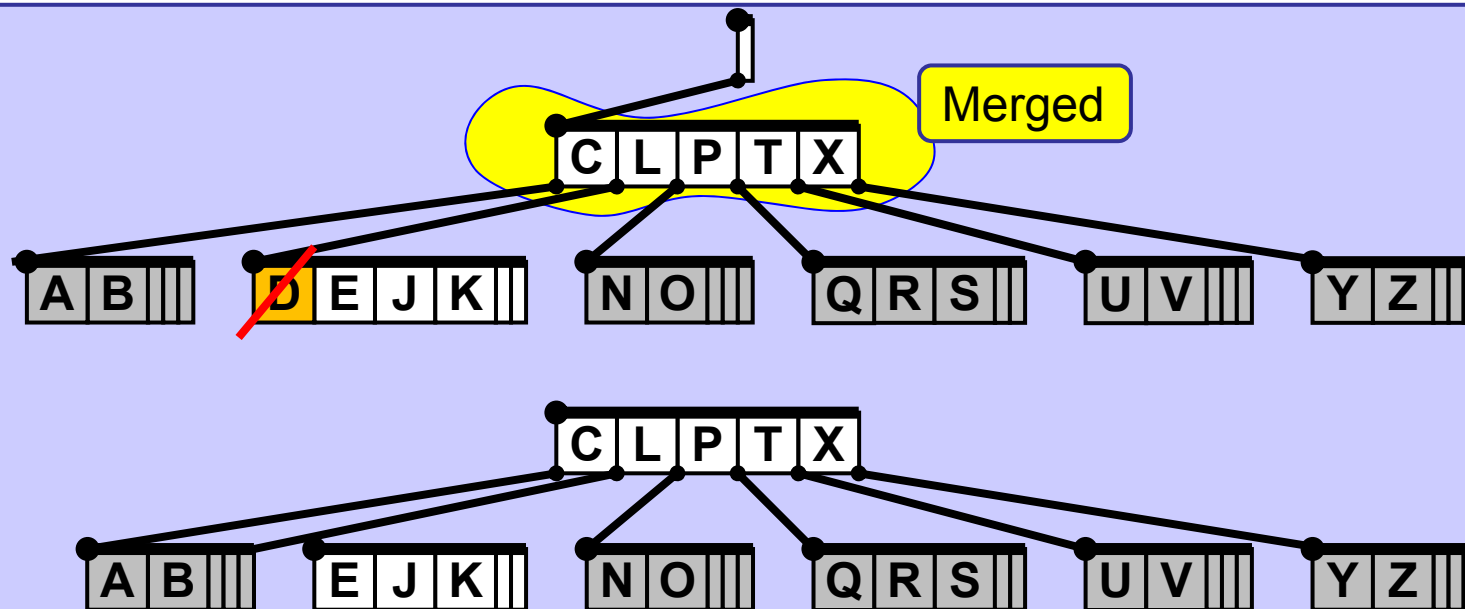
## B-strom -- Delete

## Jednofázová strategie

Delete D



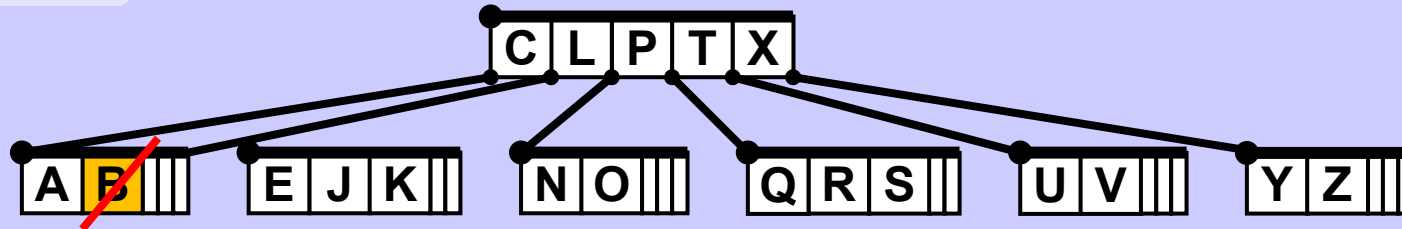
3a. Pokud uzel  $X.c$  a oba jeho sousední sourozenci (pokud existují) obsahují  $t-1$  klíčů, přesuň klíče z jednoho sourozence do  $X.c$ , smaž tohoto sourozence a navíc do  $X.c$  přesuň z  $X$  klíč, který bude nyní mediánem v  $X.c$ .



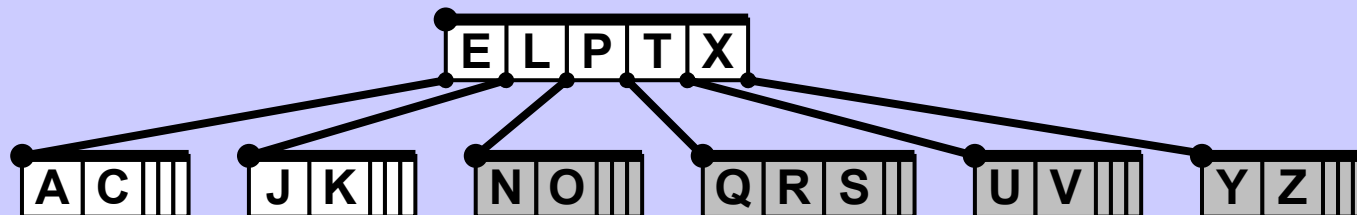
## B-strom -- Delete

## Jednofázová strategie

## Delete B



3b. Pokud uzel  $X.c$  obsahuje  $t-1$  klíčů a jeho sousední sourozenec  $Y$  obsahuje alespoň  $t$  klíčů, přesuň z  $X$  do  $X.c$  klíč  $m$  rozdělující  $X.c$  a  $Y$ . Na bývalé místo  $m$  v  $X$  přesuň patřičný krajní (největší nebo nejmenší) klíč z  $Y$ . Příslušný krajní odkaz v  $Y$  na jeho potomka přesuň do  $X.c$  ke klíči  $m$ .



## B-strom -- asymptotické složitosti operací

<b>Find</b>	$O(b \cdot \log_b n)$
<b>Insert</b>	$\Theta(b \cdot \log_b n)$
<b>Delete</b>	$\Theta(b \cdot \log_b n)$

**n** je počet klíčů ve stromu, **b** je větvící faktor, tj. řád stromu tj. maximální počet potomků uzlu.

**Poznámka:** Někteří autoři (např. CLRS) definují řád (order) B-stromu jako  $\lfloor b/2 \rfloor$ , tj. minimální počet potomků vnitřního uzlu různého od kořene. Terminologie není ustálená, je třeba vždy vztahy interpretovat v patřičných souvislostech.