

RB strom

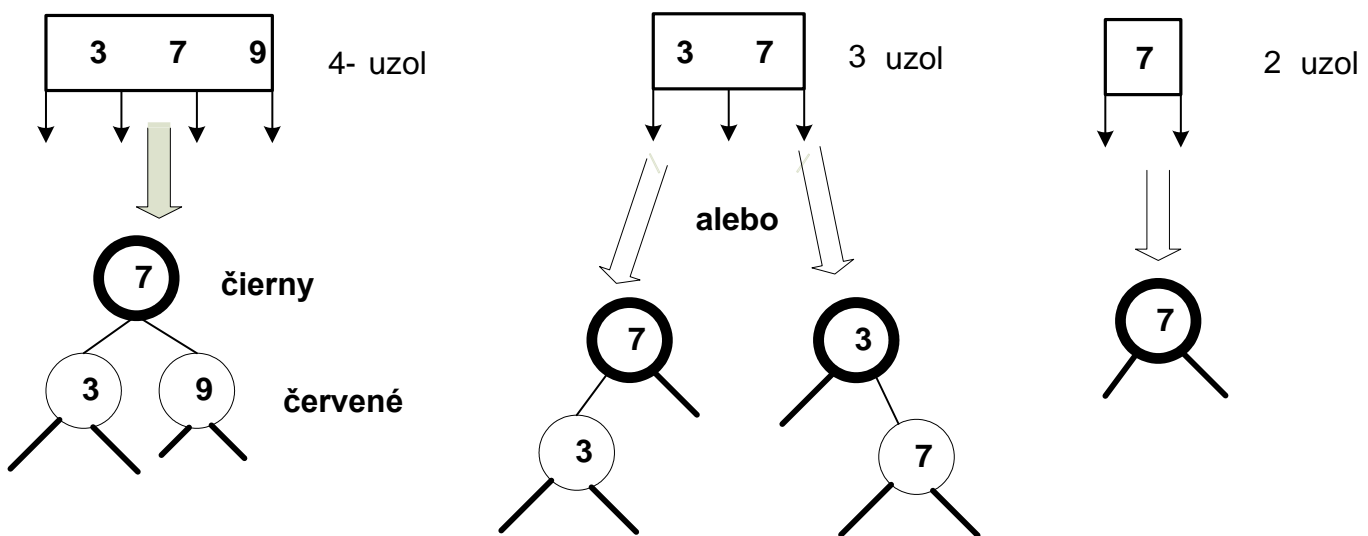
(Red-Black tree, červeno - čierny strom)

- modifikácia 2-4 stromu na binárny tvar (RB je binárny)
- každý uzol má dodatočnú informáciu (farbu)
- štruktúra podporuje jednorozmerné intervalové vyhľadávanie

Vlastnosti:

- koreň je čierny
- červený uzol má výlučne čiernych synov => v reťazcoch vpravo ani vľavo nemôžu byť za sebou 2 červené uzly
- počet čiernych vrcholov na ceste od koreňa do každého z listov je rovnaký (ekvivalent rovnakej dĺžky z koreňa do každého z listov v 2-4 strome)

Korešpondencia medzi uzlami 2-4 stromu a RB stromu:



Null budeme považovať za čierny smerník na prázdny čierny uzol - tento prázdny uzol nekreslíme.

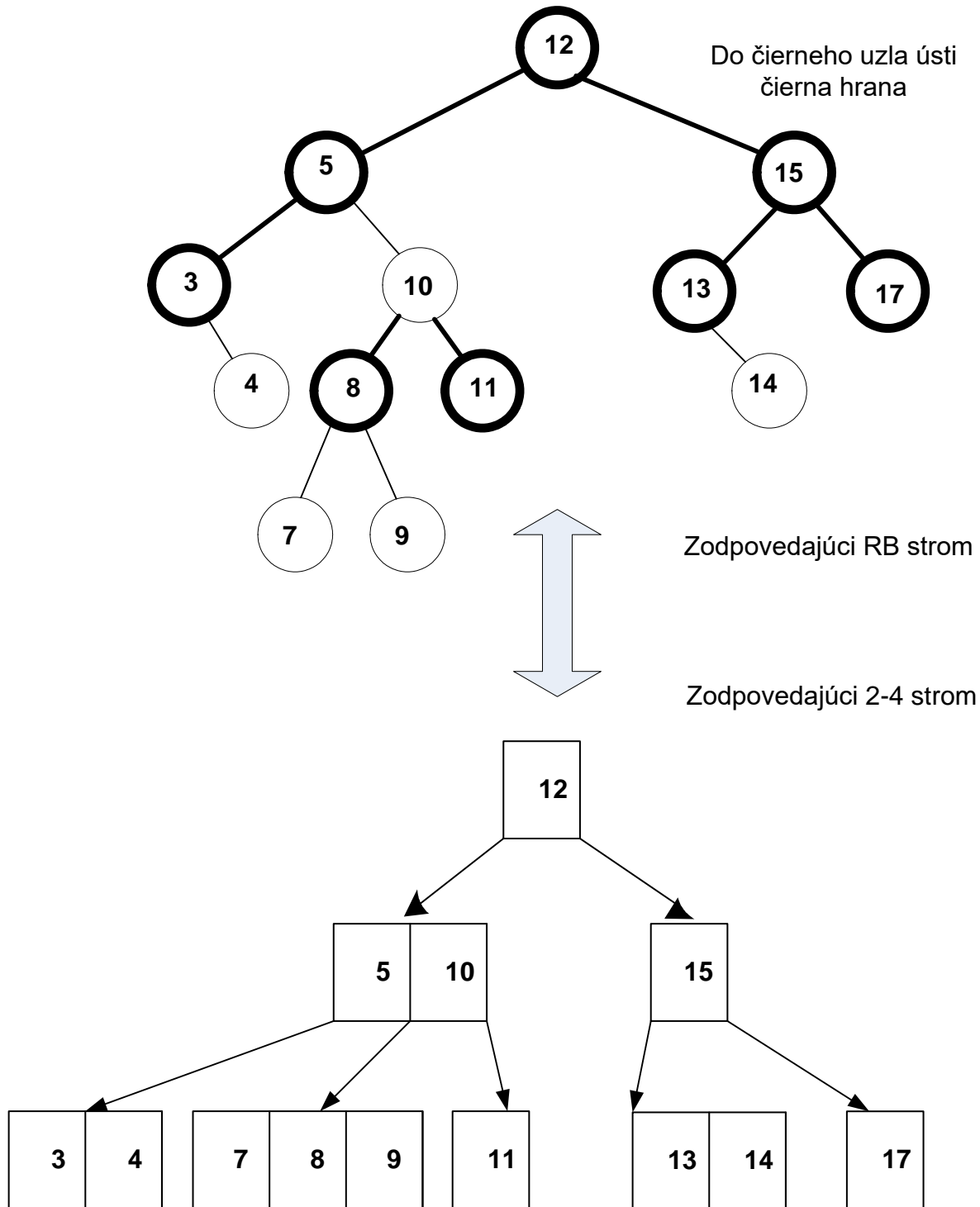
Výhody RB- stromu:

- vyhľadávanie a prehliadky ako v BVS (ignoruje sa farba uzlov)
- operácie vkladania a odoberania sú modifikované algoritmy z 2-4 stromov a teda udržiavajú strom vyvážený

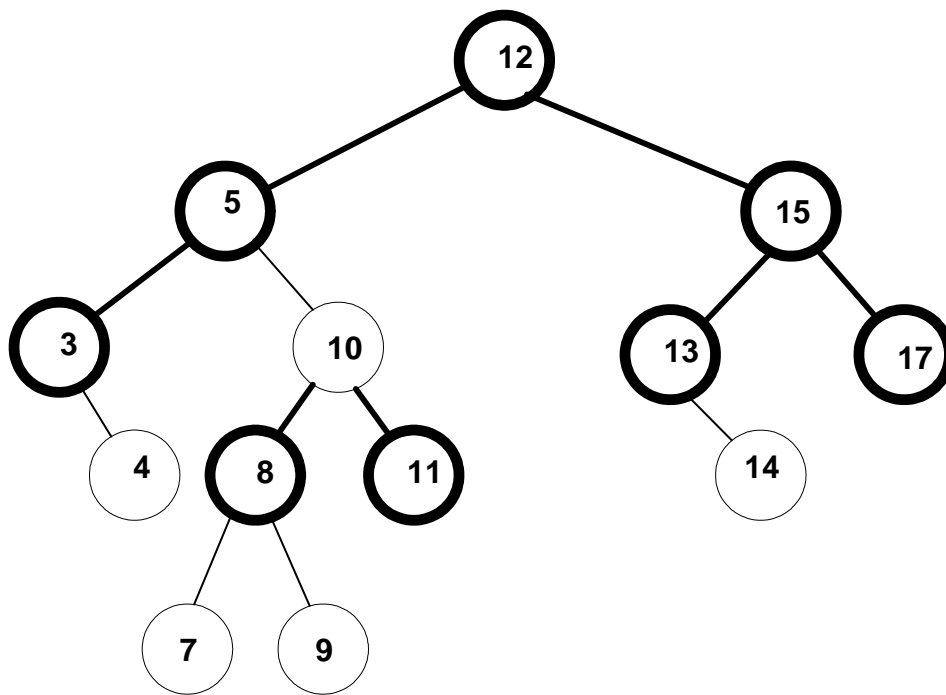
Priemerná aj najhoršia zložitosť všetkých operácií na RB-strome je $O(\log_2 n)$. RB strom nemôže zdegenerovať na lineárny zoznam, ale nie je to

dokonale vyvážený strom. Na rozdiel od AVL stromu môže byť rozdiel výšky ľavého a pravého podstromu väčší ako 1.

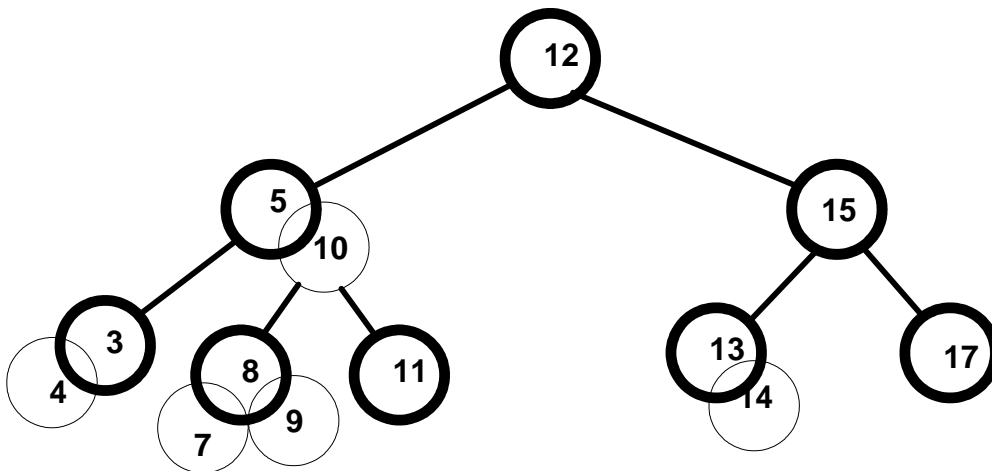
Príklad:



Priemerná výška RB stromu je $O(\log_2 n)$.



Intuitívna transformácia na 2-4 strom:



Vkladanie do RB stromu: (záznam X s kľúčom K)

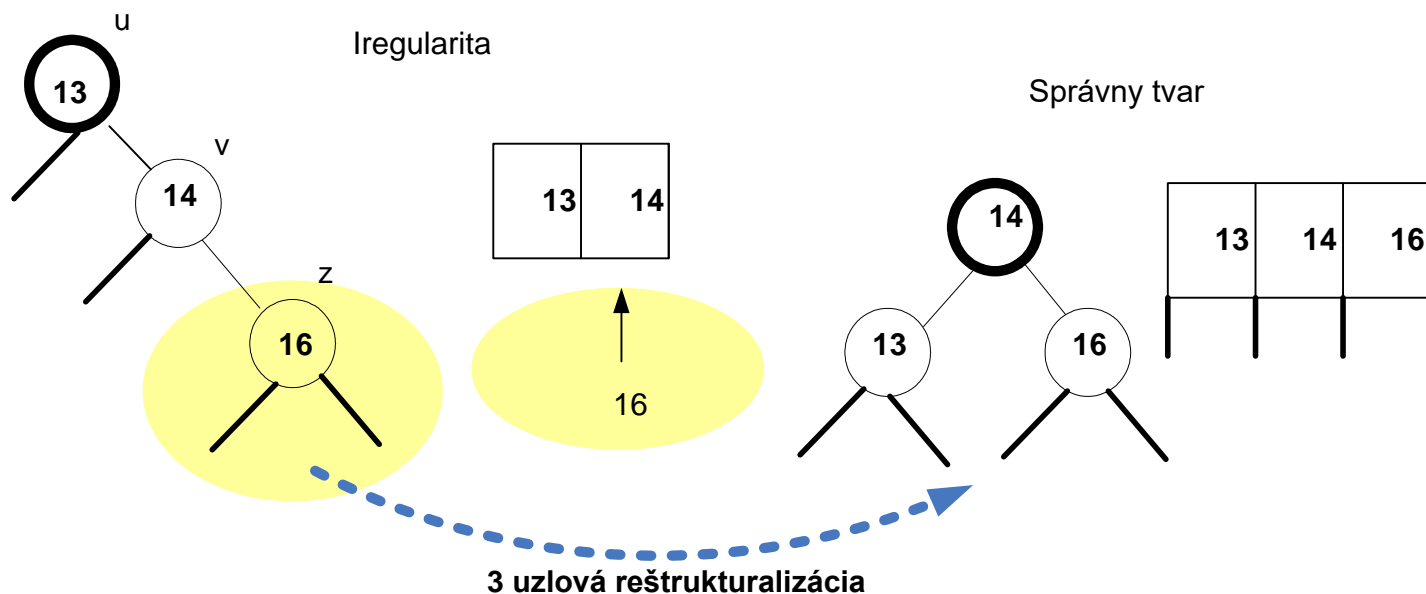
- nájdeme miesto pre vloženie algoritmom BVS,
- vytvoríme uzol-list z, vložíme záznam X
- ak z je koreň, zafarbíme ho na čierne, inak na červené

Môže nastať iregularita: z nie je koreň a jeho otec v je červený (dva červené uzly za sebou).

Pre ošetrenie tejto iregularity rozoznávame dva prípady:

Prípad 1. Brat w uzla v je čierny.

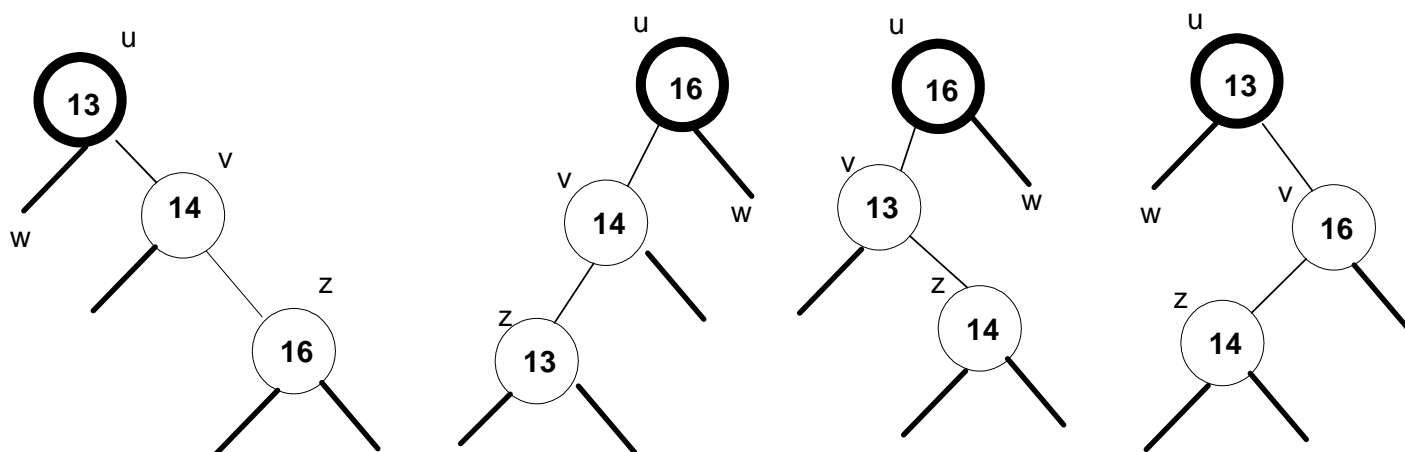
Ide o pokus vytvorenia nesprávneho 4-uzla v odpovedajúcom 2-4 strome. Situácia z príkladu hore:



Algoritmus reštrukturalizácie sa vykoná v štyroch príbuzných situáciách ukázaných na ďalšom obrázku.

Značenie:

- z ... vkladací
- v ... otec (z)
- u ... otec (v)
- w ... syn (u) brat (v)



Algoritmus reštrukturalizácie:

a.) rotácia

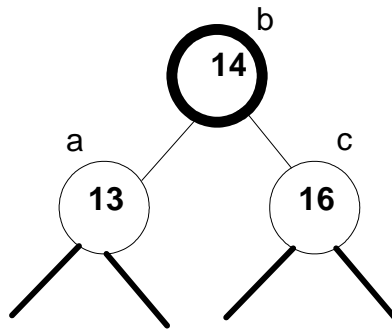
- uzly z, v a u dočasne označ písmenami a, b, c v poradí podľa veľkosti (odpovedajúco poradiu v prehliadke *inorder*)

- uzol u nahraď uzlom označeným b
- ustanov a ľavým a c pravým synom u , vrchol a (resp. c) prevezme ľavý (resp. pravý) podstrom uzla b tak, aby uzly ostali v poradí *inorder* (podobne ako pri jednoduchéj rotácii)
- ide o rovnakú jednoduchú rotáciu ako pri AVL strome (na obrázku stačí v prípade 1 a 2 vykonať jednoduchú rotáciu okolo vrcholu 14, v prípade 3 a 4 sa najskôr vykoná jednoduchá rotácia okolo vrcholu 13, resp. 16, čím sa získa situácia 1 alebo 2).

b.) farbenie

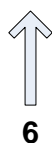
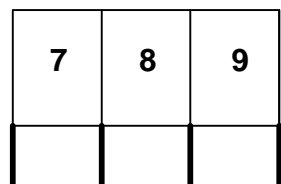
- zafarbi b na čierne, zafarbi a, c na červené

Výsledok:

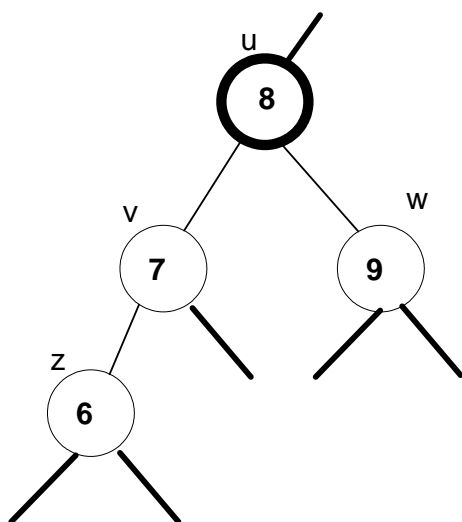


Prípád 2. Brat w uzla v je červený.

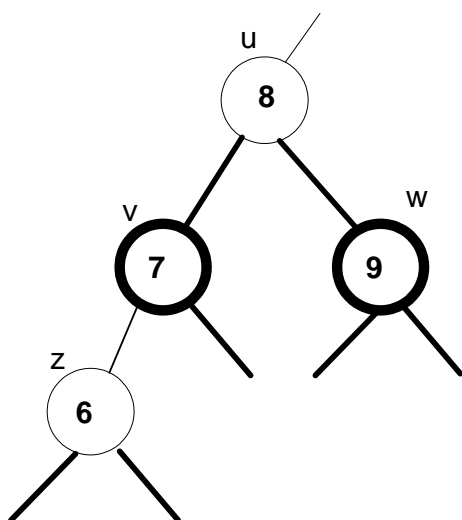
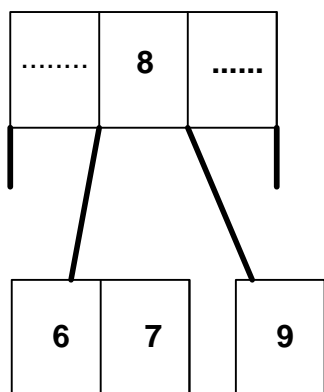
Tento prípad predstavuje snahu vložiť záznam do 4-uzla v 2-4 stromu (preplnenie uzla). Prípad sa rieši operáciou *prefarbenia* uzlov bez rekonštrukcie uzlov.



vkladaný



Neregulárna situácia
po vložení
(dva červené)



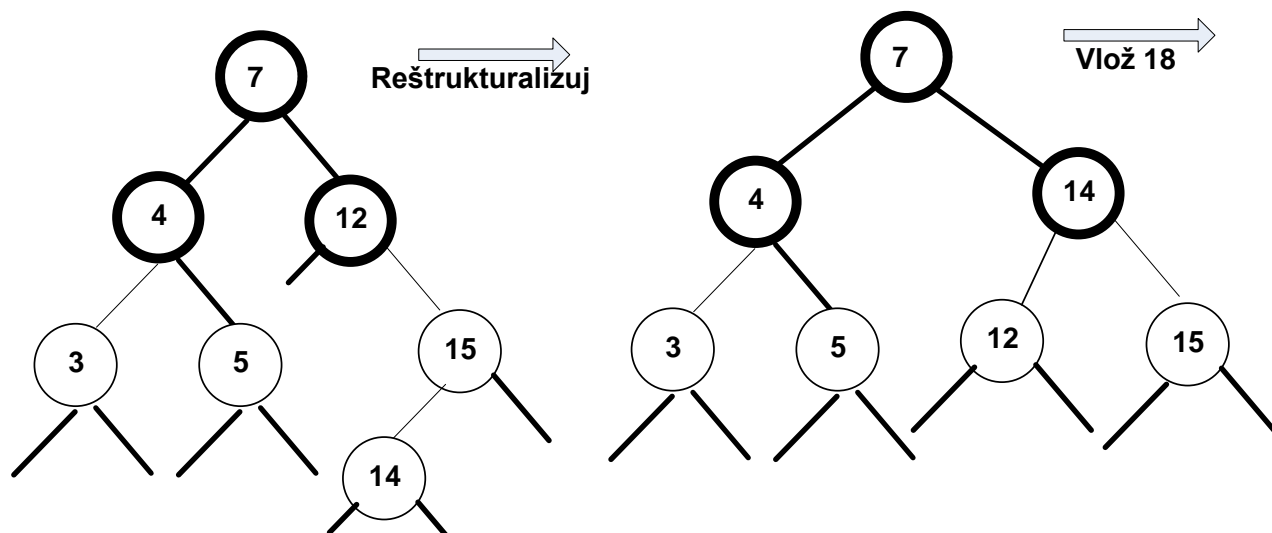
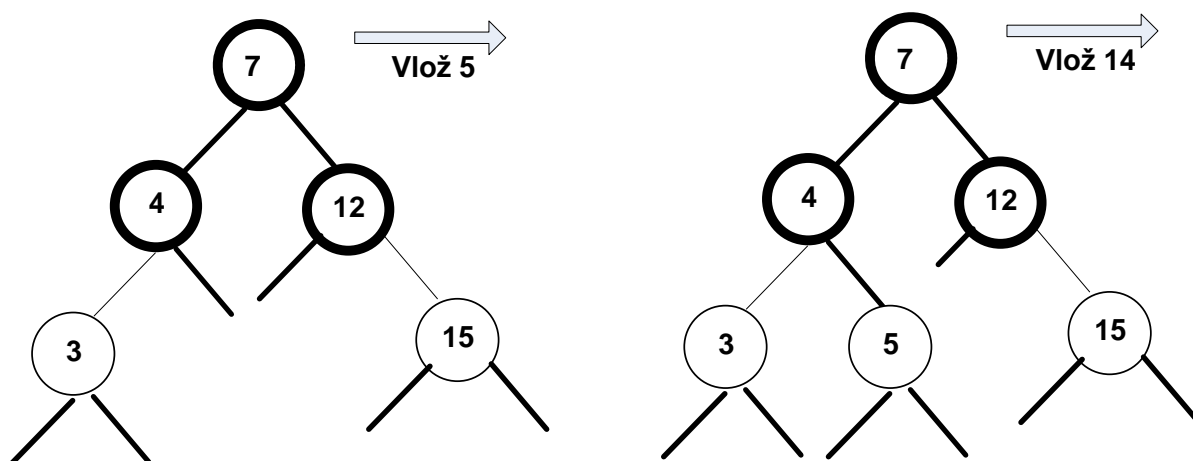
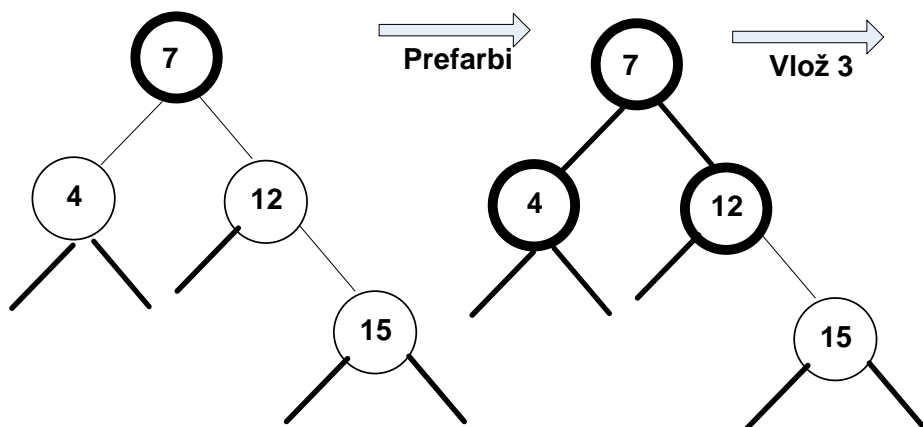
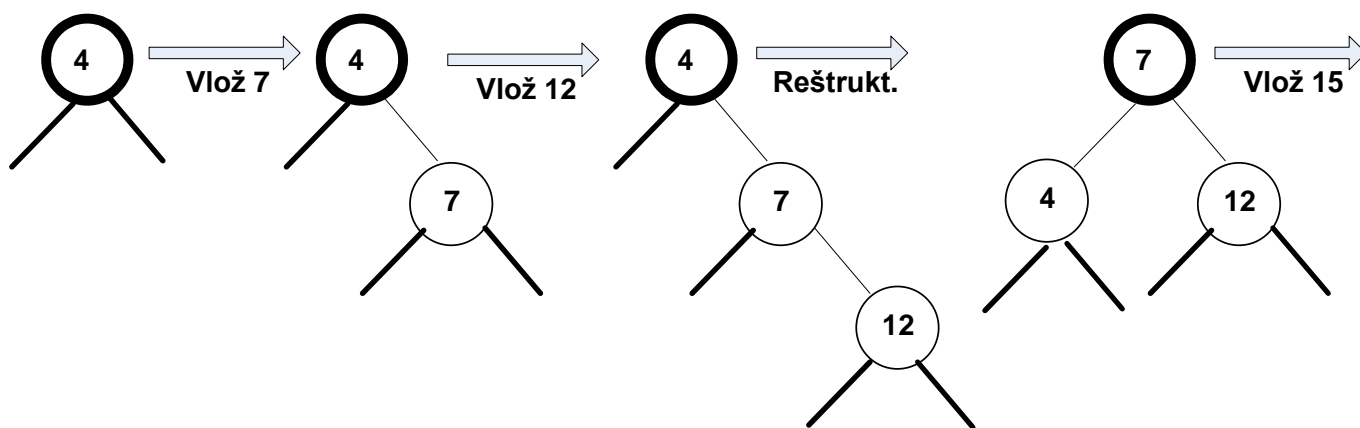
Situácia po
prefarbení

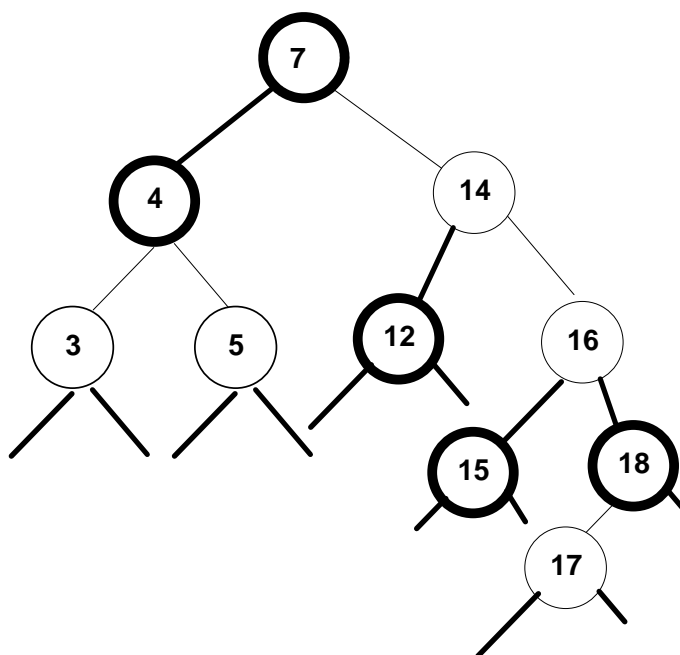
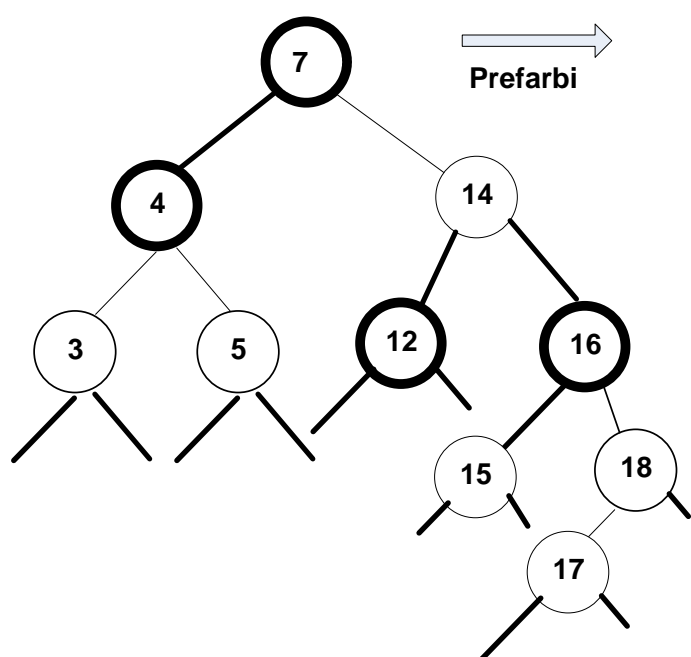
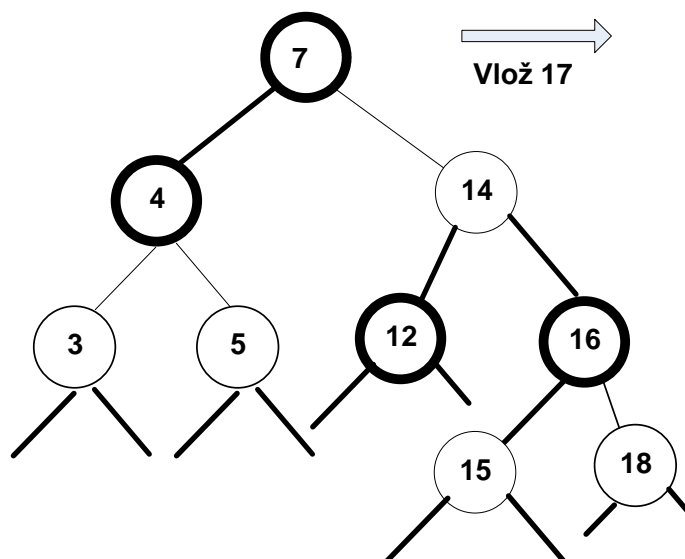
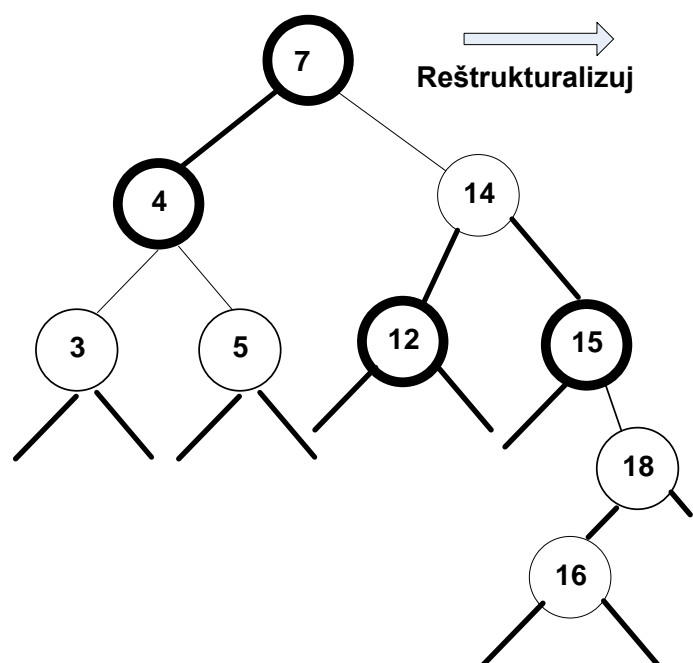
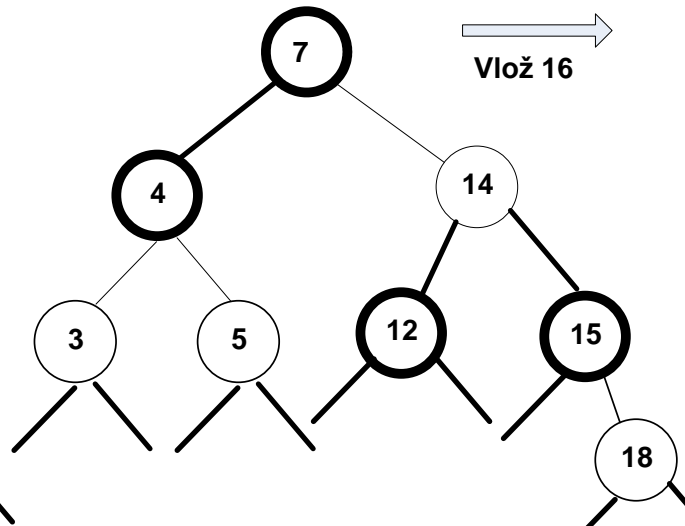
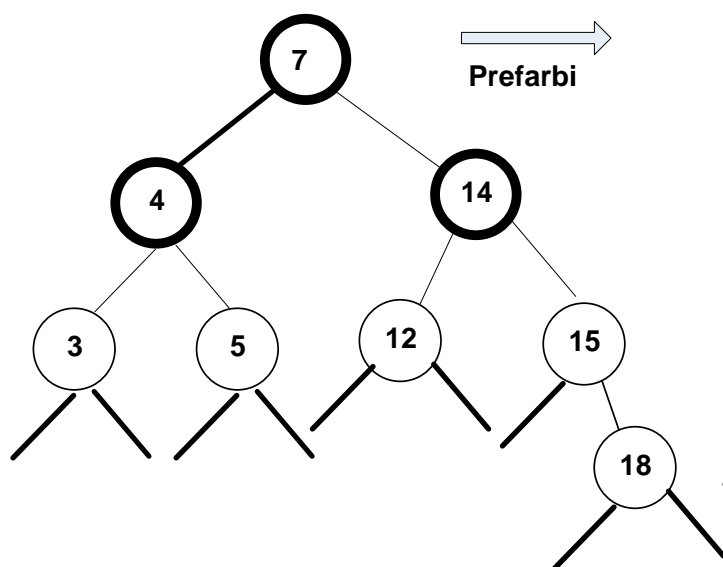
Algoritmus prefarbenia:

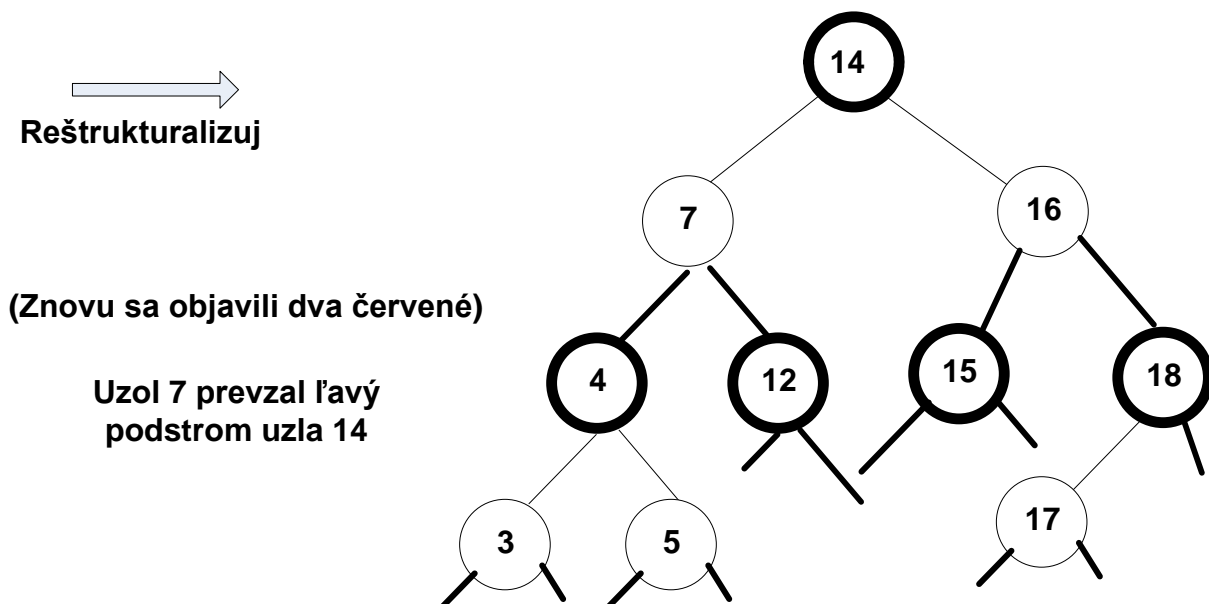
- zafarbi v a w na čierno;
- ak u nie je koreň, zafarbi ho na červenou, inak na čierno.

Po prefarbení treba preskúmať, či sa problém dvoch červených neobjaví vyššie. Ak áno, treba preskúmať, ktorý je to prípad a riešiť ho buď reštrukturalizáciou alebo prefarbením.

Takto sa pokračuje rekurzívne smerom nahor až do vyriešenia problému dvoch červených.







Odoberanie z RB- stromu:

Prvok vymažeme rovnako ako v binárnom vyhľadávacom strome. Jeden z listov je teda odstránený. Ak bol odstránený vrchol červený, tak je mazanie ukončené (ekvivalent situácie mazania v 2-4 strome, kedy sa z 4-vrcholu stane 3-vrchol, alebo z 3-vrcholu stane 2-vrchol).

Ak odstránený list stromu je čierny na jeho mieste si predstavíme **dvojité čierny** (“double black”) vrchol. Odstránený vrchol je akoby nahradený špeciálnym vrcholom s dvojitou čiernou farbou. Vymazaním čierneho listu sa nám totiž narušila dôležitá vlastnosť RB-stromu, rovnaký počet čiernych vrcholov od koreňa do každého z listov. V 2-4 strome ide o situáciu, kedy sa z 2-vrcholu odstráni jediný kľúč a tým vzniká prázdny vrchol.

Ak predpokladáme, že odstránený prvok je ľavý syn svojho otca, tak môžu nastať 4 prípady (analogicky situáciu riešime na opačnú stranu):

1. Brat odstráneného vrcholu je čierny a obaja jeho synovia sú tiež čierny. Situáciu riešime prefarbením brata odstráneného prvku na červeno a ich otec dostane čiernu farbu. Ak otec bol pôvodne čierny, tak si ho označíme ako dvojité čierny, čím sme presunuli problém vyššie. Opätovne riešime situáciu rovnako ako po odstránení prvku z listu (identifikácia farby brata a jeho potomkov a následný výber krokov).

2. Brat odstráneného vrcholu je čierny a jeho pravý syn je červený. Situáciu riešime prefarbením červeného syna na čierneho, prefarbením brata mazaného prvku farbou ich otca, prefarbením otca na čierneho a ľavou rotáciou okolo otca mazaného prvku. Mazanie končí.
3. Brat odstráneného vrcholu a jeho pravý syn sú čierny, ľavý syn brata je červený. Situácia sa rieši prefarbením, kedy brat dostane červenú farbu, jeho červený syn sa prefarbí na čierne. Potom vykonáme pravú rotáciu okolo brata mazaného prvku. Následne riešime situáciu z kroku 2.
4. Brat odstráneného vrcholu je červený. Ich otec, ktorý je určite čierny sa prefarbí na červeno. Brat sa prefarbí na čierne. Následne aplikujeme ľavú rotáciu okolo otca. Opätovne riešime situáciu rovnako ako po odstránení prvku z listu (identifikácia farby brata a jeho potomkov a následný výber krokov).