8.* Navrhněte operaci Split(T, x), která zadaný (a, b)-strom T rozdělí na dva stromy. V jednom budou klíče menší než x, v druhém ty větší. Pokuste se o logaritmickou časovou složitost.

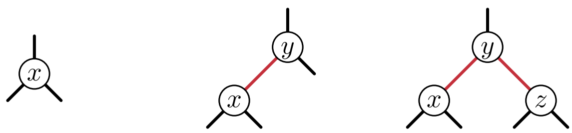
9. Nevýhodou (a, b)-stromů je, že plýtvají pamětí – může se stát, že vrcholy jsou zaplněné jen z poloviny. Navrhněte úpravu, která zaručí zaplnění z alespoň 2/3.

8.4* Červeno-černé stromy

Nyní se od obecných (a, b)-stromů vrátíme zpět ke stromům binárním. Ukážeme, jak překládat $(2, 4)$-stromy na binární stromy, čímž získáme další variantu BVS s logaritmickou hloubkou a poměrně jednoduchým vyvažováním. Říká se jí *červeno-černé stromy* (red-black trees, RB stromy). My si je předvedeme v trochu neobvyklé, ale příjemnější variantě navržené v roce 2008 Robertem Sedgewickem pod názvem left-leaning red-black trees (LLRB stromy).

Překlad bude fungovat tak, že každý vrchol $(2, 4)$-stromu nahradíme konfigurací jednoho nebo více binárních vrcholů. Aby bylo možné rekonstruovat původní $(2, 4)$-strom, rozlišíme dvě barvy hran: *červené hrany* budou spojovat vrcholy tvořící jednu konfiguraci, *černé hrany* povedou mezi konfiguracemi, čili to budou hrany původního $(2, 4)$-stromu. Barvu hrany si můžeme pamatovat například v jejím spodním vrcholu.

Strom přeložíme podle následujícího obrázku. Vrcholům $(2, 4)$-stromu budeme v závislosti na počtu synů říkat 2-vrcholy, 3-vrcholy a 4-vrcholy. 2-vrchol zůstane sám sebou. 3-vrchol nahradíme dvěma binárními vrcholy, přičemž červená hrana musí vždy vést doleva (to je ono LL v názvu LLRB stromů, obecné RB stromy nic takového nepožadují, což situaci později dost zkomplikuje). 4-vrchol nahradíme „třešničkou“ ze tří binárních vrcholů.



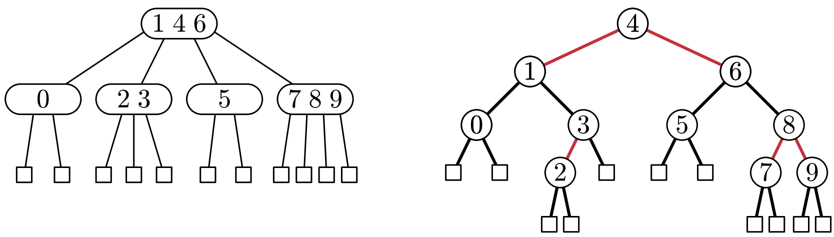
Pokud podle těchto pravidel transformujeme definici $(2, 4)$-stromu, vznikne následující definice LLRB stromu.

Definice: *LLRB strom* je binární vyhledávací strom s vnějšími vrcholy, jehož hrany jsou obarveny červeně a černě. Přitom platí následující axiomy:

1. Neexistují dvě červené hrany bezprostředně nad sebou.

2. Jestliže z vrcholu vede dolů jediná červená hrana, pak vede doleva.
3. Hrany do listů jsou vždy obarveny černě. (To se hodí, jelikož listy jsou pouze virtuální, takže do nich neumíme barvu hrany uložit.)
4. Na všech cestách z kořene do listu leží stejný počet černých hran.

Prvním dvěma axiomům budeme říkat červené, zbylým dvěma černé.



Obrázek 8.11: Překlad $(2,4)$-stromu na LLRB strom

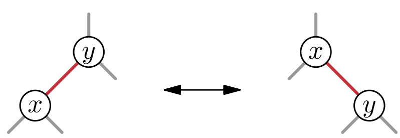
Pozorování: Z axiomů plyne, že každá konfigurace pospojovaná červenými hranami vypadá jedním z uvedených způsobů. Proto je každý LLRB strom překladem nějakého $(2,4)$-stromu.

Důsledek: Hloubka LLRB stromu s n klíči je $\Theta(\log n)$.

Důkaz: Hloubka $(2,4)$-stromu s n klíči činí $\Theta(\log n)$, překlad na LLRB strom počet hladin nesníží a nejvýše zdvojnásobí. □

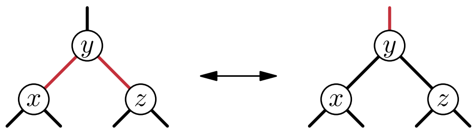
Vyvažovací operace

Operace s LLRB stromy se skládají ze dvou základních úprav. Tou první je opět rotace, ale používáme ji pouze pro červené hrany:



Rotace červené hrany zachovává nejen správné uspořádání klíčů ve vrcholech, ale i černé axiomy. Platnost červených axiomů záleží na barvách okolních hran, takže rotaci budeme muset používat opatrně. (Rotování černých hran se vyhýbáme, protože by navíc hrozilo porušení axiomu 4.)

Dále budeme používat ještě *přebarvení 4-vrcholu.* Dvojici červených hran tvořících 4-vrchol přebarvíme na černou, a naopak černou hranu vedoucí do 4-vrcholu shora přebarvíme na červenou:



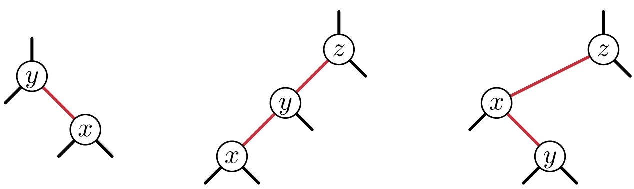
Tato úprava odpovídá rozštěpení 4-vrcholu na dva 2-vrcholy, přičemž prostřední klíč y přesouváme do nadřazeného k-vrcholu. Černé axiomy zůstanou zachovány, ale může dojít k porušení červených axiomů o patro výše.

Dodejme ještě, že přebarvení jde použít i v kořeni. Můžeme si představovat, že do kořene vede shora nějaká virtuální hrana, již můžeme bez porušení axiomů libovolně přebarvovat.

Vkládání štěpením shora dolů

Nyní popíšeme, jak se do LLRB stromu vkládá. Půjdeme na to asi takto: místo pro nový vrchol budeme hledat obvyklým způsobem, ale kdykoliv cestou potkáme 4-vrchol, rovnou ho rozštěpíme přebarvením. Až dorazíme do listu, připojíme místo něj nový vnitřní vrchol a hranu, po které jsme přišli, obarvíme červeně. Tím se nový klíč připojí k nadřazenému 2-vrcholu nebo 3-vrcholu. To zachovává černé axiomy, ale průběžně jsme porušovali ty červené, takže se budeme vracet zpět do kořene a rotacemi je opravovat.

Nyní podrobněji. Během hledání sestupujeme z kořene dolů a udržujeme invariant, že aktuální vrchol není 4-vrchol. Jakmile na nějaký 4-vrchol narazíme, přebarvíme ho. Tím se rozštěpí na dva 2-vrcholy a prostřední klíč se stane součástí nadřazeného k-vrcholu. Víme ovšem, že to nebyl 4-vrchol, takže se z něj nyní stane 3-vrchol nebo 4-vrchol. Jen možná bude nekorektně zakódovaný: 3-vrchol ve tvaru pravé odbočky nebo 4-vrchol se dvěma červenými hranami nad sebou:



Nakonec nás hledání nového klíče dovede do listu, což je místo, kam bychom klíč chtěli vložit. Nad námi leží 2-vrchol nebo 3-vrchol. List změníme na vnitřní vrchol s novým klíčem, pod něj pověsíme dva nové listy připojené černými hranami, hranu z otce přebarvíme na červenou: