

ALG 06

AVL strom

rotace L, R, LR, RL

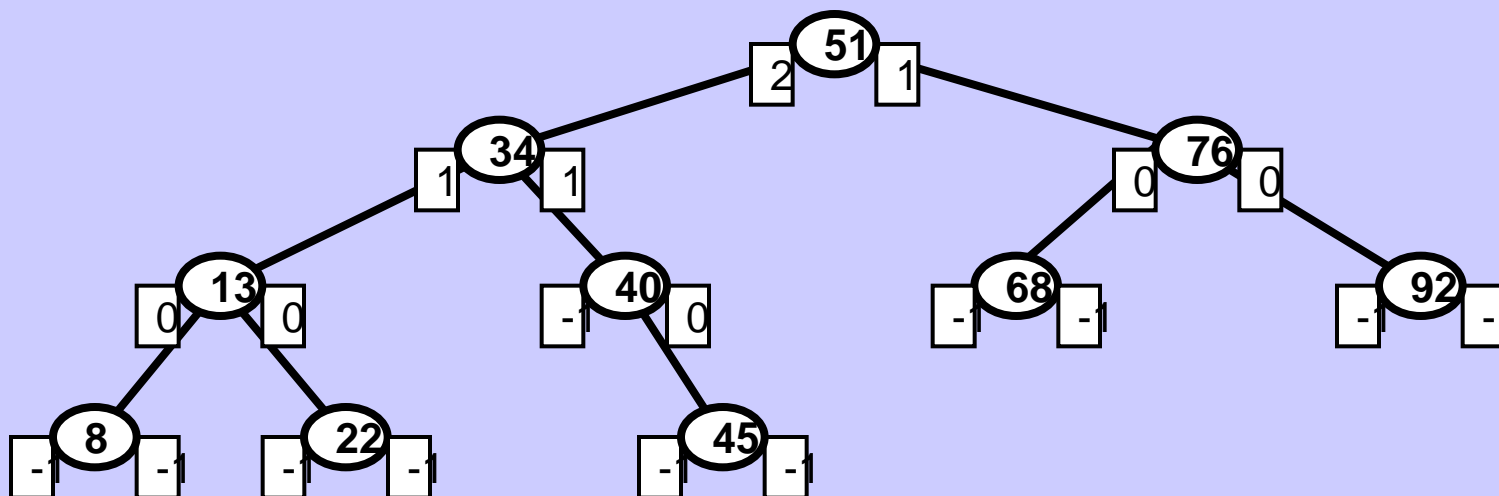
B-strom

Operace Find, Insert, Delete

AVL strom

AVL strom je BVS s přidáním vlastností, které jej udržují (téměř) vyvážený.

AVL má také operace Find, Insert, Delete.



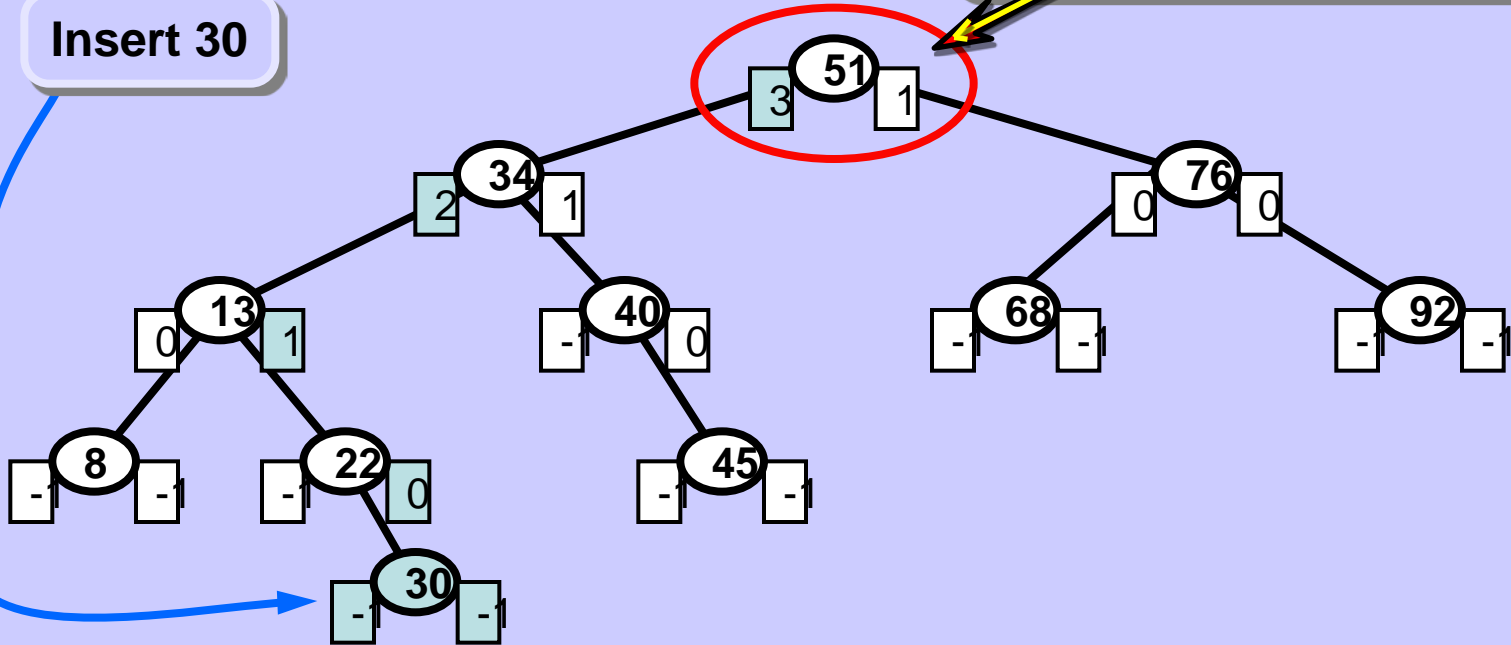
Každý uzel registruje hloubku svého levého a pravého podstromu, hloubka prázdného stromu je -1.

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

Vložení uzlu může způsobit rozvážení AVL stromu.

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

Insert 30

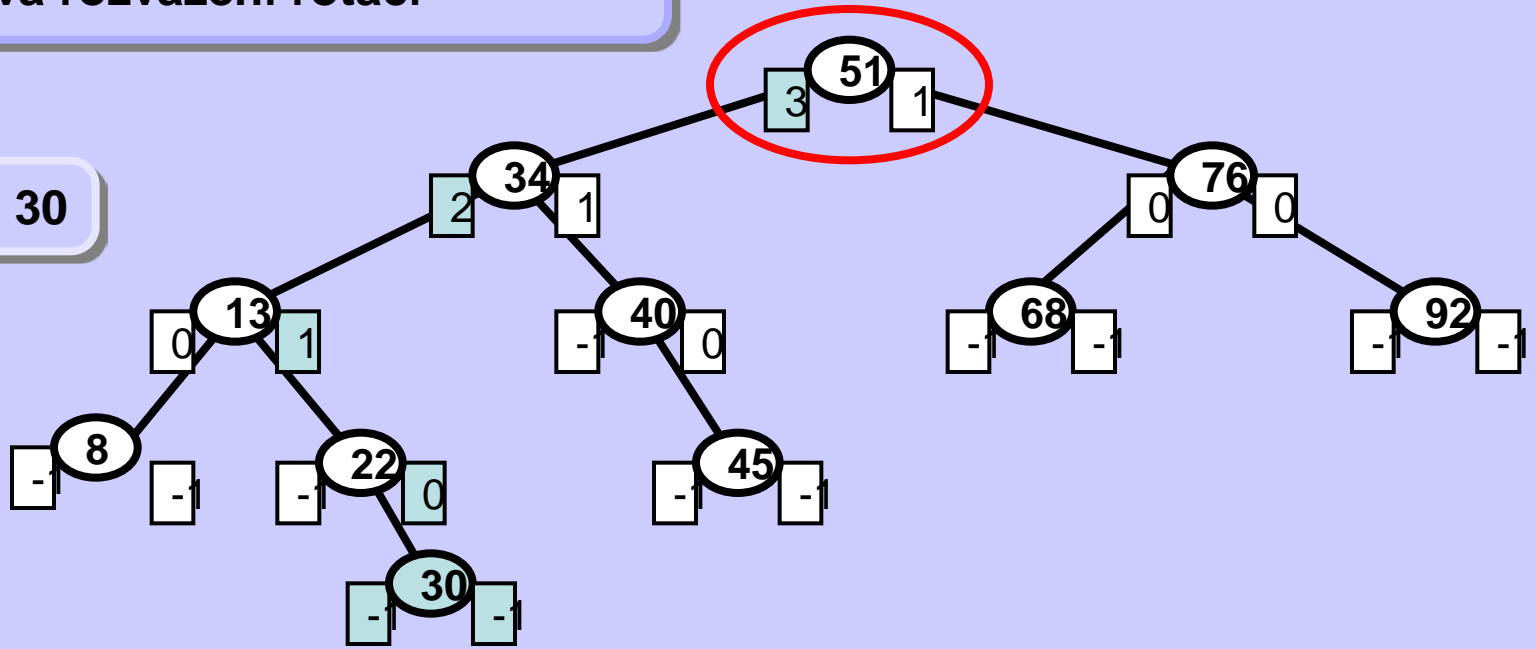


Změněné hloubky

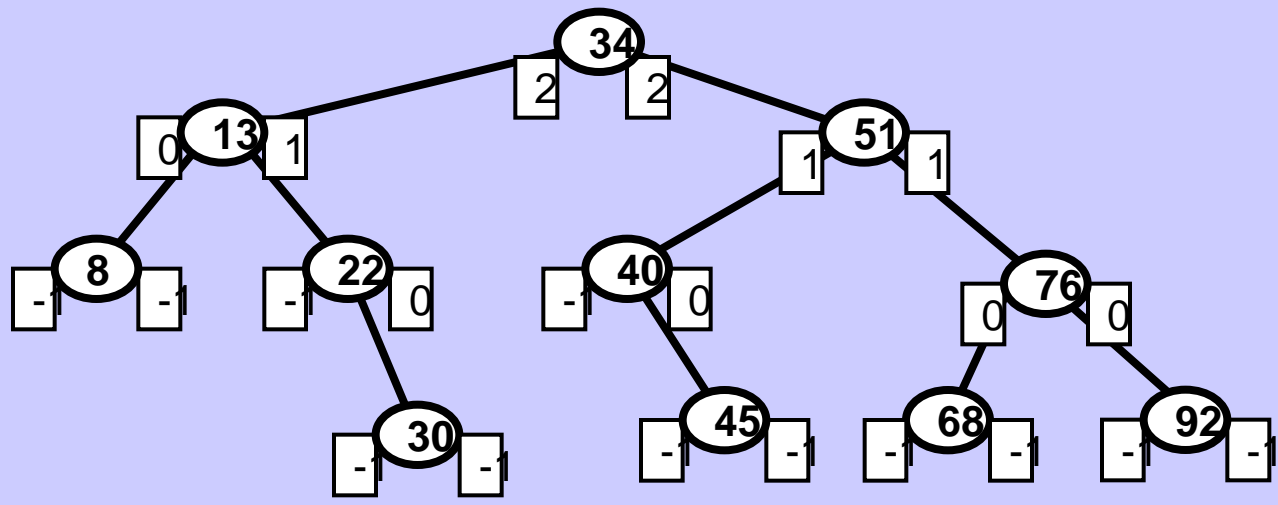
Levý podstrom uzlu 51 je příliš hluboký,
strom přestal být AVL.

Náprava rozvážení rotací

Insert 30

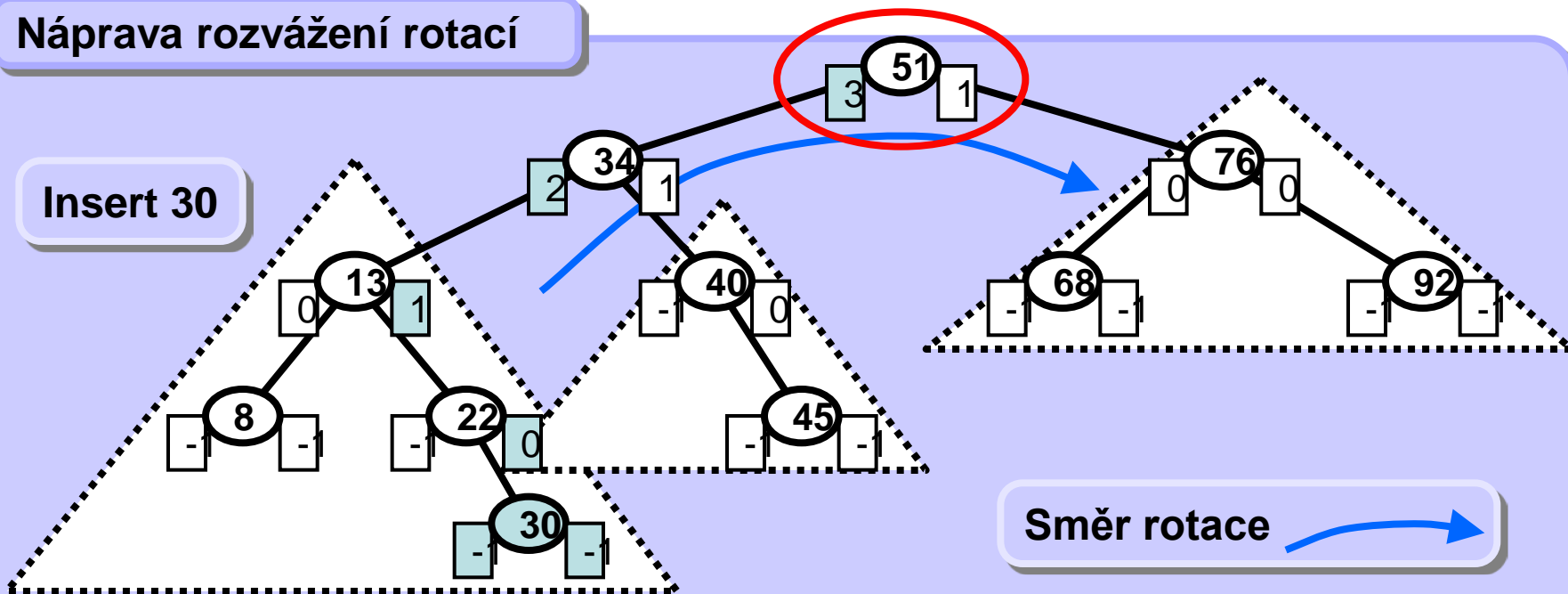


Vyvážený strom
po pravé
jednoduché rotaci,
tzv. R rotaci



Náprava rozvážení rotací

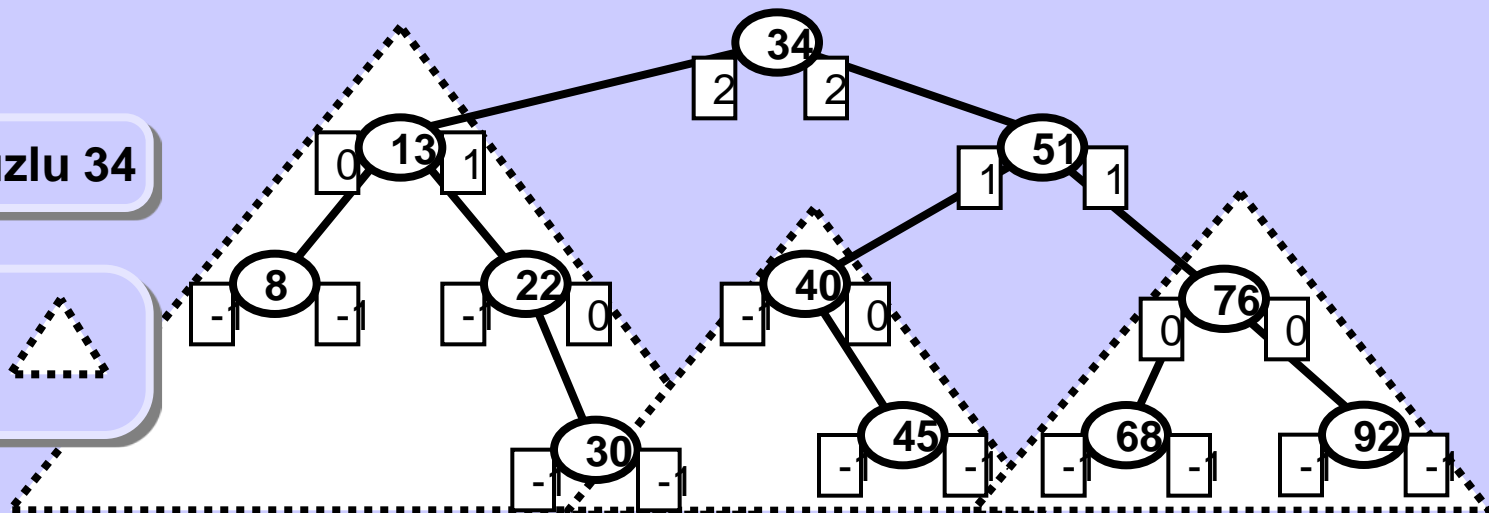
Insert 30



Směr rotace

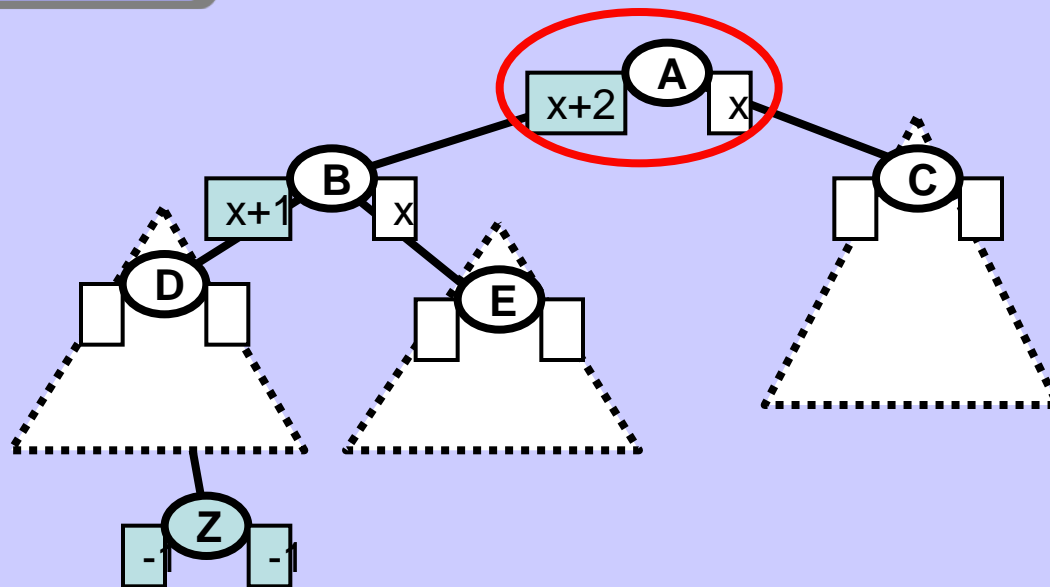
R rotace v uzlu 34

Podstromy
beze změn

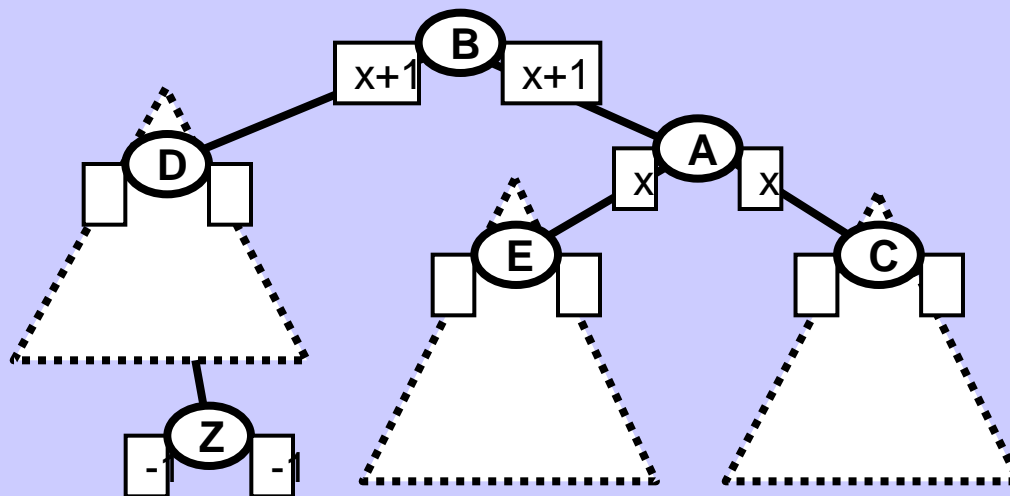


Rotace R obecně

Před

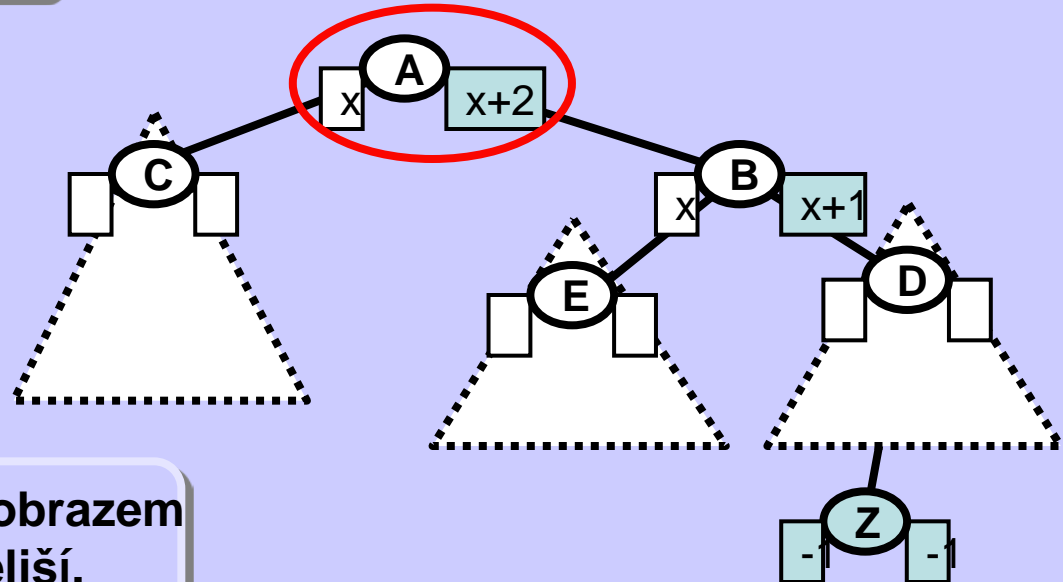


Po



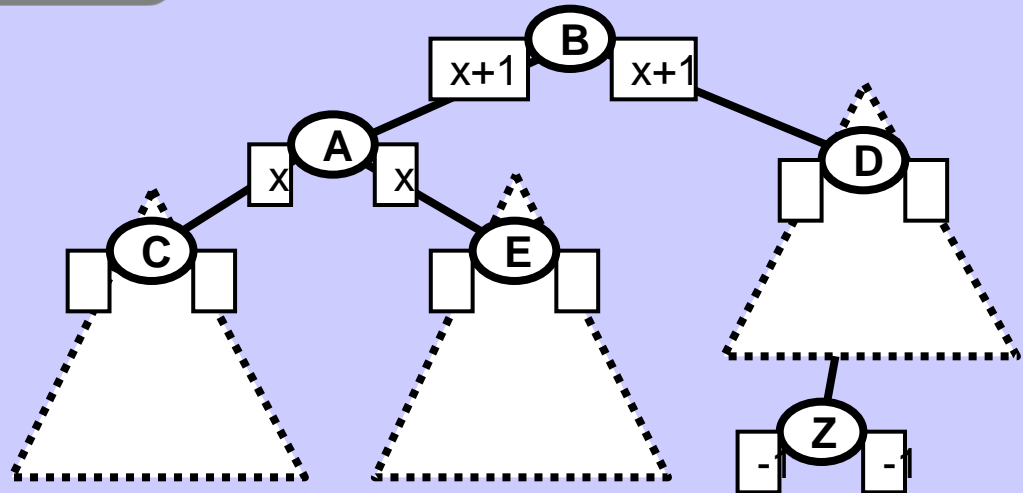
Rotace L obecně

Před

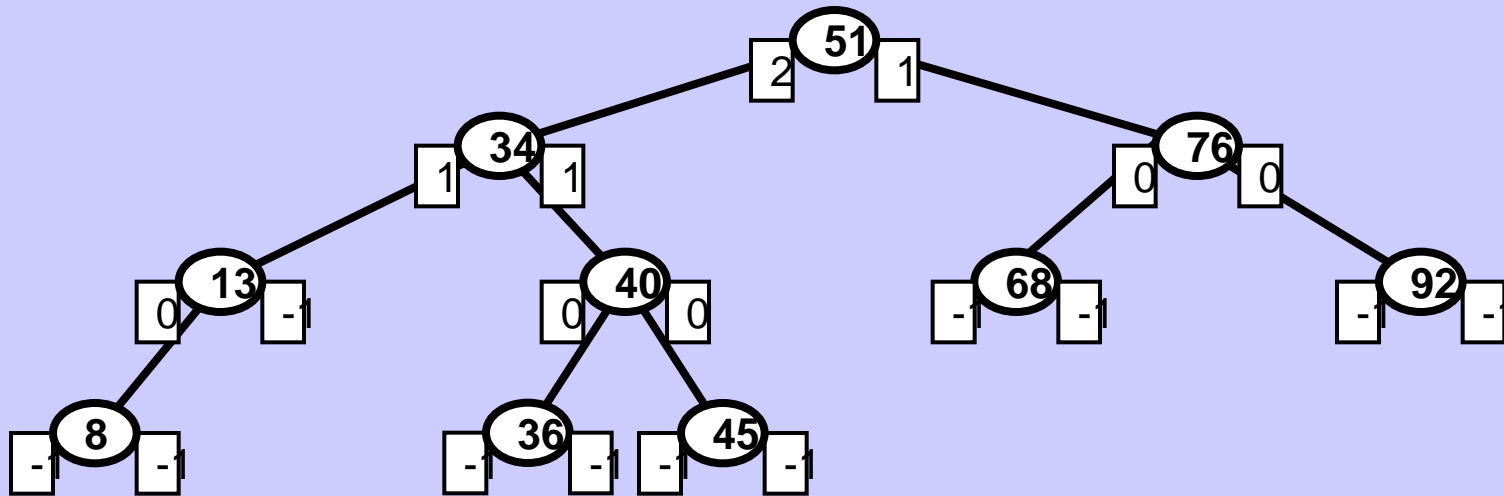


Rotace L je symetrickým obrazem rotace R, jinak se od ní neliší.

Po



AVL strom

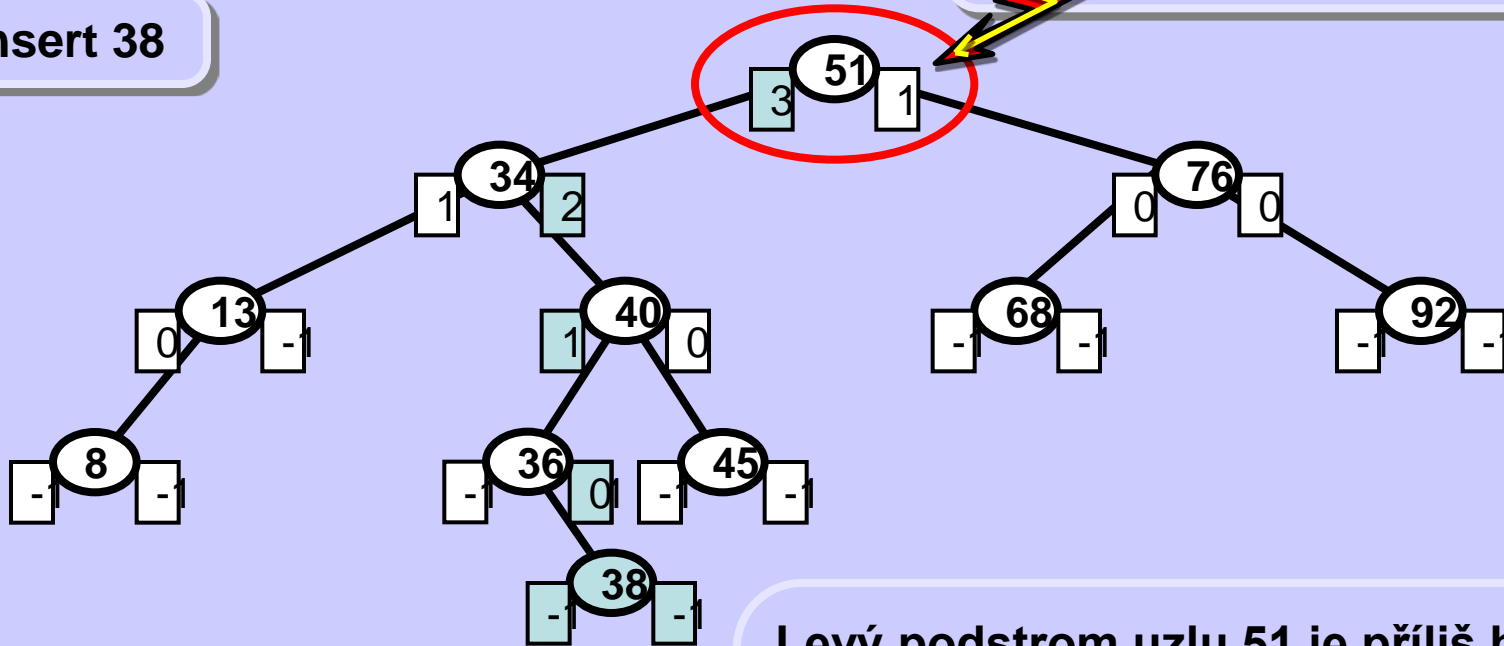


Strom pro demonstraci LR rotace

Vložení uzlu může způsobit rozvážení stromu.

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

Insert 38

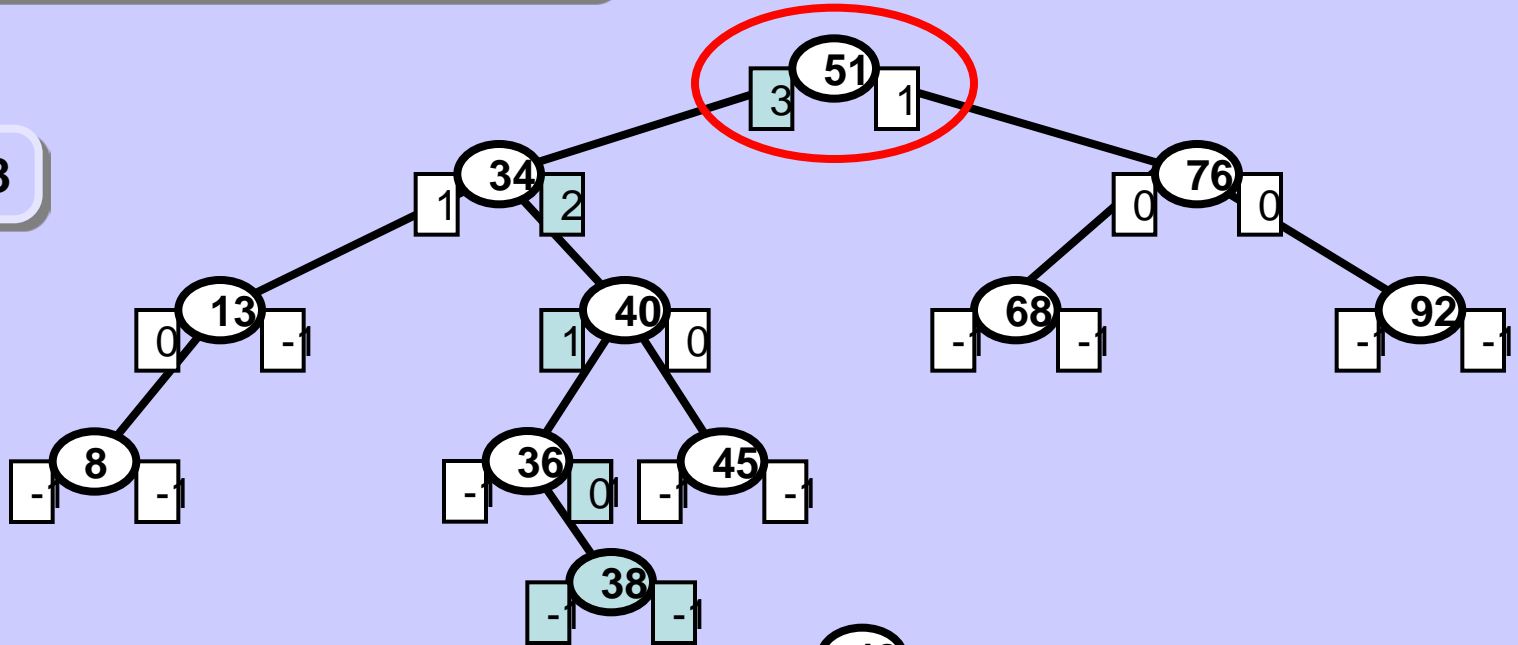


Změněné hloubky

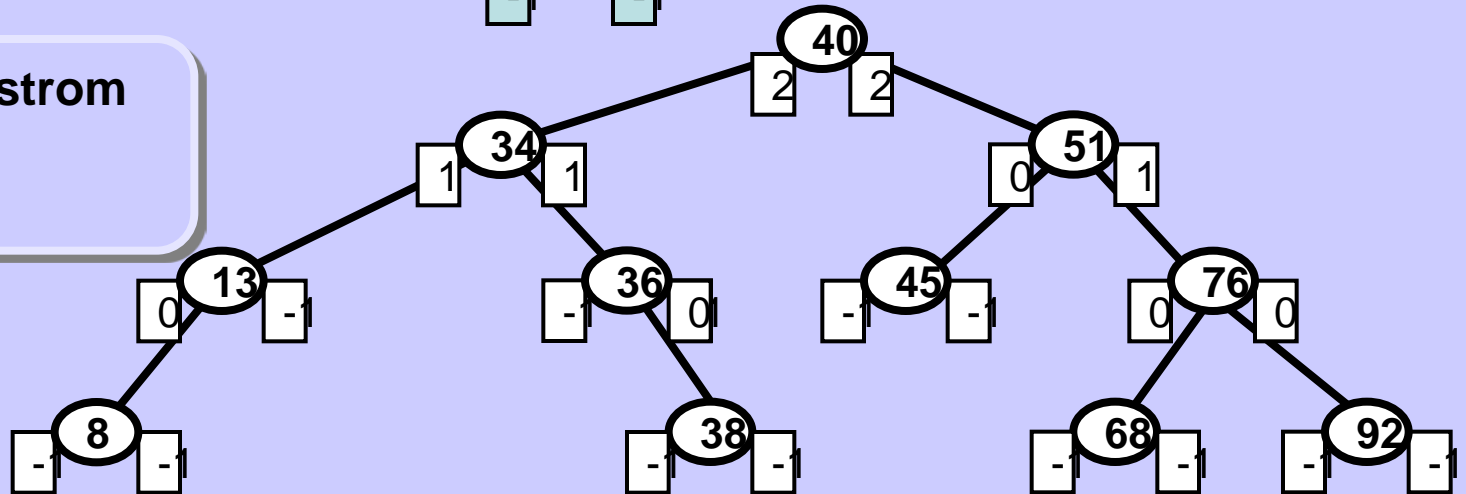
Levý podstrom uzlu 51 je příliš hluboký,
strom přestal být AVL.
Použití rotace R by nepomohlo,
příliš hlubokým by se stal
pravý podstrom uzlu 34.

Náprava rozvážení rotací

Insert 38

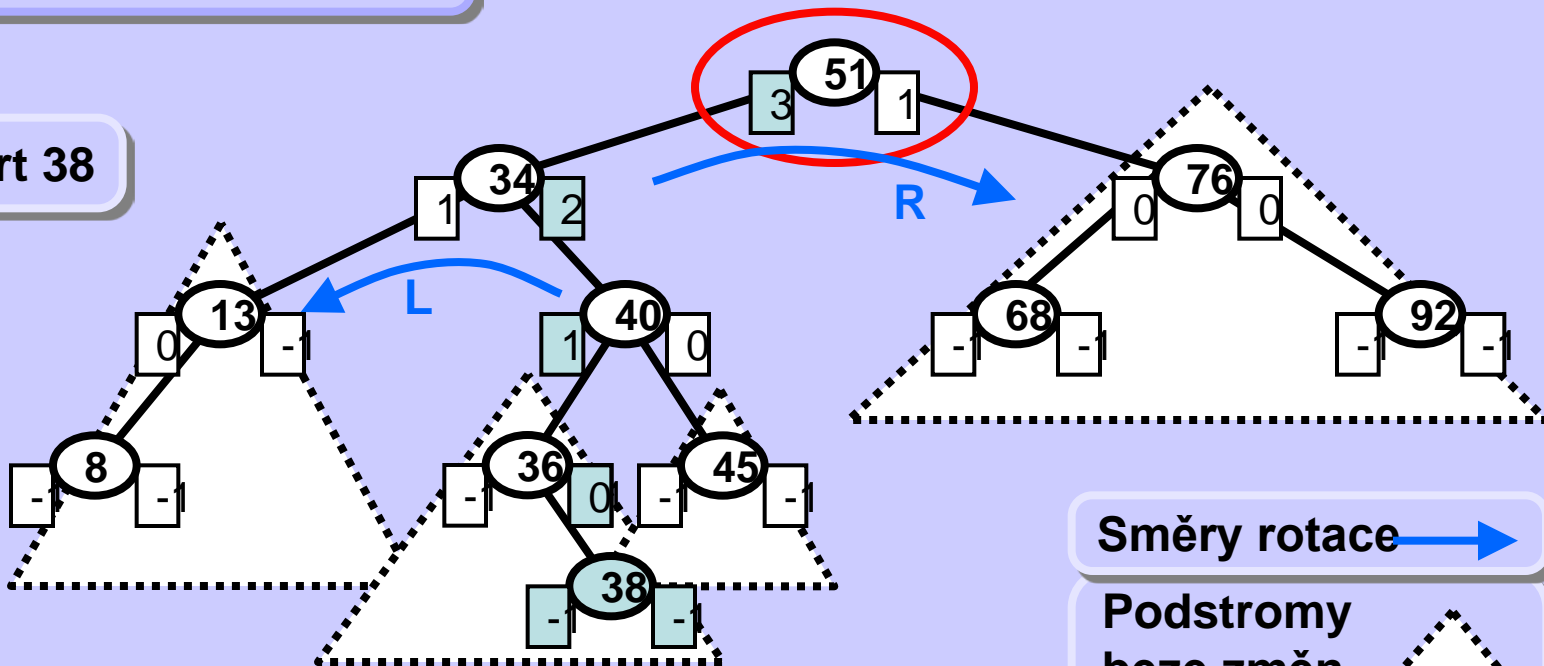


Vyvážený strom
po dvojité
LR rotaci



Náprava rozvážení rotací

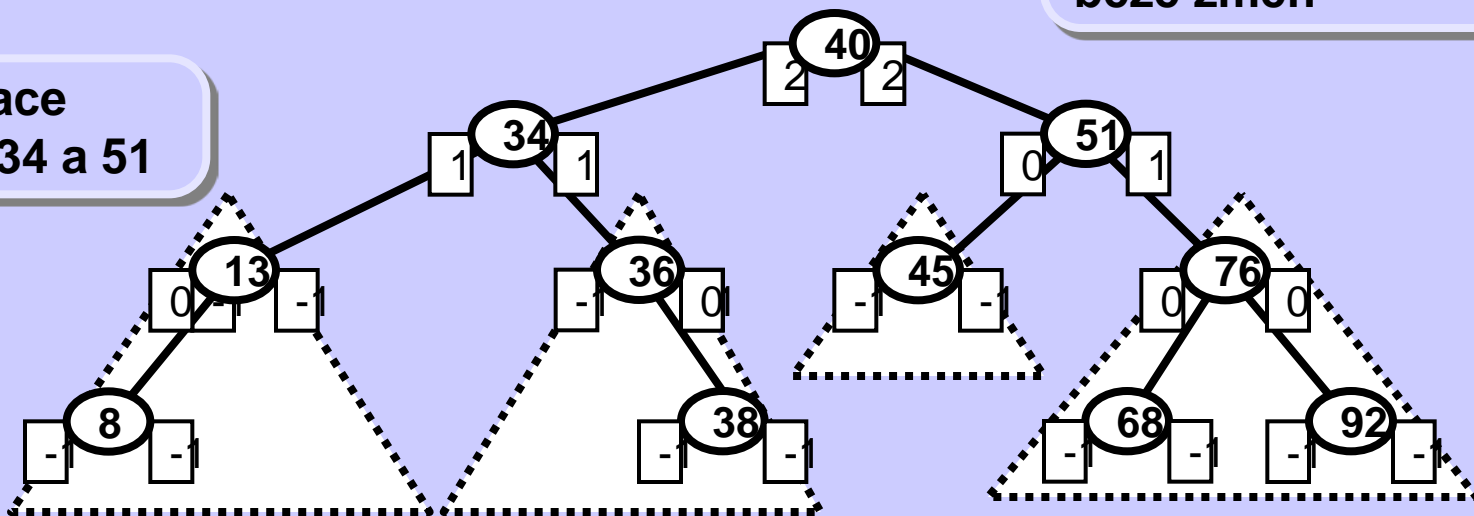
Insert 38



Směry rotace →

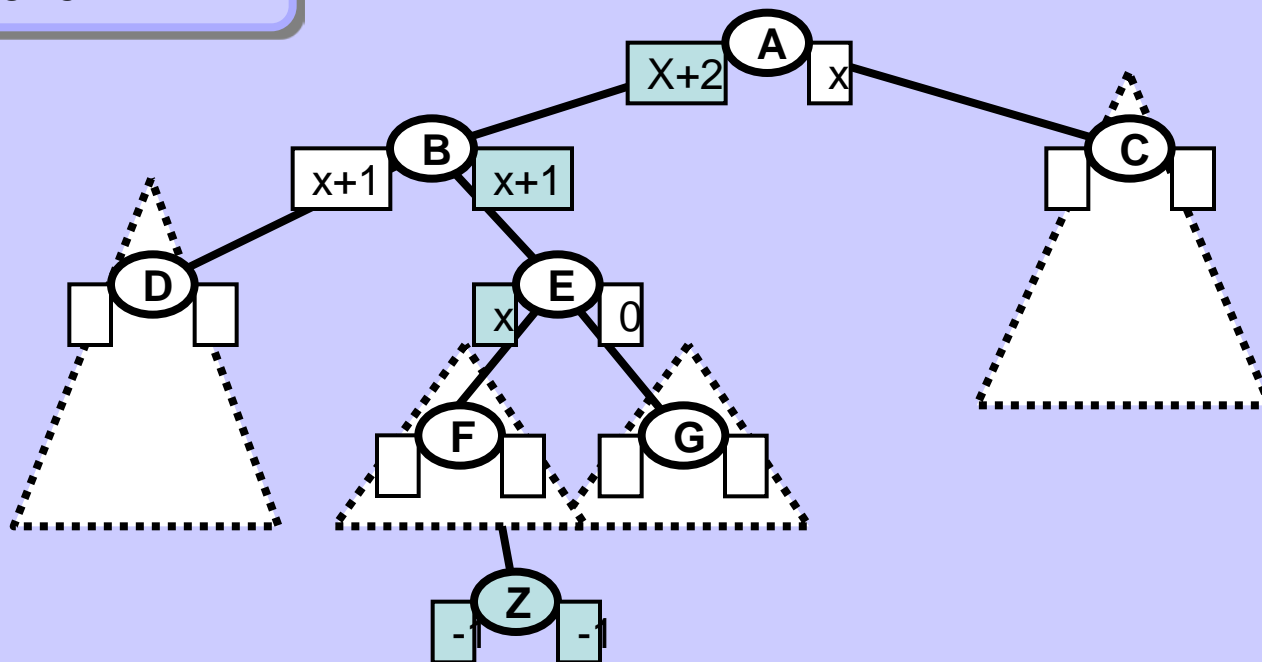
Podstromy
beze změn

LR rotace
v uzlu 34 a 51

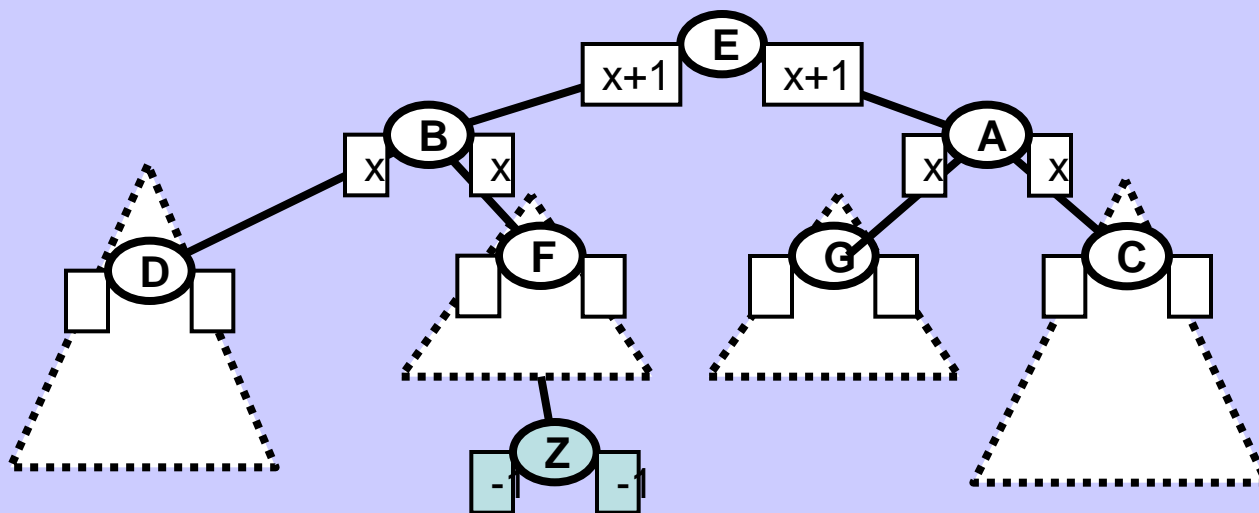


Rotace LR obecně

Před

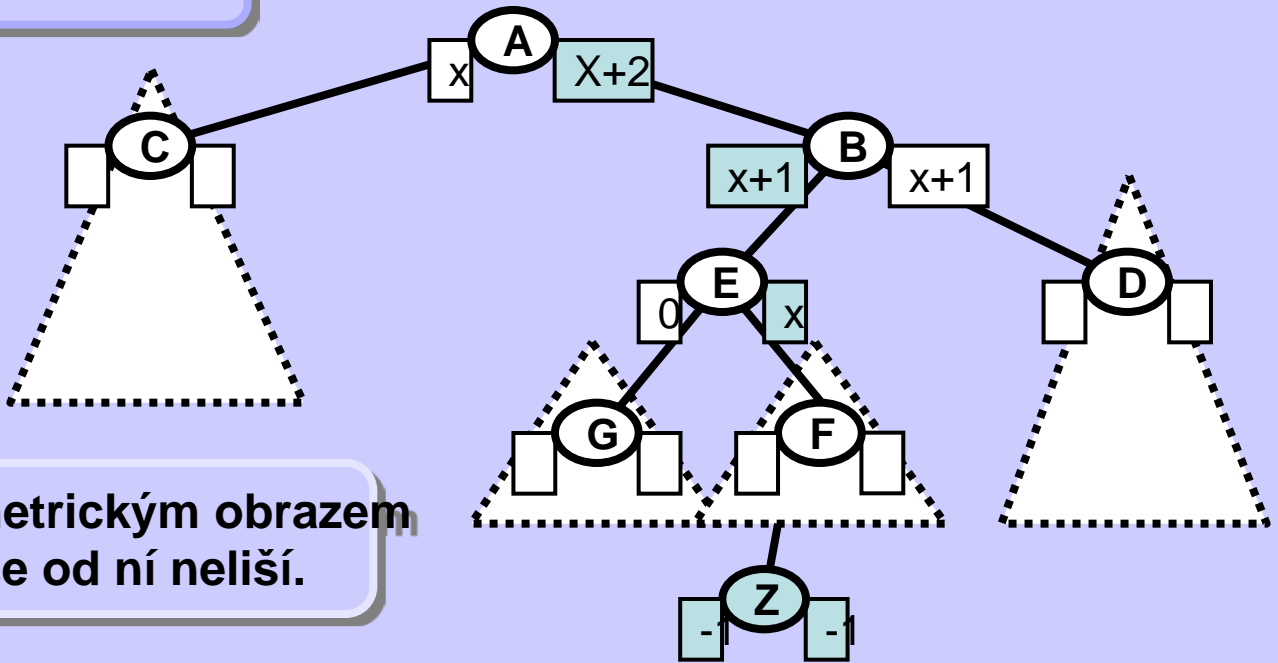


Po



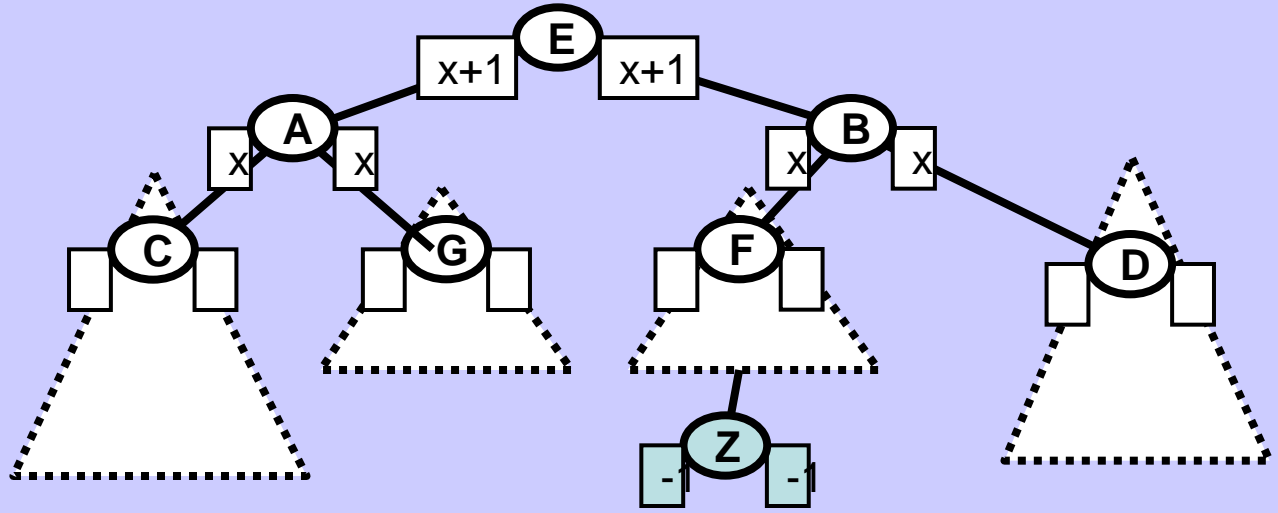
Rotace RL obecně

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 (nebo smazaného, viz dále) 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.

Když narazíme na rozvážený uzel, do kterého jsme bezprostředně došli dvěma hranami *doleva* nahoru, provedeme v tomto uzlu L rotaci.

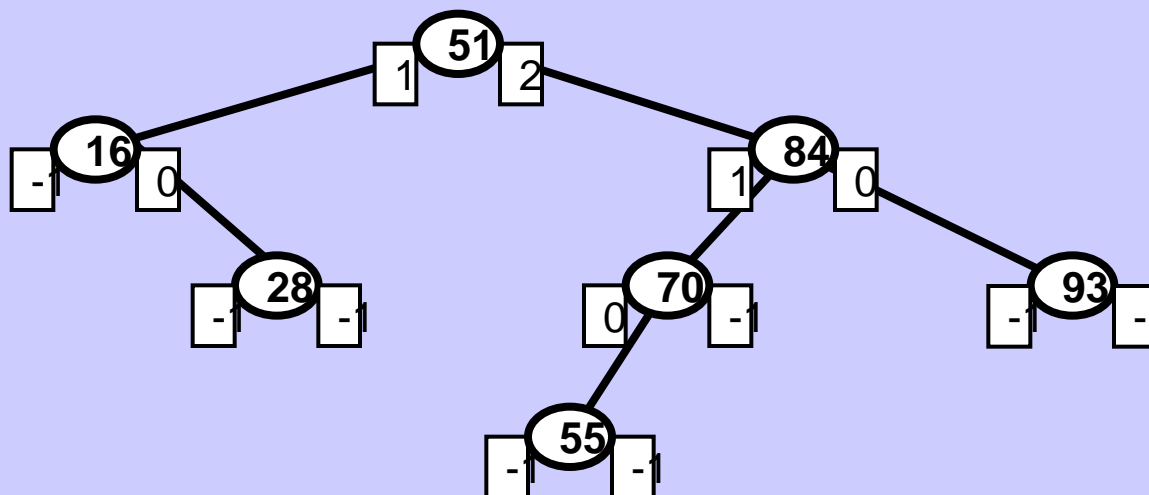
Když narazíme na rozvážený uzel, do kterého jsme bezprostředně došli hranami *doleva a pak doprava* nahoru, provedeme v tomto uzlu LR rotaci.

Když narazíme na rozvážený uzel, do kterého jsme bezprostředně došli hranami *doprava a pak doleva* nahoru, provedeme v tomto uzlu RL rotaci.

Po provedení jedné rotace je AVL strom opět vyvážen .

Delete v AVL stromu

Strom pro demonstraci rotace po smazání uzlu



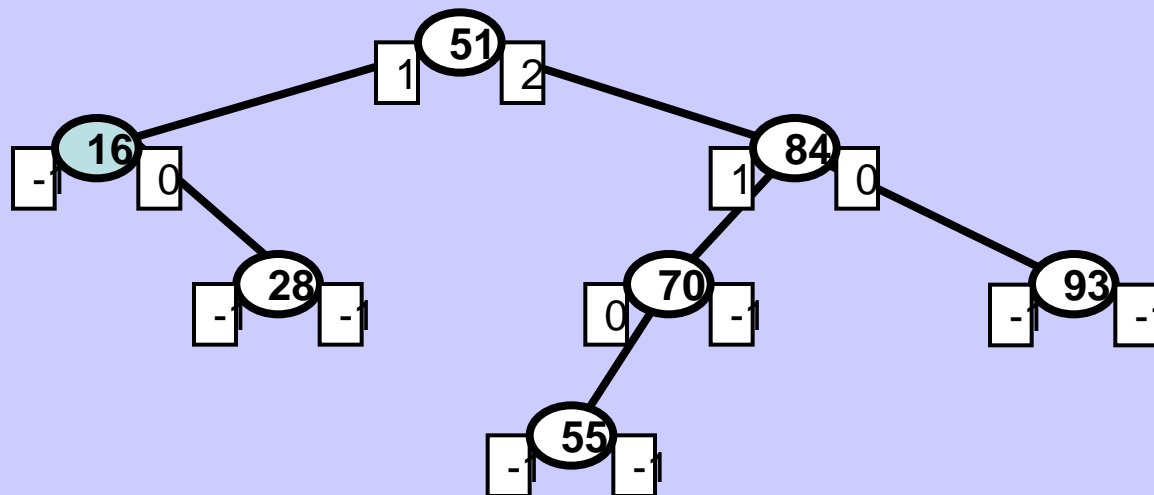
Delete 16

Delete proběhne standardně jako v obyčejném BVS.

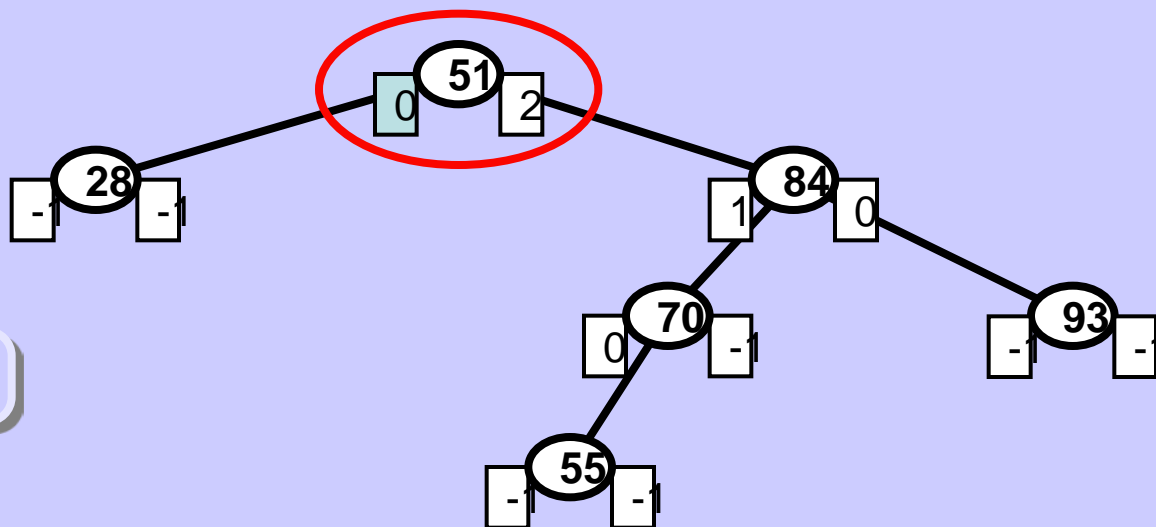
Poté postupujeme od místa smazání nahoru ke kořeni a aktualizujeme výšky podstromů v každém uzlu. Při rozvážení aplikujeme rotaci podobně jako při vkládání.

Delete v AVL stromu

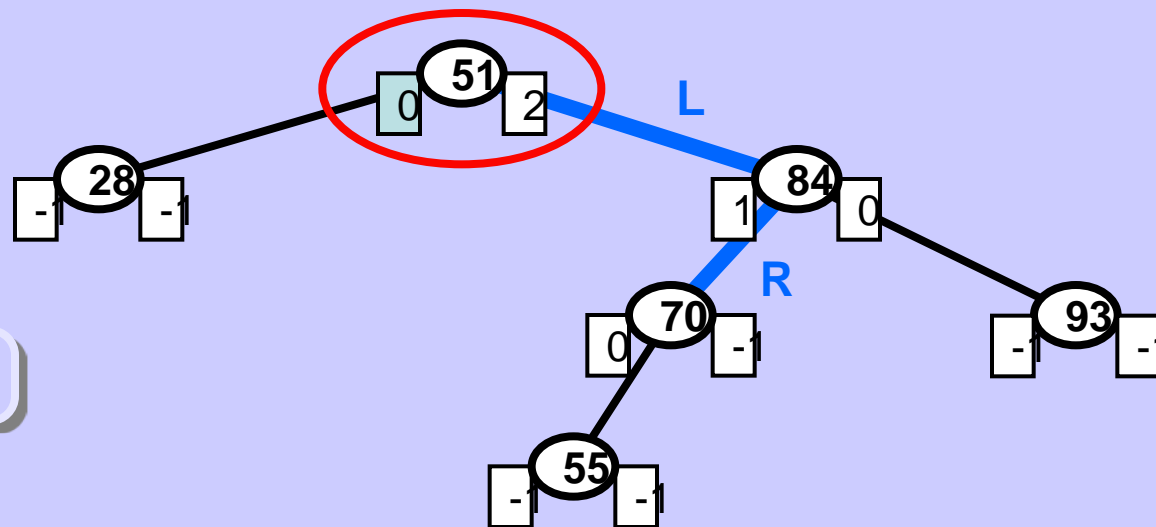
Delete 16



Změněné hloubky



Delete v AVL stromu

Změněné hloubky

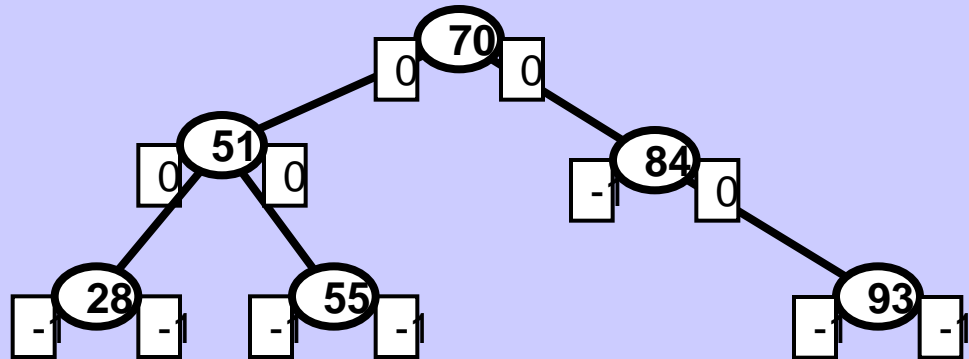
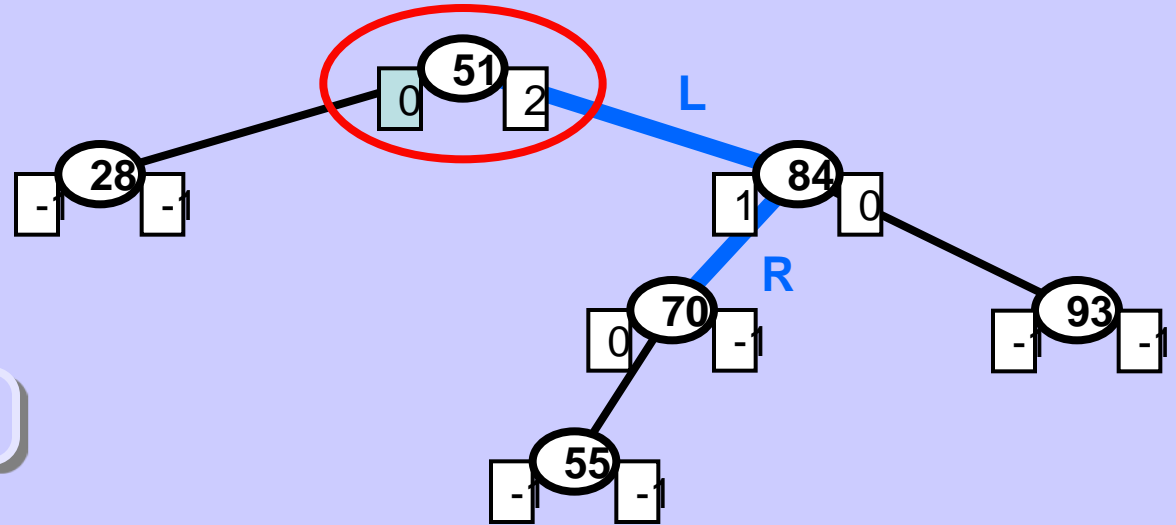
Z rozváženého uzlu 51 prozkoumáme kořen sousedního podstromu, než ze kterého jsme přišli, v tomto případě uzel 84. Má-li tento oba své podstromy stejně hluboké použijeme jednoduchou L nebo R rotaci. Má-li je různě hluboké (nejvýše se liší o 1), rozhodneme, zda použijeme L, R, LR, RL rotaci, jako kdyby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



Implementace operací v BVS a AVL stromu

...

// homework...

Asymptotické složitosti operací Find, Insert, Delete v BVS

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

B-strom

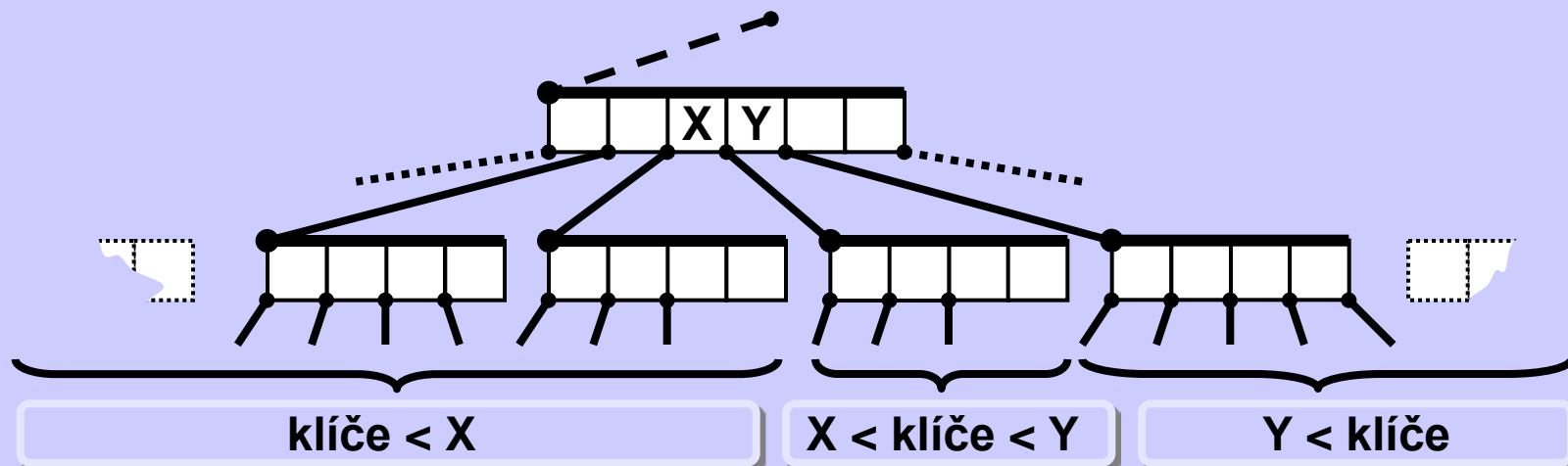
Všechny cesty z kořene do listu jsou stejně dlouhé
tj. B-strom je ideálně vyvážený.

Klíče jsou v uzlu seřazené.

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

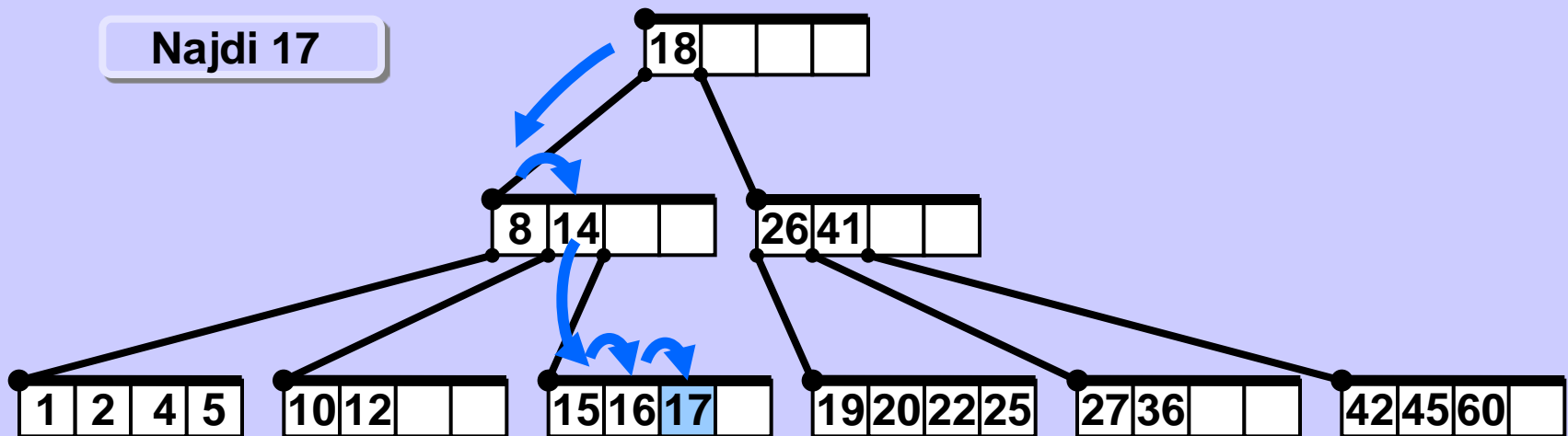
Každý uzel kromě kořene má nejméně k a nejvýše $2k$ klíčů,
každý vnitřní uzel tedy má nejméně $k+1$ a nejvýše $2k+1$ potomků.

Kořen může mít libovolný počet klíčů. Není-li zároveň listem,
má alespoň 2 potomky.



B-strom -- Find

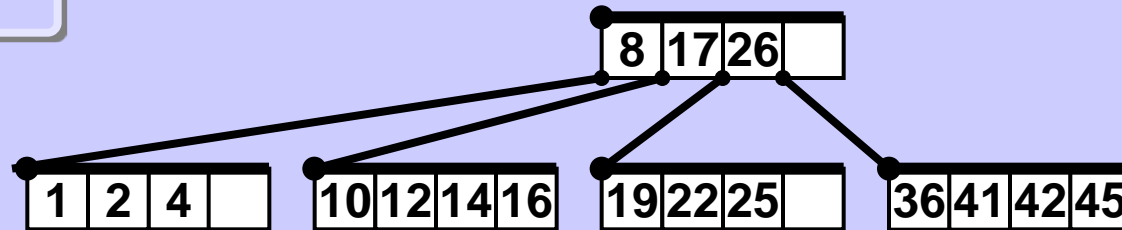
Najdi 17



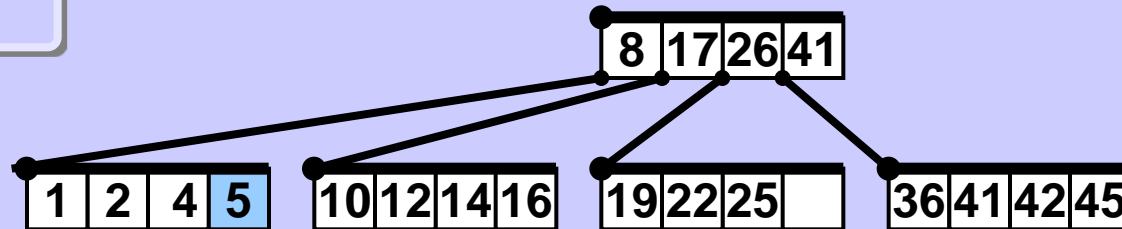
V uzlu se vyhledává sekvenčně.
 Pokud uzel není listem a klíč v něm není,
 hledání pokračuje v odpovídajícím potomku.
 Pokud uzel je listem a klíč v něm není, nenalezeno.

B-strom -- Insert

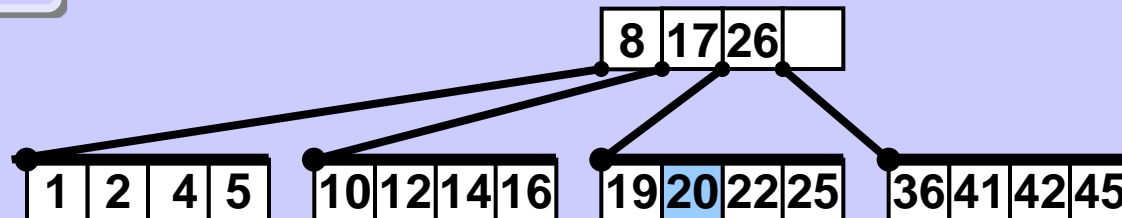
B-strom



Vlož 5

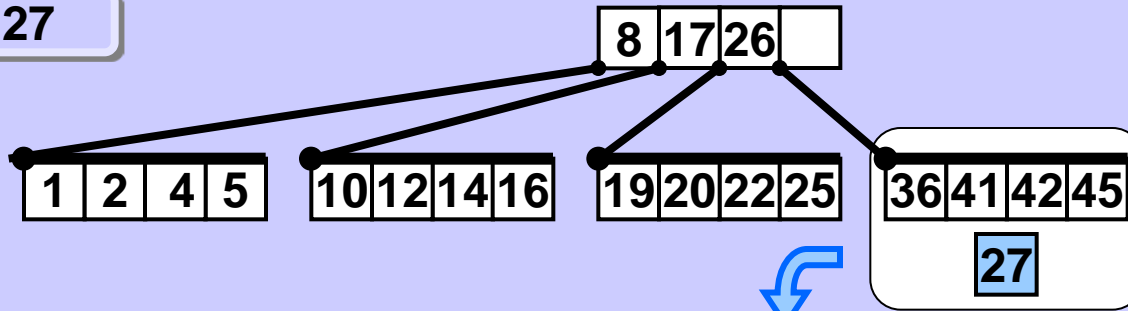


Vlož 20



B-strom -- Insert

Insert 27

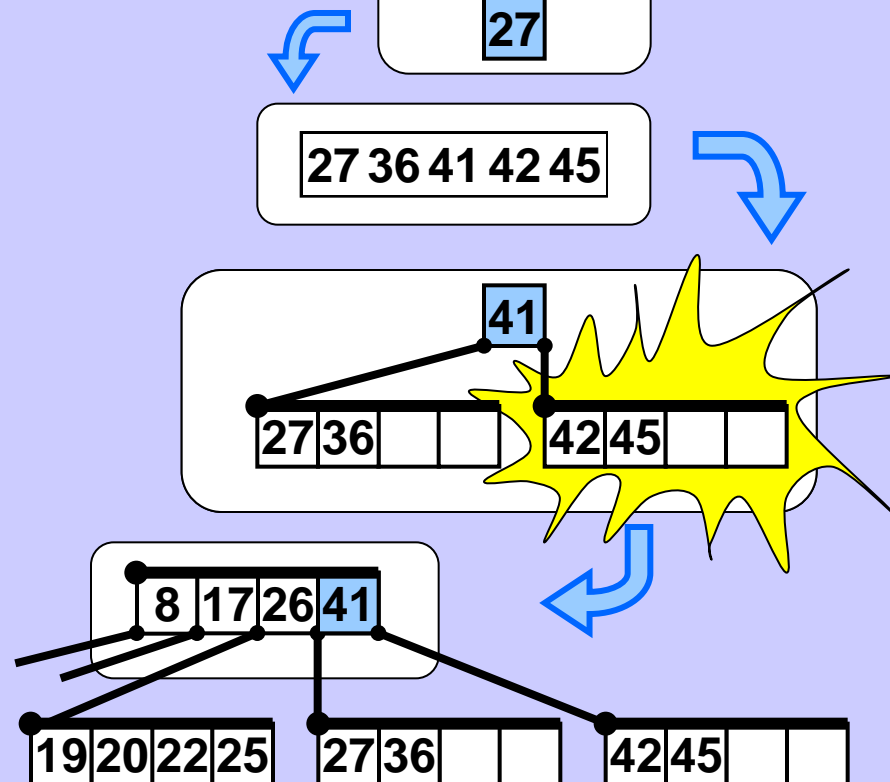


Seřad' mimo strom.

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

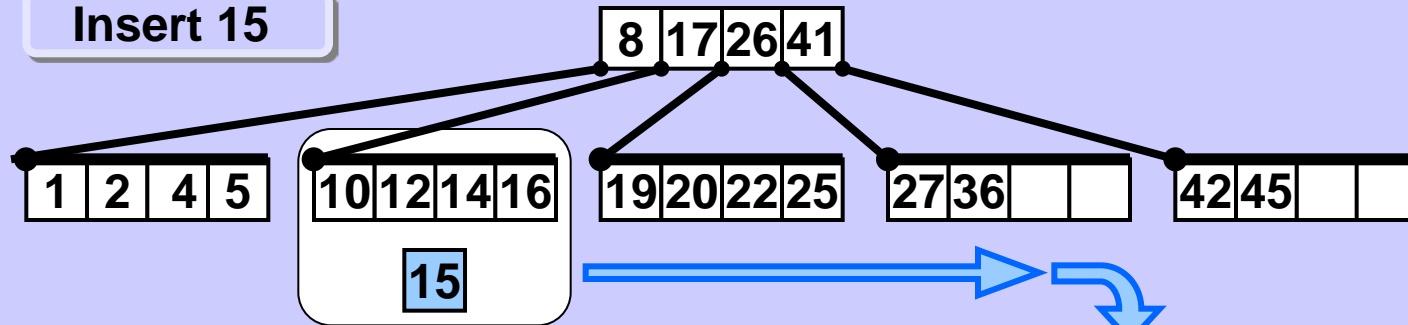
Medián zkus vložit
do rodiče.

Zdařilo se.



B-strom -- Insert

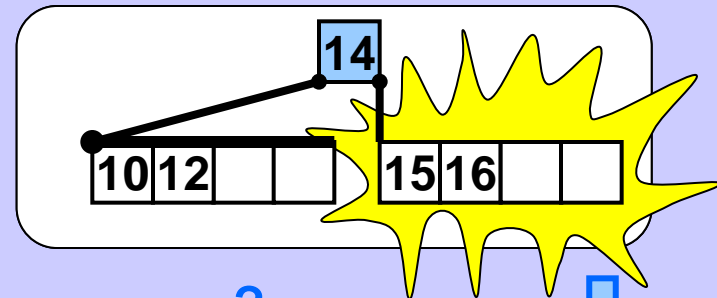
Insert 15



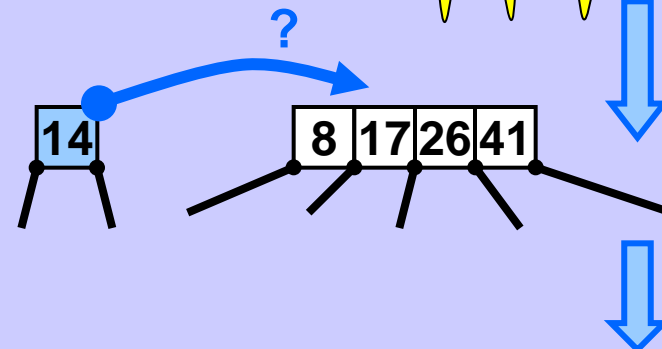
Seřad' mimo strom.

10 12 14 15 16

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

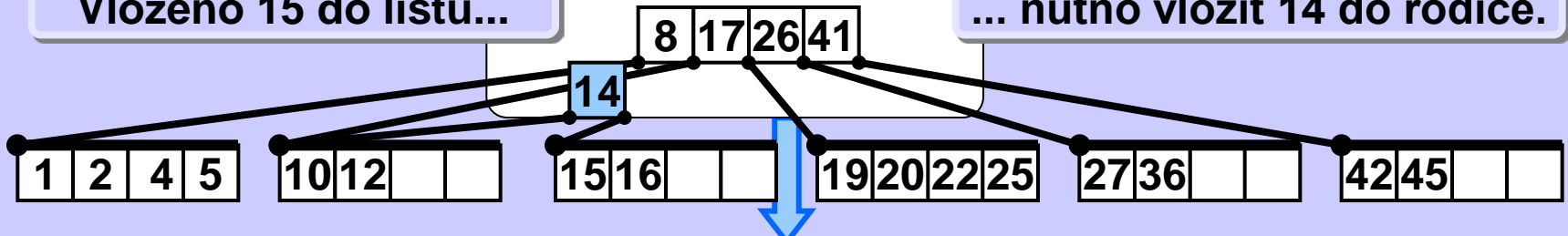


Medián zkus vložit
do rodiče.



B-strom -- Insert

Vloženo 15 do listu...



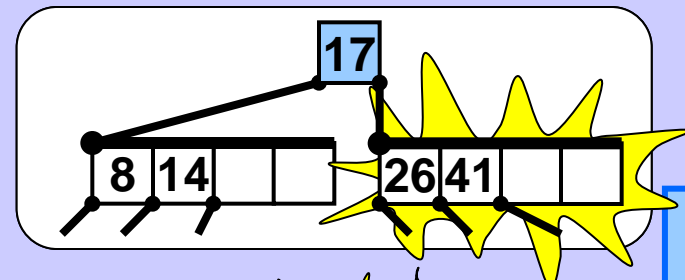
... nutno vložit 14 do rodiče.

Rodič je zaplněn – Analogický další postup směrem ke kořeni.

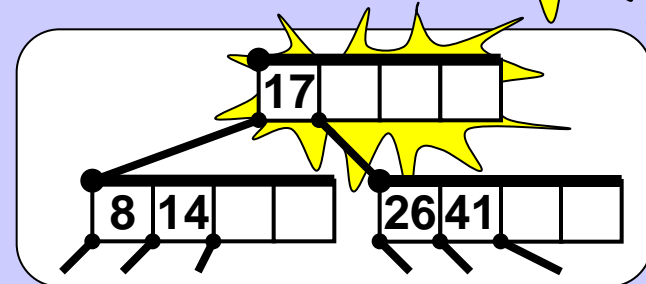
Seřad' mimo strom.

8 14 17 26 41

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



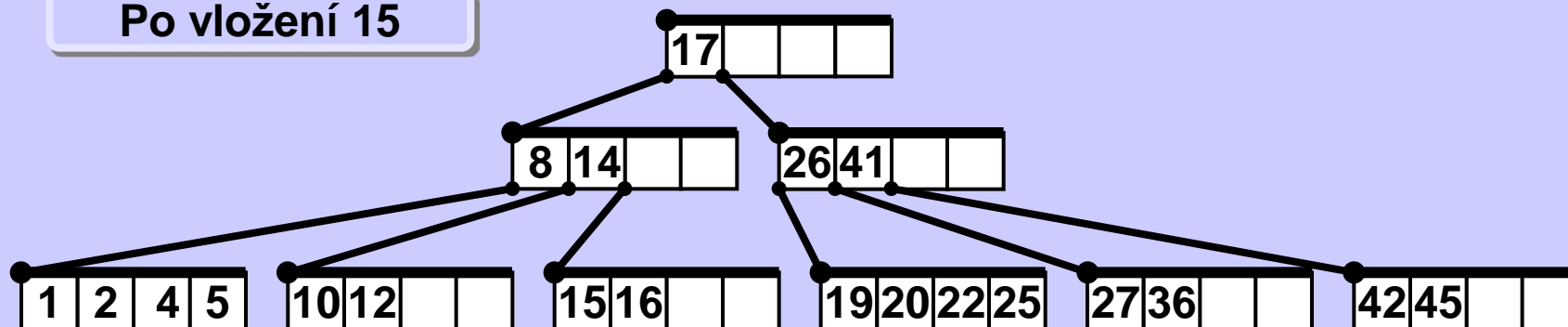
Medián nelze vložit do
rodiče, žádný není, tedy
se zřídí nový kořen.





B-strom -- Insert

Po vložení 15

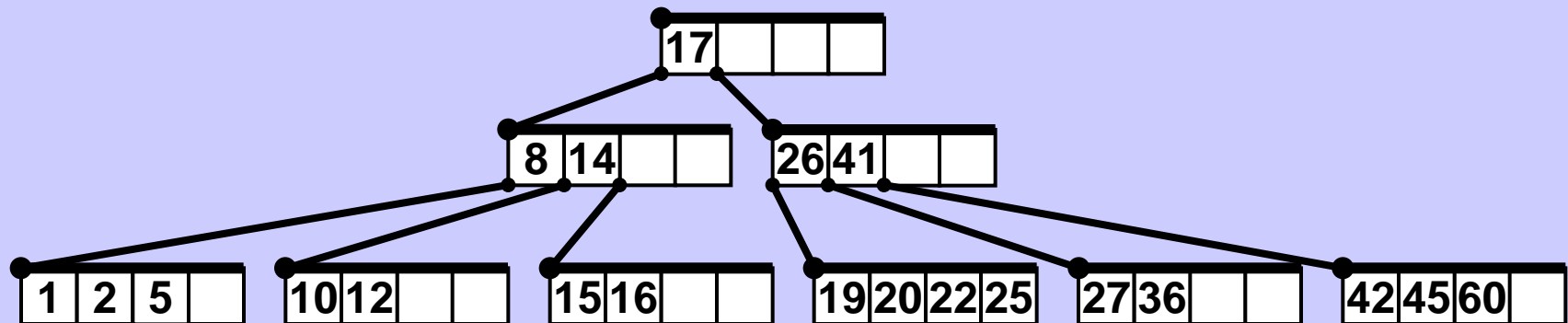
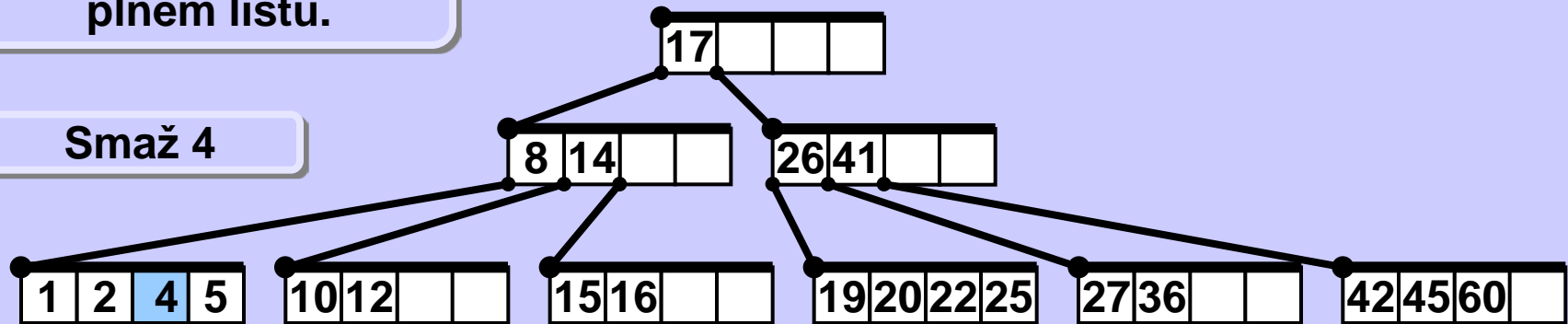


V každém patře přibyl jeden uzel, kromě toho přibyl nový kořen, strom ale roste směrem "vzhůru", zůstává ideálně vyvážený.

B-strom -- Delete

Mazání v dostatečně plném listu.

Smaž 4

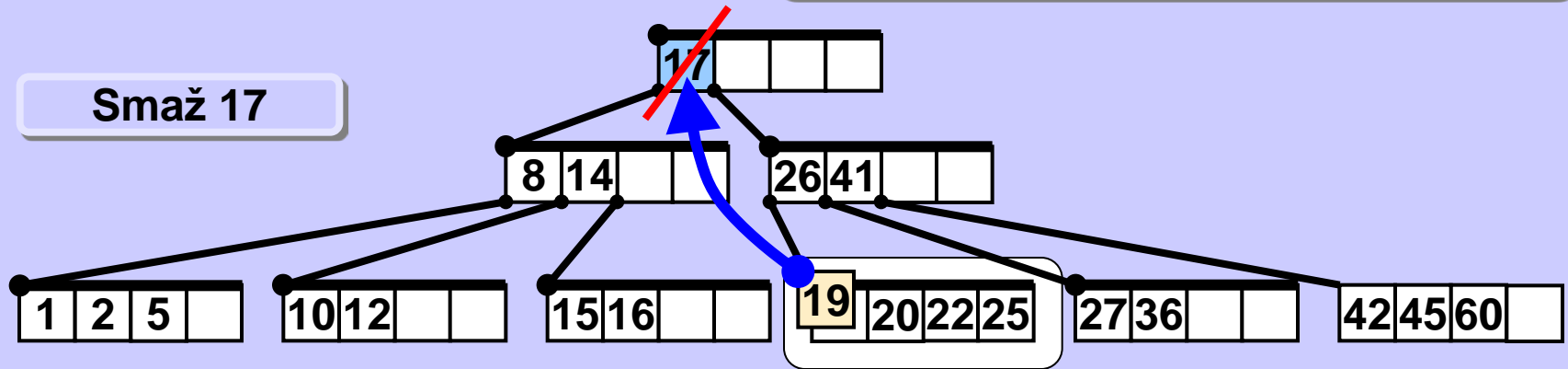


B-strom -- Delete

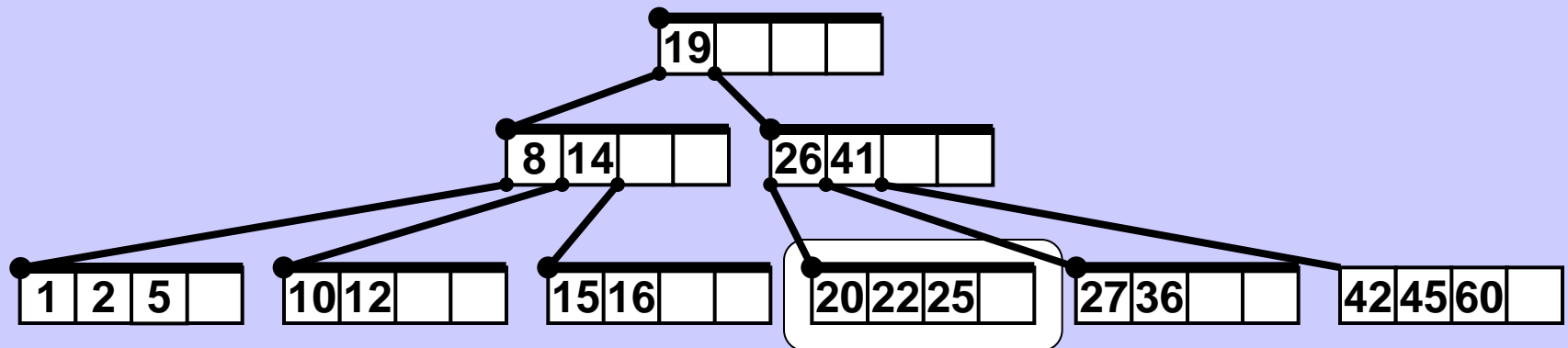
Mazání ve vnitřním uzlu

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

Smaž 17



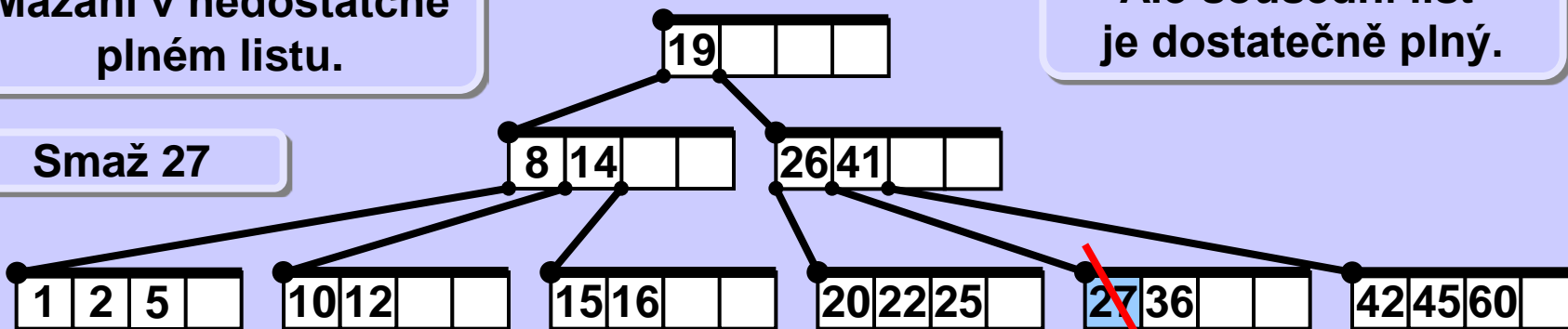
Nejbližší větší(menší) klíč je vždy v B-stromu v listu, má-li tento list dostatečný počet klíčů, jsme hotovi.



B-strom -- Delete

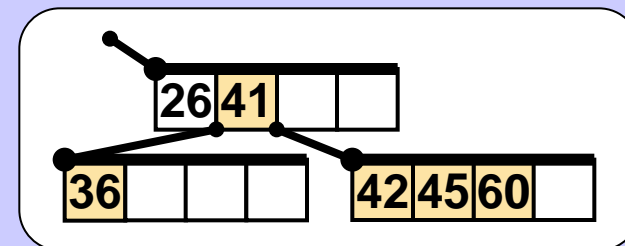
Mazání v nedostatečně plném listu.

Smaž 27



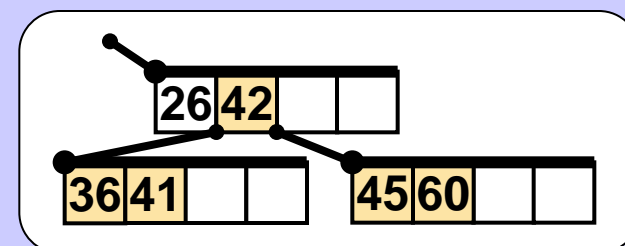
Ale sousední list je dostatečně plný.

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



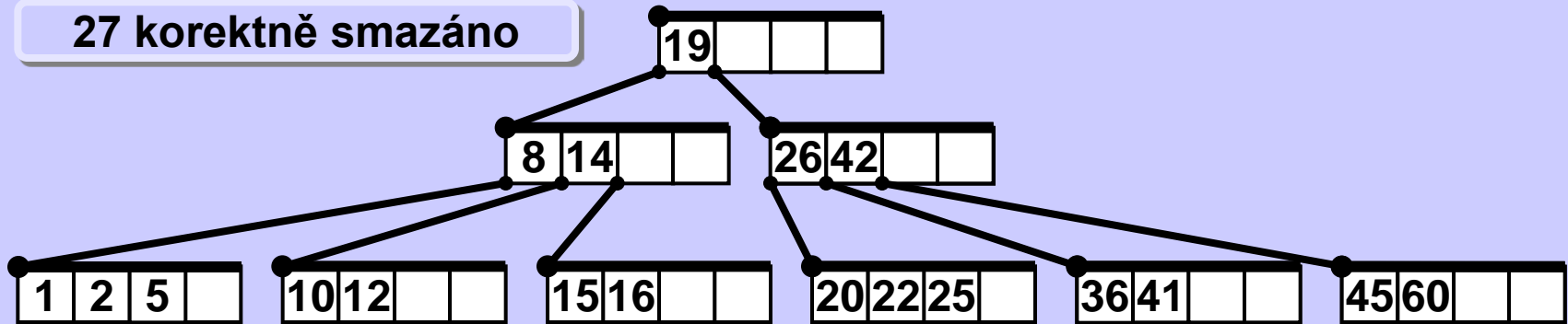
36 41 42 45 60

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



B-strom -- Delete

27 korektně smazáno

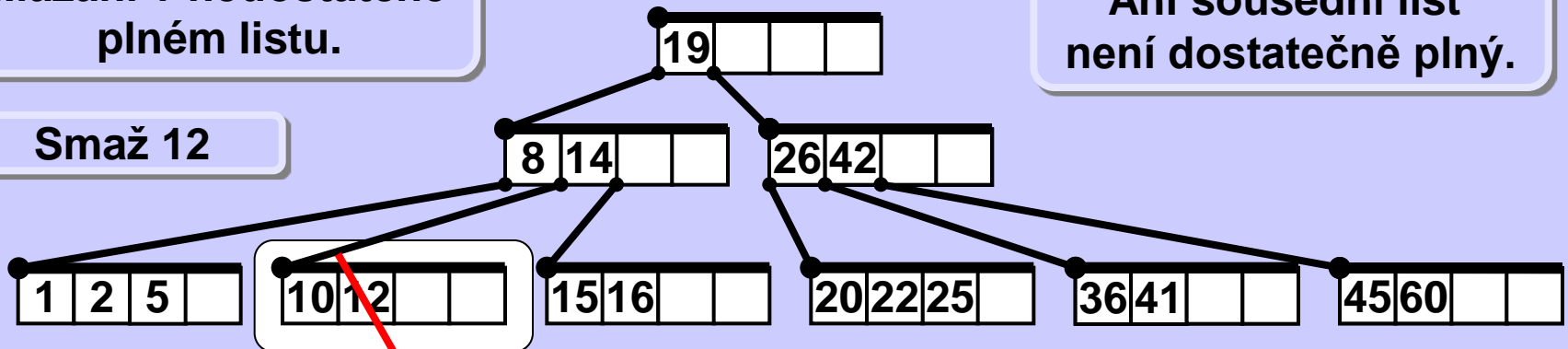


B-strom -- Delete

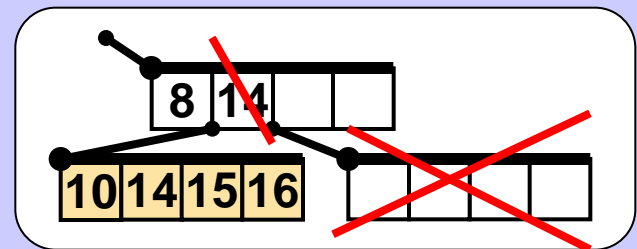
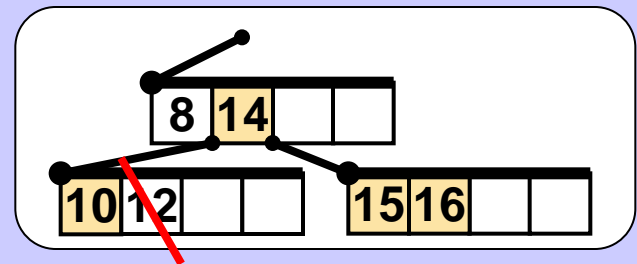
Mazání v nedostatečně plném listu.

Smaž 12

Ani sousední list není dostatečně plný.



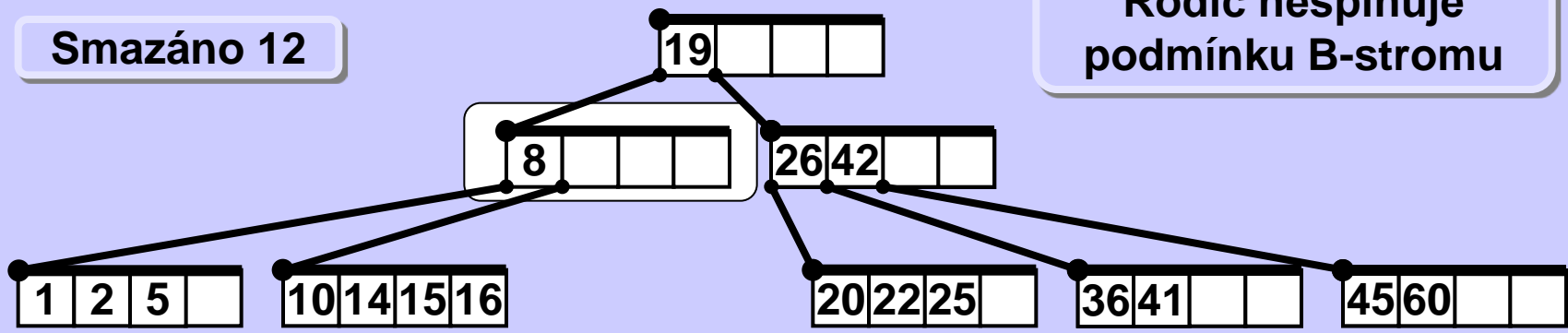
Sjednot' klíče s klíči v sousedním listu a s dělicím klíčem v rodiči a seřad'. Vše vlož do původního listu, sousední list smaž, dělicí klíč v rodiči také smaž.



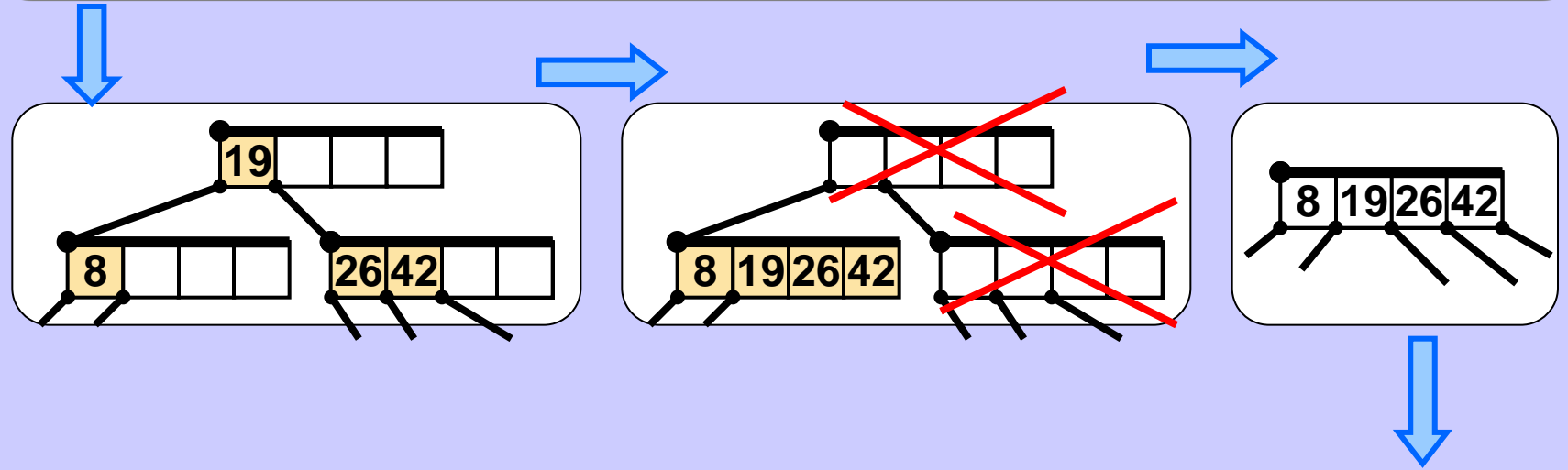
B-strom -- Delete

Smazáno 12

Rodič nespĺňuje podmínku 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.



B-strom -- Delete

Smazáno 12
a strom byl adekvátně restrukturován.

