1 Binární stromy

Scéna: *Úvod*

Úvodní scéna je jen grafická ilustrace, ukazující binární strom o 500 vrcholech. Levý (pravý) syn libovolného vrcholu v a jeho potomci jsou zobrazeni vlevo (vpravo) od vrcholu v. Zkuste si ale přepnout mezi volbami [**Náhled: Kořenový**] a [**Náhled: Lineární**]. Při prvním se kořen umístí (horizontálně) do poloviny obrazovky, jeho synové do 1. a 3. čtvrtiny, jejich synové do 1., 3., 5. a 7. osminy obrazovky atd. V druhém případě jsou zase zleva doprava pravidelné rozestupy mezi x-ovými souřadnicemi vrcholů. Oba způsoby mají své výhody a nevýhody a oba budou používány, v mnoha případech bude mít uživatel volbu podobně jako v této scéně.

Scéna: *Binární strom*

Byl vytvořen binární vyhledávací strom (BVS) o 25 vrcholech. Podívejte na strom na displeji. Nejvyšší vrchol nazýváme *kořen* (strom tedy roste dolů, opačně než v přírodě). Každý vrchol může mít až dva *syny*, jeden z nich je *levý* a druhý je *pravý*; v případě, že vrchol má pouze jednoho syna, je určeno, zda je levý nebo pravý. Vrchol, který nemá žádného syna, se nazývá *list.* Je-li vrchol v synem vrcholu u, pak řekneme, že u je *otcem* vrcholu v. Každý vrchol stromu s výjimkou kořene má právě jednoho otce.

Jestliže existuje posloupnost vrcholů $v_0, \ldots, v_k$ taková, že pro $i = 1, \ldots, k$ je v_i synem vrcholu v_{i-1}, pak řekneme, že v_k je *potomkem* v_0 a naopak v_0 je *předkem* v_k. *Podstrom* určený vrcholem u je množina obsahující u a všechny jeho potomky.

Klepněte myší na libovolný vrchol stromu; vybraný vrchol zrůžoví, vrcholy podstromu určeného vybraným vrcholem zůstanou tmavé, ale ostatní vrcholy vyblednou.

I v této scéně je možno přepínat mezi dvěma způsoby kreslení stromu - kořenovým a lineárním. Je možné si nechat vytvořit nový strom, který bude mít tolik vrcholů, kolik je číslo v poli u označení [**Kolik vrcholů**].

Scéna: *Binární vyhledávací strom*

BVS se používá pro uložení množiny čísel, se kterou můžeme provádět následující operace:

- nalezení prvku - je dáno číslo a má se zjistit, zda se v množině nachází a v kladném případě se také určí kde je uloženo (viz dále)
- vložení prvku - do množiny se vloží zadané číslo (pokud ovšem v množině již je, neprovede se nic)
- vynechání prvku - z množiny se vynechá číslo, zadané místem kde je uloženo (viz dále) - tím je zaručeno, že se v množině opravdu nachází; pokud nevíme, zda číslo, které chceme vynechat, v množině je a nebo kde je uloženo, musíme provést operaci nalezení prvku
- minimum, maximum - naleznou nejmenší, resp. největší číslo, které v množině je

K těmto operacím je ještě třeba přidat operaci inicializace, kterou se vytvoří strom, popisující prázdnou množinu. Tuto operaci je třeba zavolat předtím, než se provádí jakákoli operace s množinou. Později inicializaci můžeme volat vždy, kdy chceme množinu vyprázdnit, t.j. najednou z ní vychechat vše, co se v ní nachází (při vytvoření stromu znázorněného na obrazovce byla inicializace provedena automaticky).

Množinu $\mathcal{M}$ obsahující N čísel uložíme pomocí BVS tak, že vytvoříme *libovolný* binární strom s N vrcholy a pak do každého vrcholu uložíme jeden prvek množiny $\mathcal{M}$, zvaný *klíč*, a to tak, aby platilo pravidlo, uvedené v následující scéně. V této scéně je pouze ukázáno, jak klíče vrcholů znázorňujeme graficky: pod každým vrcholem je nakreslen sloupec, který svou výškou odpovídá velikosti klíče vrcholu. Klepněte na libovolný vrchol; zrůžoví vybraný vrchol a zároveň sloupec znázorňující velikost jeho klíče. V poli hodnot se také objeví modrá vodorovná čára, umožňující porovnávat snadno zvolený klíč s klíči ostatních vrcholů.

Do vrcholu se také často ukládají někdy velice komplexní data, která jsou jednoznačně označena stanoveným identifikačním číslem, které pak slouží jako klíč vrcholu.

Velikost znázorněného BVS i jeho způsob nakreslení lze měnit stejně jako v předchozí scéně. Navíc je možno zvolit, zda bude zobrazováno okénko se sloupci znázorňujícími velikost klíčů ve vrcholech.

Scéna: *Binární vyhledávací strom - podmínka*

Klíče, popsané v předchozí scéně, musí být do vrcholů binárního stromu uloženy tak, aby platilo následující pravidlo, které zaručuje, že se ve stromu snadno nalezne hledaný klíč:

Pro libovolný vrchol u platí, že

- má-li u levého potomka v, pak číslo uložené ve v nebo libovolném vrcholu podstromu určeného vrcholem v je *menší* než číslo uložené v u a

- má-li u pravého potomka w, pak číslo uložené ve w nebo libovolném vrcholu podstromu určeného vrcholem w je *větší* než číslo uložené v u.

Zde levý potomek označuje levého syna nebo libovolného potomka levého syna, obdobně pravý potomek je pravý syn nebo jeho libovolný potomek.

Zkontrolujte si platnost podmínky pro uvedený strom: Při klepnutí na libovolný vrchol tento vrchol zrůžoví, vrcholy jeho levého podstromu zmodrají a v pravém podstromu zezelenají. Ostatní vrcholy vyblednou. Podmínka stanoví, že klíče modrých vrcholů musí být menší než klíč růžového zvoleného vrcholu, klíče zelených vrcholů musí být větší než klíč růžového vrcholu. Při našem způsobu kreslení stromu podmínka znamená, že probíráme-li vrcholy zleva doprava, jejich klíče rostou.

Někdy se připouští, aby dva různé vrcholy měly stejný klíč, my to však pro jednoduchost výkladu připouštět nebudeme.

Scéna: *Vyhledávání v BVS*

Nyní se podíváme, jak zjistit, zda dané číslo K je klíčem některého z vrcholů stromu a v kladném případě tento vrchol určit.

Hledání začíná v kořeni stromu. Porovnáme hledané číslo s klíčem kořene. Jsou-li si rovny, hledaný vrchol byl nalezen. Tento případ je však velmi řídký. Pokud se čísla nerovnají, pak pravidlo pro rozmisťování klíčů v BVS říká, že je-li hledané číslo menší než klíč vrcholu, musíme přejít do levého syna kořene a hledat v jeho podstromu, je-li větší, musíme hledat v podstromu určeném pravým synem.

Hledání pak provádíme opakováním operace, která byla provedena v kořeni, tedy je-li klíč aktuálního vrcholu stejný jako hledané číslo, byl hledaný vrchol nalezen, jinak přejdeme do levého nebo pravého syna, pokud je hledané číslo menší, resp. větší než klíč vrcholu.

Může se ovšem stát, že bychom chtěli jít do levého (nebo pravého) syna a ten ve stromě není. To znamená, že se hledané číslo v množině popsané stromem nenachází, protože jinde než v chybějícím synu by být nemohlo.

Zkuste si nyní hledání čísla ve stromu jako klíče některého vrcholu: hledané číslo se zadá některým z následujícícho způsobů:

- buď přímo numericky do pole vpravo od návěští [**Klíč**] na ovladači, nezapomeňte nakonec zmačknout klávesu [**ENTER**], aby se hodnota zaznamenala v počítači

- nebo klepnutím na některý vrchol - tím se jako číslo zvolí klíč poklepaného vrcholu

- nebo klepnutím do pole sloupečků pod stromem - hodnota je dána výškou bodu, zadaného klepnutím (omluva - tato funkce momentálně nefunguje)

- nebo klepnutím na [**Dotaz**] - počítač vybere dotazované číslo náhodně.

Ve všech čtyřech případech se zadaná hodnota zobrazí numericky na ovladači a graficky vodorovnou čárou v poli sloupců ve výšce úměrné zvolené hodnotě.

Operaci můžeme provést najednou stisknutím knoflíku [**Proveď**] a nebo ji krokovat stisknutím knoflíku [**Krokuj**]. V druhém případě se aplet přepne do animačního módu a průběh operace vyhledávání se sleduje s využitím knoflíků [**Krok**] a [**Animuj**]. První operaci krokuje po jednotlivých vrcholech, druhý ji provede jako plynulou animovanou akci. Dva knoflíky [**Zpět**] umožňují se vracet zpět po krocích nebo najednou až na počátek. Z animačního módu se dostaneme zpět do dotazovacího módu pomocí knoflíku [**Proveď**] nebo [**Zruš**]. První operaci provede, druhý ji zruší. Jelikož při vyhledávání se strom nemění, je výsledek obou operací stejný, ale ve scénách přidávání nebo odebírání vrcholu budou mít funkci odlišnou.

Při hledání se do kořene umístí zelený žeton, který se pohybuje stromem a ukazuje kořen podstromu, ve kterém by se mohl skrývat hledaný klíč. Vrcholy nepatřící do tohoto podstromu (které určitě hledaný klíč neobsahují) vyblednou. Je-li nalezen hledaný klíč, barva žetonu se změní na červeno-hnědou. V případě, že klíč nebyl nalezen, se žeton zbarví fialově a stranou od něj se zobrazí otazník, a to v místě, kde by měl být, ale není, další vrchol při hledání klíče.

Scéna: *Hledání minima v BVS*

Hledání minima je zřejmé a velmi jednoduché; z kořene postupujeme doleva dokud to jde. V okamžiku kdy se dostaneme do vrcholu, kterému schází levý syn, nalezli jsme vrchol s minimálním klíčem, způsob krokování viz výše.

Scéna: *Hledání maxima v BVS*

Hledání maxima je prakticky totožné s hledáním minima, pouze postupujeme stále doprava.

Scéna: *Přidávání klíče do BVS*

Úvodní část této operace je stejná jako vyhledávání. Pokud se zjistí, že přidávaný klíč už ve stromu je, nic dalšího se neprovede. Pokud vyhledávání zjistí, že klíč ve stromu není, identifikuje zároveň místo, kde by se měl nacházet (chybějící syn některého vrcholu). Pak se chybějící syn vytvoří a přidá a uloží se do něj přidávaný klíč.

Zadejte klíč podobně jako u operace vyhledávání, ale tak, aby klíč v množině nebyl (tedy např. při zadání klepnutím na vrchol pak číslo uvedené na ovladači pozměňte). Krokování provádí výše uvedeným způsobem.

Scéna: *Vynechávání vrcholu z BVS*

Operace vynechávání má dvě varianty podle toho, jak specifikujeme vynechávaný prvek. Jestliže totiž vypustíme z množiny jedno číslo (klíč), pak musíme vynechat i některý vrchol; obvykle je to vrchol ve kterém byl vynechávaný klíč uložen, ale v některých případech postupujeme jinak.

Pokud je zadáno číslo které se má vynechat, aniž by bylo určeno v kterém vrcholu leží, pak napřed zavoláme operaci vyhledávání a příslušný vrchol určíme (může se stát, že zjistíme, že dané číslo ve skutečnosti není klíčem žádného vrcholu; pak pochopitelně neprovedeme nic).

Nyní se tedy budeme věnovat otázce, jak vynechat zadaný vrchol (i s jeho klíčem, který tím vypadne z representované množiny). Jsou tři možnosti, které budou postupně probrány v rámci této scény. Zvolte si vrchol klepnutím podle toho, jaký případ chcete ukázat. (Pokud by vhodný vrchol ve stromu nebyl, což se stává někdy je-li požadován vrchol s jedním synem a vyjímečně i pokud je požadován vrchol s oběma syny, nechte si vytvořit nový strom a zkuste to znovu.)

Případ 0: vynechání listu

Nejjednodušší je vynechání listu, tedy vrcholu bez synů. Ten prostě odtrhneme i s klíčem. Pokud vrchol měl otce, v závislosti na tom, zda byl levým či pravým synem svého otce, otci vynulujeme příslušný ukazatel na syna. Pokud vrchol neměl otce (byl tedy kořenem, což znamená, že vynechávaný vrchol byl jediným vrcholem stromu), bude po vynechání strom prázdný, což obvykle znamená, že je nutno upravit referenci na strom (kterou bývá ukazatel na nyní neexistující kořen).

Případ 1: vynechání vrcholu s jedním synem

Vynechávání vrcholu s jedním synem (buď levým nebo pravým) je o něco složitější. Je-li vynechávaný v vrchol kořenem (nulový otec), pak se vrchol v odtrhne a jeho syn (jediný) se stane novým kořenem stromu. V opačném případě se vrchol v také odtrhne, ale jeho jediný syn je jako syn adoptován otcem vrcholu v (levým nebo pravým podle toho, zda v byl levým nebo pravým vrcholem).

Případ 2: vynechání vrcholu s 2 syny

Nakonec nejsložitější je vynechání vrcholu v se dvěma syny. Takový vrchol nelze přímo vynechat, jeho dva syny by nebylo možno na jeho otce napojit. Proto postupujeme jinak: z vrcholu v napřed odstraníme klíč, potom nalezneme vrchol w, jehož klíč je nejblíže nižší vynechanému klíči a klíč z w přesuneme do v a nakonec vynecháme vrchol w.

Volba nejblíže nižšího klíče je diktována tím, aby přesunem klíče z w do v se neporušila podmínka o umístění klíčů v BVS (bylo by také možno volit klíč nejblíže vyšší).

Snadno se nahlédně, že vrchol s nejblíže nižším klíčem se nalezne tak, že se z v přejde do jeho levého syna a pak se opakovaně postupuje do pravého syna, dokud existuje.

Ze způsobu hledání vrcholu s nejblíže nižším klíčem také plyne, že nalezený vrchol w nebude mít pravého syna (a možná ani levého), takže jej již umíme vynechat jednou z výše uvedených operací, popsaných v Případu 0 nebo Případu 1.

Scéna: *Vynechávání klíče z BVS*

Pro úplnost si ještě ukážeme vynechání klíče, který ale není zadán volbou vrcholu ve kterém leží, ale číselnou hodnotou; zadávání dotazu je obdobné jako při přidávání. Je to vlastně kombinace hledání klíče a vynechávání klíče daného polohou.

Scéna: *Operace s BVS*

V této scéně je možno si vyzkoušet všechny operace s binárním vyhledávacím stromem; stačí zvolit si na ovladači operaci a výše uvedeným způsobem ji provést nebo krokovat. Vrchol pro vynechávání je nyní nutno zvolit klepnutím myší.

Scéna: *Hloubka náhodného BVS*

Libovolná z výše uvedených operací se provádí podél cesty, která vede z kořene do některého z vrcholů (například nalezený vrchol u vyhledávání, minima a maxima, přidaný vrchol u přidávání) nebo mezi některým vrcholem a jeho potomkem (otec a syn vynechávaného listu nebo vrcholu s jedním synem nebo cesta z vrcholu do maxima jeho levého podstromu při vynechávání vrcholu s dvěma syny). Na každé hladině přitom provedeme pouze několik málo operací, jejichž počet je omezen konstantou nezávislou na velikosti stromu. Proto je doba potřebná k provedení libovolné z výše popsaných operací i v nejhorším případě úměrná hloubce stromu, t.j. délce nejdelší cesty z kořene do nějakého listu.

Hloubka BVS může být velmi malá ve srovnání s počtem jeho vrcholů. V optimálním případě, kdy s vyjímkou nejnižší vrstvy vrcholů má každý vrchol 2 syny, je v nejvyšší vrstvě (budeme ji označovat jako nultou) kořen, tedy 1 vrchol, v první vrstvě 2 vrcholy a v každé další vrstvě dvojnásobek vrcholů vzhledem k předchozí vrstvě, tedy v j-té vrstvě je 2^j vrcholů a má-li strom k vrstev, pak obsahuje $1 + 2 + 4 + \cdots + 2^k = 2^{k+1} - 1$ vrcholů, tedy jeho hloubka k je rovna $\lceil \log_2 n + 1 \rceil - 1 \approx \log_2 n$.

Jesliže prvky přidáváme do binárního vyhledávacího stromu náhodně (například u appletu náhodně se stejnou pravděpodobností z jistého velkého intervalu celých nezáporných čísel), pak hloubka stromu není o mnoho větší, než u optimálního stromu (dá se dokázat, že je nejčastěji okolo $1,44 \log_2 n$).

Knoflíkem [**Proveď**] přidávejte do původně prázdného stromu najednou větší množství náhodně a rovnoměrně generovaných klíčů. Počet pro jednu dávku je nastaven na 500, ale je možné ho na ovladači změnit. Je vidět, že strom je obvykle poměrně dobře vyvážený.

Hodnoty v pravém horním rohu znamenají: N celkový počet vrcholů, $maxH$ hloubku stromu (délka *nejdelší* cesty z kořene do listu) a $aveH$ průměrnou hloubku vrcholu (tedy aritmetický průměr délek cest z kořene do jednotlivých vrcholů - všech, ne jen listů). Je patrno, že i velmi velké stromy mají malou hloubku. Nakonec $\log_2 N$ znamená dvojkový logaritmus čísla N pro ilustraci, že poměr maximální i průměrné hloubky náhodně vytvářeného stromu k $\log_2 N$ zůstává přibližně stejný (a malý).

Je také možné změnit způsob kreslení stromu přepnutím mezi [**Náhled: Lineární**] a [**Náhled: Kořenový**]. Druhý z nich pěkně ukazuje, že některé větve stromu bývají zakrnělé, ale nevyskytují se větve příliš přerostlé.

Scéna: *Hloubka náhodného BVS (pokračování)*

Tato scéna je prakticky stejná jako předchozí, vrcholy jsou přidávány po 100, ale klíče jsou generovány jiným způsobem: klíč je vybrán náhodně a rovnoměrně v rozmezí od 0 do B, kde B pozvolna roste. Samotný postup není příliš důležitý, je volen tak, aby vytvářel zajímavě vypadající obrázky, ale podstatné je, že se *střední velikost* generovaných klíčů postupně *zvětšuje*.

Knoflíkem [**Proveď**] opětně přidávejte vrcholy a sledujte tvar stromu. Ze začátku ještě strom vypadá celkem vyváženě, pak ale začne jedna větev výrazně růst na úkor druhých, takže maximální i střední hloubka vrcholu, udané čísly $maxH$ a $aveH$ jsou výrazně větší než v předchozí scéně pro srovnatelný počet vrcholů N.

Scéna: *Vyhledávání v nevyváženém BVS*

V této scéně pouze zkusíme krok po kroku vyhledat náhodně zvolený klíč vrcholu ve stromu s 2000 vrcholy, který byl generován způsobem popsaným v předchozí scéně. Vyhledávání je obvykle tak zdlouhavé, že budete-li postupovat po jednotlivých krocích, patrně je ani nedokončíte.

Scéna: *Rotace*

V předchozích třech scénách jsme viděli, že tvar binárního vyhledávacího stromu může zdegenerovat tak, že operace s ním budou probíhat velmi pomalu. Proto se zavádí různé způsoby vyvažování, které udržují tvar binárního stromu v přijatelných mezích a proto i v nejhorším případě operace s nimi probíhají velmi rychle. Základní operací, kterou se vyvažují binární stromy (AVL-stromy a červeno-černé stromy, o kterých bude řeč později) je *rotace*.

Klepněte na některou hranu stromu na displeji. Zvolená hrana se "překlopí" neboli "rotuje" a tím se změní tvar stromu. Opětovným klepnutím na překlopenou hranu ji překlopíme do původní polohy - rotace je vratná operace.

Velmi doporučuji nastavit rychlost animace tak, aby bylo možno sledovat, co se při rotaci děje. Provede se to změnou čísla vedle návěští [**Kroky**]. Asi předtím budete muset zvolit [**Více ovladačů**]. Číslo udává, kolik snímků se zobrazí na jeden krok animace - tedy čím vyšší, tím pomaleji se operace zobrazuje.

Důležité je, že při rotaci se vrcholy posouvají pouze vertikálně, takže zůstává v platnosti podmínka na rozmístění klíčů v BVS, kterou je možno formulovat tak, že procházíme-li vrcholy zleva doprava (podle x-ové souřadnice), jejich klíče rostou.

Povšimněte si také, že operace je vratná. Druhé klepnutí na rotovanou hranu vrátí tvar stromu do výchozí polohy.

Dobré pochopení rotace je podstatné pro pochopení funkce AVL-stromů i červeno-černých stromů, proto tuto a následující tři scény neopouštějte brzo, ale pohrajte si se stromem. Zkuste například dostat rotacemi vrchol z nejnižší hladiny nahoru, aby se stal kořenem, nebo naopak spustit kořen až dolů, aby se stal listem.

Scéna: *Rotační schéma*

Tento obrázek, který můžete spatřit ve všech učebnicích, které o BVS pojednávají, ukazuje rotaci schematicky. Klepejte na knoflík [**Rotace**]. Hrana uprostřed se překlápí; z jejího horního konce vychází kromě rotující hrany ještě vazba na druhého syna horního vrcholu a jeho podstrom (naznačený jako trojůhelníková oblast) a z dolního konce vycházejí hrany ke dvěma synům a jejich podstromům. Povšimněte si, že při překlopení trojúhelník představující levý podstrom se pohybuje vertikálně opačným směrem než pravý podstrom a podstrom uprostřed zůstává ve stejné výšce. Tento nepohyblivý strom nám v některých případech zkomplikuje návrh vyvažovacích operací.

Scéna: *Rotace ve velkém stromu*

Scéna ukazuje totéž jako rotační schéma z předchozí scény, ale ne schematicky, nýbrž na příkladě velkého stromu. Klepněte na libovolnou hranu; pro názornost je lepší volit dlouhou hranu nahoře blízko kořene. Je dobře vidět v opačných směrech vertikálně se pohybující podstromy a nehybný podstrom mezi nimi.

Scéna: *AVL strom*

Aby se předešlo vzniku nevyvážených BVS, ve kterých provádění operací trvá velmi dlouho, přidávání a vynechávání vrcholů se upravuje různými způsoby, které zabraňují vzniku výrazných nevyvážeností. Jednou z těchto úprav jsou AVL stromy (pojmenované podle autorů metody, Adelson-Velského a Landise).

AVL-strom je binární vyhledávací strom, který pro každý svůj vrchol v splňuje následující podmínku:
výška levého podstromu vrcholu v se od výšky jeho pravého podstromu liší nejvýše o 1.

Vrcholy AVL-stromu budeme znázorňovat jako vahadélka. Jestliže pro jistý vrchol je jeho levý podstrom (t.zn. podstrom určený jeho levým synem) stejně hluboký jako jeho pravý podstrom, je vahadélko vodorovné. Podobně je vodorovné, pokud vrchol postrádá jak levého, tak i pravého syna. V opačném případě se vahadélko nakloní na stranu hlubšího podstromu.

Ovladač je na počátku nastaven na přidávání vrcholů po jednom: knoflíkem [**Další vrchol**] přidávejte vrcholy a sledujte, že strom stále splňuje AVL pravidlo. Změnou čísla v poli [**Kolik vrcholů**] se nastaví počet v jednom kroku přidaných vrcholů. Co se při přidávání děje bude vysvětleno později.

Zkuste si strom vymazat knoflíkem [**Zruš strom**] a změnit způsob náhodné volby přidávaných čísel na [**Posunující se distribuci**], která má devastující účinky u prostého BVS. U AVL stromu ke vzniku nevyváženosti nedochází, vyváženost stromu je lepší, než byla u stejnoměrně náhodného rozložení vyváženost BVS.

Můžete též strom vymazat znovu a změnit způsob náhodné volby přidávaných čísel na [**Monotónní distribuci**]. Ta přidává čísla z rostoucí řady $0, 1, 2, \ldots$. U prostého BVS by to vedlo k degeneraci stromu na jedinou větev, hloubka stromu by byla největší možná (max. hloubka vrcholu $N-1$, průměrná $(N-1)/2$). Ani zde nedojde u AVL stromu ke vzniku podstatnější nevyváženosti. Je také dobře vidět, jak vrcholy se stále objevují na pravé straně, ale jakmile se pravá větev trochu prodlouží, vrcholy se "přelijí" nebo "překlopí" nalevo a strom se vyváží.

Scéna: *Rotace v AVL stromu*

Tato scéna je opakováním scény s rotací hran, ale strom je zobrazován jako AVL strom s vahadélkovými vrcholy.

Znázorněný strom ale nemusí a obvykle není AVL-strom; AVL-podmínka není splněna. Vrcholy, které ji porušují, jsou také nakloněny na stranu hlubšího podstromu, ale navíc ještě červeně obroubeny. Pokud takové vrcholy nejsou přítomny, vygenerujte si nový strom.

Zkuste si opět provádět rotace. Rotacemi lze dosáhnout takřka libovolných změn tvaru stromu. Zůstaňte u této scény nejméně tak dlouho, dokud strom nepřeformujete tak, aby to byl AVL-strom, tedy aby neměl žádný červeně obroubený vrchol. Uvidíte, že to poměrně snadné, ale nikoliv triviální.

Pokud červeně obroubené vrcholy odstaníte, zkuste naopak AVL-podmínku hodně pokazit, například větším množstvím náhodně vybraných rotací, takže hodně vrcholů má červený okraj. Rotace je vratná, dá se jí tvar stromu zkazit, ale pak zase napravit. AVL algoritmus pochopitelně používá rotace uvážlivě, aby vyváženost zachovával, nikoli kazil.

Pak ještě znovu silně nevyvážené vrcholy "vyrotujte" pryč. Celý postup třeba několikrát zopakujte, dokud Vám nebude zcela jasné, co to je AVL-strom a jak ho získat rotacemi.

Scéna: *Přidávání do AVL-stromu*

Po seznámení se s rotacemi se již můžeme pustit do operací AVL stromu. Jelikož vyhledávání a určování minima a maxima nemění tvar stromu, provádějí se stejně jako v BVS. Operace které ale tvar mění a obecně mohou porušit podmínku AVL jsou přidávání a vynechávání klíče nebo vrcholu. V této scéně si probereme operaci přidávání.

Přidávání může probíhat 3 způsoby, které budou v této scéně probrány všechny. Napřed budeme krokovat operaci, která probíhá prvním způsobem; po jejím ukončení se situace (volba přidávaného čísla) automaticky nastaví na druhý způsob a pak na třetí.

Ve všech třech případech se nejprve přidá do stromu nový vrchol se zadaným klíčem úplně stejně jako u BVS stromu. Poté postupujeme jedním z následujících tří způsobů:

Případ 1:

Po přidání nového vrcholu BVS algoritmem neporuší podmínka AVL a proto není třeba tvar stromu měnit.

Případ 2 nebo 3 nastávají, pokud přidáním vrcholu podle BVS algoritmu vznikne alespoň jeden silně nevyvážený vrchol. Vzniknout jich může víc, ale jestě postřehnete, že jsou vždy na cestě spojující kořen a přidaný vrchol. Označme jako v nejnižší ze silně nevyvážených vrcholů a w je ten z jeho listů, který má vyšší podstrom.

Případ 2: Buď je w vyvážen a nebo jsou vrcholy v a w nakloněny na stejnou stranu.

V případě 2 stačí provést rotaci hrany spojující v s w. Tím se spraví nejen vyvážení vrcholu v, ale i všech případných silně nevyvážených vrcholů nad ním.

Podmínka tohoto případu totiž říká, že nevyváženost vrcholu v způsobuje nízký podstrom druhého syna vrcholu v než je w a případně příliš vysoký podstrom syna vrcholu w, který je vzdálenější od v. Při rotaci se ale nízký podstrom prodlouží a příliš hluboko zasahující podstrom naopak stoupne a tím se rovnováha v původním podstromu vrcholu v (jehož novým kořenem je nyní w) obnoví. Navíc se tomuto stromu přidáním vrcholu zvýšila hloubka o 1 (což způsobilo vznik silně nevyvážených vrcholů nad v), ale rotací se jeho hloubka vrátila na původní velikost a tím se silně nevyvážené vrcholy nad v zase spravily.

Případ 3: Vrcholy v a w jsou nakloněny na opačné strany.

V tomto případě by samotná rotace hrany $v-w$ silnou nevyváženost v neodstranila. Ta je totiž způsobena tím, že příliš hluboko zasahuje podstrom toho syna w který je blíže k v. Označme tento vrchol z. Podstrom vrcholu z se ale při rotaci nepohybuje a proto dosahuje stále stejně hluboko.

V tomto případě se proto provede nejprve rotace hrany $w - z$. Po rotaci je sice v stále silně nevyvážený, ale namísto případu 3 nastane nevyváženost z případu 2 a následná rotace hrany $v-w$ tedy ze stromu učiní opět AVL strom.

Povšimněte si, že po přidání vrcholu se u některých vrcholů na cestě od přidaného vrcholu zpět ke kořeni objeví rozdvojení. Pod černým vahadélkem vrcholu se objeví neúplně zakryté bílé vahadélko, skloněné pod jiným úhlem. Vysvětlím nyní, co to znamená. Při zjišování, který vrchol je popřípadě silně nevyvážený nebudeme vždy znovu určovat výšky podstromů jeho synů, protože by to bylo příliš zdlouhavé. Namísto toho bude každý vrchol zahrnovat proměnou, které má hodnoty “nakloněn vlevo”, “vyvážený” a “nakloněn vpravo” (a popřípadě “silně nevyvážený vlevo/vpravo”). Pokud se vyváženost vrcholu změní, musíme hodnotu proměnné opravit.

Přidáním listu do AVL-stromu se vyváženost řady vrcholů změní, ale chvíli potrvá, než příslušným způsobem opravíme hodnoty proměnných, udávajících u jednotlivých vrcholů jejich vyváženost. Tyto proměnné jsou tedy po jistou dobu neaktuální, což je vyjádřeno dvojitými vahadélky. černá vahadélka v popředí vyjadřují skutečný stav a proto se jejich polohy okamžitě změní po přidání listu. Bílá vahadélka v pozadí označují hodnoty proměnných, uložené v počítači; jejich hodnoty zůstávají po jistou dobu nesprávné a vzhůru stoupající ukazatel aktuálního vrcholu je aktualizuje. Z tohoto důvodu nemůžeme přidávání okamžitě po přidání listu ukončit, ani kdyby AVL-podmínka zůstala zachována. Jistě poznáte, proč někdy ukazatel aktivního vrcholu vystoupá až ke kořeni a jindy se poměrně brzo zastaví.

Stoupající ukazatel také přirozeným způsobem určí požadovaný nejnižší ze silně nevyvážených vrcholů, pokud silně nevyvážené vrcholy vzniknou. Je to ten vrchol, na který ukazatel narazí nejdříve. Je jistě zřejmé, že po provedení případné rotace nebo dvojrotace se proměnné vyvážení vrcholů ležících nad rotovanou hranou stanou aktuálními aniž by bylo třeba je měnit. Provedením rotace/dvojrotace tedy přidávání vrcholu vždy skončí.

Scéna: *Vynechávání vrcholu AVL-stromu*

Vynechávání vrcholu je jako obvykle nejkomplikovanější operací. Některé případy se ale odbudou stejně jako u BVS. Vynechávání na základě klíče se převede na vyhledávání a vynechávání na základě vrcholu. Vynechávání vrcholu se 2 syny se převede na přenos klíče a vynechání listu nebo vrcholu s jedním potomkem přesně stejně jako u BVS. Proto se budeme zabývat pouze vynecháváním zadaného vrcholu s nejvýše jedním synem.

Ať je vynecháván list nebo vrchol s jedním synem, otci vynechávaného vrcholu se sníží výška podstromu jednoho z jeho synů, levého nebo pravého. Zde se budeme zabývat snížením výšky podstromu na straně levého syna, druhý případ se vyřeší na základě symetrie.

První možnost je, že vynecháním vrcholu způsobem podle BVS algoritmu se AVL podmínka neporuší, proto by tím vynechávání mohlo skončit, ale stejně jako v případě přidávání vrcholu je třeba aktualizovat informaci o vyvážení ve vrcholech (tedy graficky řečeno upravit polohu vyčnívajících bílých vahadélek). Pro úplnost je tato možnost nabídnuta ke krokování.

Po provedení prvního vynechání se dostaneme do situace, kdy bude vynechávání komplikovanější. Při krokování této operace se setkáme s oběma případy, které je nutno ošetřit. Podobně jako u přidávání od místa provedení operace (zde vynechání vrcholu) stoupá zelený ukazatel vzhůru, aktualizuje polohu bílých vahadélek a popřípadě nachází silně nevyvážené vrcholu, jejichž vyváženost obnovuje rotacemi.

Proti přidávání vrcholu jsou zde ale dva rozdíly: vytržením vrcholu může vzniknout *nejvýše jeden* silně nevyvážený vrchol (u přidávání jich mohlo být hodně). Úprava vyváženosti případného silně nevyváženého vrcholu ale může vytvořit jiný silně nevyvážený vrchol nad ním (zatímco při přidávání se po vyvážení nejnižšího defektního vrcholu vše napravilo). Proto obecně může dojít k tomu, že ukazatel stoupá a opakovaně naráží na silně nevyvážené vrcholy, které je třeba opravit rotací nebo dvojrotací.

Způsob vyvažování silně nevyváženého vrcholu detekovaného ukazatelem nebudu podrobněji rozebírat; je prakticky stejné jako pri přidávání.

Scéna: *Operace s AVL*

Podobně jako tomu bylo u BVS, v této scéně je možno si vyzkoušet všechny výše popsané operace s AVL stromy.

Scéna: *červeno-černý strom*

Nyní bude ukázán jiný způsob, jak předejít vzniku příliš nevyváženého stromu, t.zv. červeno-černý strom. Tato scéna se obsluhuje přesně stejně jako v úvodní scéně AVL-stromů, ale je použit jiný typ stromu.

Jak je vidět, název stromu pochází z toho, že se jedná o binární vyhledávací strom speciálního tvaru, pro jehož popis je každý vrchol stromu obarven červenou nebo černou barvou.

U červeno-černého stromu musí platit následující podmínky:

1. *každá cesta začínající v kořenu a končící ve vrcholu, kterému schází jeden nebo oba synové, musí obsahovat stejný počet černých vrcholů*

2. *syn červeného vrcholu nesmí být červený*

3. *kořen stromu je černý.*

Vrcholu, kterému schází jeden nebo oba synové, budeme říkat *neúplný* vrchol.

Povšimněte si, že první podmínka implikuje, že počet černých vrcholů na cestě z kořene do *neúplného* vrcholu v nemůže být menší než počet černých vrcholů na cestě z kořene do *libovolného* jiného vrcholu w, protože cesta z kořene do w se dá protahnout na cestu z kořene přes w do nějakého neúplného vrcholu z a cesty z kořene do v a do z musí mít stejně černých vrcholů. Tuto podmínku budeme často používat.

Podmínky implikují, že poměr délek nejkratší a nejdelší mezi cestami z kořene do neúplného vrcholu je nejvýše 2: černých vrcholů je na takových cestách stejně (první podmínka) a červených nemůže být více než černých (druhá a třetí podmínka). Červeno-černý strom je proto poměrně dobře vyvážený.

Třetí podmínka je nepodstatná, pokud by nebyla splněna, stačí kořen začernit; tím se ani první ani druhá podmínka neovlivní (kořen je jediný vrchol, který leží ve všech cestách uvažovaných v první podmínce a proto jeho barvu lze libovolně měnit, aniž by tím první podmínka byla ohrožena a začernění vrcholu nemůže ohrozit druhou podmínku). Používáme ji jen proto, že jinak by počet červených vrcholů na cestě podle 1. podmínky mohl být o 1 větší než počet černých vrcholů - drobná komplikace.

Technicky vzato je první podmínka zdaleka nejzávažnější, neboť její porušení se velmi obtížně napravuje, protože má globální charakter a týká se všech cest v celém stromě. Budeme se proto snažit ji stále dodržovat i za cenu dočasného porušení druhé podmínky, která se týká lokální situace a lépe se napravuje postupným přebarvováním vrcholů.

Ověřte si platnost všech podmínek a vyváženost stromu při postupném přidáváním do stromu.

Podobně jako u AVL-stromů se vyhledávání, určování minima a maxima provádějí algoritmy binárního vyhledávacího stromu, které nemění tvar stromu ani obarvení vrcholů a proto nemohou porušit výše uvedené tři podmínky.

Scéna: *Rotace v červeno-černém stromu*

Následující scény ukáží, jak se provádí přidávání a vynechávání v červeno-černém stromu. Stejně tak jako u AVL-stromů je zde základní operací rotace, přibývají také přebarvování vrcholů.

V červeno-černém stromu budeme provádět rotaci hrany od vrcholu u k jeho synovi w v případě, kdy barva syna w je červená. Pokud je i barva vrcholu u červená, rotace barvy žádných vrcholů nezmění (tato možnost by správně neměla nastat, ale u některých operací povolíme na krátkou dobu porušení druhé červeno-černé podmínky). Pokud je barva vrcholu u černá, pak se barvy vrcholů prohodí. V animaci je to znázorněno tak, že se černá barva vrcholu u převede na vrchol w (a u tím zčervená).

Výše uvedená podmínka provádění rotace i způsob jejího provedení jsou vedeny tím, že za této situace se neporuší první podmínka červeno-černého stromu. Pokud by se rotovala hrana, jejíž spodní vrchol je černý, k porušení podmínky by došlo. Zkuste si různé možnosti rotace v appletu.

I přípustná rotace ale může porušit druhou podmínku červeno-černého stromu, pokud horní vrchol rotované hrany je černý a má dva červené syny. Jak ale bylo řečeno, porušení druhé podmínky občas na chvíli tolerujeme, a proto takovou rotaci považujeme za přípustnou.

Scéna: *Červeno-černé vkládání - případ 0*

Nebudeme se zvlášť zabývat přidáním do prázdného stromu, kdy stačí vytvořit černý kořen jako jediný vrchol stromu a vložit do něj klíč.

U neprázdného stromu se, obdobně jako u AVL-stromů, i zde se nejprve přidá vrchol způsobem podle BVS algoritmu. V každém případě je přidán list vrcholu, kterému scházel syn. Nově přidaný vrchol obarvíme červeně. Tím se nemůže porušit první podmínka. Jelikož neměníme kořen stromu ani jeho barvu, neporuší se ani třetí podmínka.

V této scéně byl nový červený vrchol přidán jako syn k černému vrcholu a proto nedojde ani k porušení druhé podmínky a proto po přidání vrcholu je možno skončit. Z čistě formálních důvodů je animace provedena tak, že vtchol je přidán "na slepo" a teprve poté se podíváme, jaká je barva otce nově přidaného vrcholu; v této scéně vidíme, že je správná a proto skončíme.

Scéna: *Červeno-černé vkládání - případ 1*

Podobně jako v předchozí scéně, i zde přidáme nový vrchol pomocí BVS algoritmu a obarvíme jej červeně. Je-li dán klíč, který má ležet v nově přidaném vrcholu, nemůžeme ovlivnit, kam bude nový vrchol přidán, a zde je přidán jako syn červeného vrcholu a proto přidáním dojde k porušení druhé červeno-černé podmínky. Proto musí následovat operace odstraňování dvou červených vrcholů nad sebou, kterou je obecně nutno provádět opakovaně.

Při odstraňování dvou červených vrcholů nad sebou budeme rozlišovat tři případy. Tato scéna ilustruje první z nich, kdy nově přidaný vrchol je červený, jeho otec je také červený a navíc nově přidaný vrchol má dědečka a *červeného* strýce.

Jak uvidíme dále, je výhodné si situaci popsat obecněji: ve stromu existuje vrchol v, který je červený a má červeného otce a kromě toho dvojice v a jeho otec je jediná dvojice porušující druhou červeno-černou podmínku. V dalším budeme ostatní vrcholy budeme označovat podle jejich vztahu k vrcholu v.

V této scéně tedy předpokládáme, že v a jeho otec jsou červení a navíc vrchol v má dědečka a červeného strýce.

Uvědomte si, že právě popsané situace dvojice dědeček a otec vrcholu v je v pořádku a jelikož otec je červený, dědeček musí být *černý*.

Operace, kterou zde provedeme, je následující: dědečka přebarvíme na červeno a jeho syny (tedy otce a strýce nově přidaného vrcholu - oba červené) začerníme. Operace je animována tak, že se černá barva z dědečka převádí na otce a strýce.

Uvedená operace, kterou budeme často provádět i v dalších scénách, zjevně neporuší platnost první červeno-černé podmínky. Navíc se otec nově přidaného vrcholu začerní a tedy dvojice nový vrchol - jeho otec již neporušuje druhou červeno-černou podmínku.

Provedení právě popsané operace však nutně nezaručí, že vznikne správný červeno-černý strom. Mohou nastat dva problémy.

Pokud dědeček vrcholu v měl červeného otce, vznikne nová nepřípustná dvojice červených vrcholů, přesně tak, jak se tomu stane v této scéně. Na tuto dvojici musíme znovu opakovat buď operaci popsanou v této scéně, kdy za vrchol v budeme nyní pokládat dědečka původního vrcholu v, nebo operace popsané dále, v závislosti na jejich aplikovatelnosti. V této scéně se zde popsaná operace - převedení černé barvy z dědečka aktuálního vrcholu v na jeho otce a strýce - aplikuje postupně třikrát. Jelikož však provedením této operace se problém v nejhorším případě převede o dvě patra výše, počet opakování nemůže být i v tom nejhorším případě větší než polovina výšky stromu.

Přebarvení dědečka vrcholu v na červeno může také způsobit jiný problém: porušení třetí podmínky v okamžiku, kdy dědeček je kořen. Jak jsme ale již řekli, zde je náprava jednoduchá: zčervenalý kořen prostě začerníme a tím dostaneme správný červeno-černý strom a můžeme skončit. Takto také končí právě probíraná scéna.

Scéna: *Červeno-černé vkládání - případ 2*

Pokud ve stromu existuje právě jedna dvojice červených vrcholů, které jsou ve vztahu otec - syn (syna budeme zase označovat v), může nastat i jiná komplikace než ty, které byly popsána výše. V předchozí scéně jsme probrali dva případy: v nemá dědečka a nebo v má dědečka a červeného strýce. V této scéně a scéně následující probereme případ, kdy v má dědečka (který nutně musí být černý) a buď *nemá* strýce nebo je strýc *černý*. V obou případech budeme postupovat stejně, pro jednoduchost ukážeme jen postup v případě černého strýce.

Jistě si všimnete, že posledně jmenovaný případ nemůže nastat ihned po přidání nového vrcholu do červeno-černého stromu, protože by cesta z kořene do černého strýce měla více černých vrcholů než cesta do nového vrcholu, který je v okamžiku přidání list, tedy neúplný. Zde tedy nejprve provedeme operaci z minulé scény, ale vrchol, který při této operaci zčervená - budeme jej označovat v a bude to dědeček nově přidaného vrcholu - pak bude mít červeného otce, černého dědečka a černého strýce. V této scéně navíc je vrchol v levým synem svého otce a jeho otec také levým synem svého otce, tj. dědečka vrcholu v. Obdobně by se postupovalo v zrcadlově symetrické situaci, kdy by jak v, tak i jeho otec byli praví synové.

V takovém případě provedeme rotaci hrany, spojující dědečka a otce vrcholu v. Jelikož je to hrana vedoucí od černého vrcholu dolů k červenému vrcholu, je rotace přípustná a nedojde proto k porušení první červeno-černé podmínky. Navíc má horní vrchol rotované hrany (dědeček vrcholu v) jednoho syna červeného (otec vrcholu v) a jednoho černého (strýc vrcholu v) a proto se nevytvoří nová hrana, porušující druhou červeno-černou podmínku. K vytvoření takové hrany by nedošlo ani pokud by vrchol v strýce neměl (případ, který zde neukazujeme).

Na druhé straně ale vidíme, že rotace začerní otce vrcholu v a proto hrana spojující otce vrcholu v s vrcholem v (která zůstává ve stromu i po rotaci) bude druhou podmínku splňovat také. Provednou operací tedy vznikne správný červeno-černý strom a můžeme skončit.

Scéna: *Červeno-černé vkládání - případ 3*

Tato scéna je obdobná jako předchozí: po přidání nového vrcholu vznikne nepřípustná červená hrana a po provedení spuštění černé barvy z dědečka nového vrcholu na dědečkovy syny se objeví vrchol v, který má červeného otce, černého dědečka a černého strýce. Na rozdíl od předchozí scény ale je vrchol v pravým synem svého otce a jeho otec je naopak levým synem svého otce, tj. dědečka vrcholu v. Zrcadlově obráceně by se postupovalo, pokud by naopak byl v levým synem a jeho otec pravým synem.

Za této situace provedeme nejprve rotaci hrany, spojující vrchol v s jeho otcem. Je to případ, o kterém jsem se zmiňoval ve scéně o rotaci - rotujeme hranu, která by ve správném červeno-černém stromu neměla být, ale dočasně ji připouštíme. Důležité je, že tato rotace je přípustná - neporuší první podmínku. Kromě toho zjevně rotovaná hrana zůstává jedinou hranou, porušující druhou podmínku. Povšimněte si také, že vrchol, který byl původně dědečkem vrcholu v je nyní jeho otcem.

Nyní je již vidět, proč jsme rotaci prováděli: převede strom na případ z minulé scény a proto stačí provést ještě rotaci hrany spojující vrchol v s jeho současným otcem (před chvílí ještě dědečkem) a dostáváme správný červeno-černý strom a operace končí.

Tím jsme probrali všechny situace, které mohou nastat při přidávání nového vrcholu do červeno-černého stromu.

Scéna: *Červeno-černé vynechávání - případ 1*

V této scéně a scénách následujících se budeme zabývat vynecháváním vrcholu z červeno-černého stromu. Vynechávání rozdělíme do 14 případů, z nichž každý má svou podmínku aplikovatelnosti a související akci. Při vynechávání najdeme první aplikovatelný případ (v pořadí, ve kterém jsou uváděny) a provedeme odpovídající akci. Při provádění akce některého případu tedy můžeme předpokládat nejen to, že je splněna jeho podmínka aplikovatelnosti případu, ale také, že *nejsou* splněny podmínky případů předchozích.

V dalším budeme vynechávaný vrchol značit v a označovat jej fialovým podkladem. Dále označíme (pokud existují) jeho otce jako u, jeho bratra jako w, levého syna bratra w jako x a pravého syna bratra w jako y. Počet černých vrcholů na cestě z kořene do nějakého vrcholu budeme nazývat *černé skóre* daného vrcholu.

Zopakujme s první scény týkající se červeno-černých stromů, že z první červeno-černé podmínky plyne následující tvrzení: žádný vrchol r červeno-černého stromu nemůže mít černé skóre větší než libovolný neúplný vrchol s.

Specifické vlastnosti červeno-černého stromu vedou k tomu, že někdy bude jednodušší provádět vynechávání jinak, než se dělá v klasickém BVS.

První scéna ukazuje vynechávání klíče úplného vrcholu, tedy vrcholu s oběma syny (v našem případě vynecháváme přímo kořen, ale to není podstatné). Zde budeme postupovat přesně stejně jako u BVS: zahodíme klíč, ale ponecháme vrchol, převedeme do něj nejblíže nižší klíč a pak vynecháme vrchol bez klíče který, jak již dávno víme, nemá pravého syna a proto se na něho vztahuje některý z následujících případů.

Akci si proveďte s tím, že vynechávání vrcholu bez pravého syna asi nebudete rozumnět.

Scéna: *Červeno-černé vynechávání - případ 2*

V této scéně se vynechává klíč vrcholu v s jedním synem. V červeno-černém stromě tato situace nastává jen ve velmi speciálním případě. Ani syn, ani případný vnuk vrcholu v totiž nemůže být černý, protože v je neúplný, ale cesta do neúplného v by obsahovala méně černých vrcholů než cesta do jeho syna nebo vnuka. Syn vrcholu v tedy musí být červený. Kdyby existoval vnuk vrcholu v, pak jsme si právě řekli, že nemůže být černý, ale nemůže být ani červený jako syn červeného syna vrcholu v. v tedy nemá vnuky neboli jeho jediný syn je červený list.

Akce v tomto případě je následující (odlišná od BVS): klíč vrcholu v vyhodíme, přesuneme do něj klíč jeho syna a pak syna vynecháme. Jelikož syn vrcholu v je červený list, nezpůsobíme si tím, žádné potíže.

Scéna: *Červeno-černé vynechávání - případ 3*

Další jednoduchý případ je vynechávání červeného listu. O tom jsme se již vlastně zmínili v závěru předchozí scény - prostě vrchol vynecháme a je to.

Scéna: *Červeno-černé vynechávání - případ 4*

V této a následujících scénách tedy budeme vynechávat vrchol, který je černým listem. Nemůžeme jej prostě vyhodit, protože by se tím porušila první červeno-černá podmínka způsobem, ze kterého bychom těžko hledali nápravu a proto musíme postupovat daleko opatrněji a obecně také komplikovaněji.

Tato scéna je zatím zcela primitivní: předpokládáme, že vynechávaný černý list je zároveň kořenem. Vyhodíme jej a zůstane prázdný strom (což je matematická formulace pro případ, kdy nic nezůstane).

V dalších scénách tedy budeme předpokládat, že vynechávaný černý list v má otce u. Otec má *menší* černé skóre než jeho černý syn v a proto nemůže být neúplný. Vrchol v proto má i bratra, kterého budeme značit w.

Scéna: *Červeno-černé vynechávání - případ 5*

V této scéně budeme předpokládat, že bratr w vynechávaného černého listu v je *červený*. V takovém případě otec u vrcholu v je dle druhé podmínky černý. Navíc černé skóre bratra w je menší než černé skóre vrcholu v a proto bratr w nemůže být neúplný. Jelikož bratr w je červený jeho dva synové musí být černí dle druhé podmínky. Není důležité, zda jsou to listy (jako je znázorněno) nebo mají syny.

V dané situaci provedeme rotaci hrany spojující černého otce u a červeného bratra w. Tato rotace je přípustná a neporušuje druhou podmínku, protože druhý syn otce u - vynechávaný vrchol v je černý. Dostaneme tedy správný červeno-černý strom, ve kterém vynecháváme vrchol v, který je stále černým listem, ale jehož nový bratr je jeho původní synovec (syn bratra w) a tedy je černý.

Rotace převede případ 5 na některý z případů následujících. Proveďte si animaci, i když akce po provedení rotace ještě nebyly vysvětleny.

Scéna: *Červeno-černé vynechávání - případ 6*

Od nynějška tedy budeme zkoumat situaci, kdy že vynechávaný černý list v má otce u a *černého* bratra w.

V následujících dvou scénách budeme předpokládat, že otec u vynechávaného černého listu v je *černý* a jeho bratr *není* list. V této scéně předpokládáme, že v je levý syn a w pravý syn otce u a bratr w má levého syna x (není důležité, zda má i pravého syna). Zrcadlově symetricky bychom postupovali i pokud by v byl pravý syn a w levý syn, který by měl pravého syna (bez ohledu na to, zda by měl levého syna).

Povšimněte si, že černé skóre vrcholu v a jeho bratra w jsou stejná. Kdyby syn nebo případný vnuk bratra w byl černý, pak by měl černé skóre větší než je černé skóre samotného bratra w a tedy také větší než je černé skóre vrcholu v, což ale nemůže nastat, protože v je neúplný vrchol.

Levý syn x bratra w je tedy červený a jeho případný syn nemůže být ani červený (druhá červeno-černá podmínka) ani černý (viz předchozí odstavec). Syn x je tedy červený list.

Operace provedená v daném případě je jednoduchá, i když trochu zdlouhavá. Klíč vrcholu v vyhodíme, do v přesuneme klíč z jeho otce u, do otce u přesuneme klíč z levého syna x bratra w vrcholu v. Tím jsme se dostali do případu 3 - potřebujeme vyhodit červený list x a to jednoduše provedeme, aniž bychom ohrozili platnost podmínek červeno-černého stromu. Povšimněte si, že přesuny klíčů se provádějí tak, že žádný z nich nenaruší platnost pravidla o distribuci klíčů do vrcholů binárního vyhledávacího stromu.

Scéna: *Červeno-černé vynechávání - případ 7*

Tato scéna popisuje případ velmi podobný předchozí scéně. Vynechávaný vrchol v má černého otce a černého bratra, který není list. Na rozdíl od předchozí scény však bratr postrádá jednoho syna, toho, který by byl bližší vrcholu v. Na obrázku tedy bratr w má pouze pravého syna y. Stejně jako v předchozí scéně se dá dokázat, že synovec y musí být červený list.

Postupujeme podobně jako v minulé scéně, ale přesunů klíčů je více. Klíč vrcholu v vyhodíme, přesuneme do něj klíč otce u, do otce u přesuneme klíč bratra w a nakonec do bratra w přesuneme klíč synovce y. Pak již, podobně jako v předchozí scéně, vynecháme červený list y, který je nyní bez klíče. I zde je zřejmé, že přesuny klíčů nenarušují platnost pravidla o distribuci klíčů a vynechání červeného listu je vlastně převedení na případ 3.

Scéna: *Červeno-černé vynechávání - případ 8*

Pokud není aplikovatelný žádný z předchozích případů, pak jsou vynechávaný vrchol v i jeho bratr w černé listy. Zde si nejprve probereme několik jednoduchých případů. V této scéně předpokládáme, že společný otec těchto vrcholů je červený.

Operace, kterou provedeme nejdříve bude často používána i v následujících scénách a je obrácením operace, kterou jsme prováděli při přidávání vrcholu. Ve stromu máme červený vrchol, který má dva černé syny. Černou barvu synů přesuneme na jejich společného otce. Touto operací se neporuší ani první, ani třetí červeno-černá podmínka. Pokud synové nemají žádného červeného syna (například pokud jsou oba listy), pak nemůže dojít ani k narušení druhé podmínky.

Právě popsanou operaci aplikujeme na otce u vynechávaného vrcholu a na jeho syny, tedy na vynechávaný vrchol samotný a jeho bratra. Vzpomeňte si, že zde vynechávaný vrchol a jeho bratr jsou (černé) listy. Po provedení této operace se stane vynechávaný vrchol červeným listem a bez problémů jej vynecháme, viz případ 3.

Scéna: *Červeno-černé vynechávání - případ 9*

Poslední jednoduchý případ je když vynechávaný vrchol i jeho bratr jsou černé listy a jejich otec je černý a je kořenem stromu. Stom tedy má pouze tři vrcholy. Možná trochu složitě se vynechání provede tak, že se kořen přebarví na červeno (to poruší pouze třetí červeno-černou podmínku) a tím je případ převeden na situaci z předchozí scény. Tam popsané operace vynechají vrchol bez porušení prvních dvou červeno-černých podmínek a platnost třetí se automaticky obnoví.

Scéna: *Červeno-černé vynechávání - případ 10*

Nyní bude následovat několik případů, kdy prováděné akce jsou již značně komplikované. Ve všech z nich jsou vynechávaný vrchol v a jeho bratr w černé listy a jejich společný otec u je také černý a není kořen (vrchol v tedy má dědečka).

Zde i v dalších scénách týkajících se vynechávání si zavedeme pojem *dvojnásobě černý* vrchol. Bude to vrchol, který se do počítání černých vrcholů na cestách z kořene do neúplných vrcholů započítává dvakrát. Bude tedy například přípustné, aby (jak se tomu zakrátko stane při animaci) na některých cestách z kořene do neúplného vrcholu byly čtyři černé vrcholy a na jiných dva vrcholy černé a jeden dvojnásobně černý. Během provádění operací bude ve stromu obvykle nejvýše jeden dvojnásobně černý vrchol a jen velmi vyjímečně dva. Dvojnásobně černý vrchol nebo vrcholy budou pochopitelně připuštěny jen dočasně, než vynechávání skončí, budeme se jich muset zbavit.

Prvním krokem v této scéně i scénách následujících bude přenesení černé barvy vynechávaného vrcholu v a jejho bratra w do jejich společného otce u. Totéž jsme provedli i v případu 8, kdy ale otec byl červený a stal se tak jednoduše černý. Nyní byl otec černý a proto bude dvojnásobně černý. Druhou černou barvu budeme znázorňovat jako čtvercový černý podklad vrcholu.

Po přesunutí černé barvy z vynechávaného vrcholu a jeho bratra nahoru se vynechávaný vrchol stane červeným listem a bylo by jej možno snadno vynechat způsobem podle příkladu 3, ale než to uděláme, musíme se zbavit dvojnásobně černé barvy ve stromu.

Nebudeme se zvlášť zabývat případem, kdy by byl dvojnásobně černý vrchol kořenem, protože v tomto případě se prostě kořen změní na jednonásobně černý, což nezoůsobí žádný problém. Dvojnásobně černý vrchol tedy bude mít otce. Bude mít i bratra, protože jinak by jeho otec byl neúplný vrchol a přitom by černé skóre jeho otce bylo menší než černé skóre vrcholu samého. Bratr W sám musí mít oba syny, protože jinak by byl neúplný a přitom je jeho černé skóre menší než černé skóre dvojnásobně černého vrcholu V. Protože bratr W je červený, jeho synové musí být černí.

První případ s dvojnásobně černým vrcholem je takový, že jinak správný červeno-černý strom obsahuje jeden dvojnásobně černý vrchol V, který má otce U *červeného* bratra W. Otec je tedy černý, jinak by byla porušena druhá podmínka. Provedeme rotaci hrany spojující černého otce U s červeným bratrem W. Je to přípustná rotace, která (s uvážením černosti vrcholu V samotného) neporuší žádnou podmínku červeno-černého stromu. Bratrem vrcholu V se po provedení rotace stane jeden z dřívějších synů jeho původního bratra W, tedy černý vrchol a proto se dostaneme do situace, na kterou bude aplikován některý z dalších případů.

Proveďte si animaci operací, i když další postup ještě nebyl vysvětlen. Všimněte si jen, že po odstranění dvojitě černé barvy nakonec vynecháme požadovaný vrchol, který se na začátku stal červeným listem.

Scéna: *Červeno-černé vynechávání - případ 11*

Začínáme opět stejně, černou barvu dvou černých listů, z nichž jednoho chceme vynechat, přesuneme na jejich černého otce, který se stane dvojnásobně černým. Víme, že dvojnásobně černý vrchol V má otce U a bratra W a jelikož není aplikovatelný předchozí případ, bratr W je černý.

V této scéně předpokládáme, že společný otec je červený a bratr W nemá žádného červeného syna. Prosté převedení jedné černé barvy z dvojnásobně černého vrcholu V a černé barvy jeho bratra W na společného červeného otce U neporuší podmínky a zbaví nás dvojitě černé barvy. Pak již jen vynecháme požadovaný vrchol, nyní červený list.

Scéna: *Červeno-černé vynechávání - případ 12*

Tento případ je podobný jako případ předchozí: ve stromě se nachází dvojnásobně černý vrchol V, který má černého bratra W, který nemá žádného černého syna. Jediný rozdíl je v tom, že společný otec U vrcholu V a jeho bratra W je také černý.

Operace kterou provedeme bude stejná jako v předchozím případě, ale rozdíl je v tom, že vrchol V bude po jejím provedení jednoduše černý, ale dvojnásobně černou barvu bude mít otec U. Proto budeme muset znovu provádět některý z případů 10-14, ale dvojnásobně černý vrchol se posunul o jedno patro směrem ke kořeni.

Viděli jsme, že případ 10 se převedl na případ 11 a každý z případů 11, 13 a 14 představuje krátkou konečnou posloupnost akcí, jejíž délka nezávisí na velikosti stromu, kterými se dvojnásobně černého vrcholu zbavíme, aniž bychom se znovu dostali do případu 12.

Jediný případ, kdy se provádění operací brzo nezastaví, je tedy případ 12, který vede opakovaně znovu na situaci podle případu 12. Jak ale bylo řečeno, poloha dvojnásobně černého vrcholu stále stoupá ke kořeni, takže se případ 12 nemůže opakovat vícekrát, než kolik je hloubka stromu. Nakonec se v nejhorším případě dostaneme do situace, kdy je nově vytvořený dvojnásobně černý vrchol kořenem a tomu prostě jednu čenou barvu odejmeme, jak bylo popsáno výše.

Scéna: *Červeno-černé vynechávání - případ 13*

Případ 13 nastává, když se objeví dvojnásobně černý vrchol V, který má černého bratra W a ten má červeného syna Y na straně vzdálenější od vrcholu V, zatímco syn X bratra W, bližší k vrcholu V je černý. Buď je tedy V levý syn a W pravý syn jejich společného otce U a Y je pravý syn bratra W, nebo je V pravý syn a W levý syn a Y je levý syn bratra W. Na obrazovce je první možnost.

Barva společného otce U vrcholu V a jeho bratra W není důležitá. My případ 13 rozdělíme podle barvy otce U na 13a (otec černý) a 13b (otec červený). V obou případech ale budeme provádět stejné akce.

Podívejme se nejprve na 13a. Po vzniku dvojnásobně černého vrcholu se nejprve spustí černá barva bratra W na jeho syny. Od vrcholu V odvrácený červený synovec Y vrcholu V tím zčerná (jednoduše), zatímco přivrácený synovec X se stane druhým dvojnásobně černým vrcholem.

Zdálo by se, že se situace zhoršila, ale po rotaci hrany spojující černého otce U s nyní červeným bratrem W již jist vidíte, že jsme blízko cíle. Dvojnásobně černý dřívější synovec X vrcholu V se nyní stal bratrem dvojnásobně černého vrcholu V a jejich společným otcem je vrchol U, který již byl předtím otcem vrcholu V a po provedení rotace dostal červenou barvu. Provedená rotace je přípustná, protože horní konec rotované hrany byl černý a dolní červený a navíc je vrchol V (dvojnásobně) černý a tedy se ani druhá podmínka neporuší.

Nyní již stačí převést jednu červou barvu vrcholů V a X na otce U. Barva vrcholů V a X se sníží na jednoduše černou a vrchol U (jednoduše) zčerná a tím vznikne správný červeno-černý strom bez dvojnásobně černých vrcholů.

Nyní již stačí vynechat zvolený vrchol, nyní červený list a vynechávání je ukončeno.

V případě 13b je otec U červený a provádíme tytéž operace. Po spuštění černé barvy bratra W dolů však vznikne hrana spojující červeného otce U s nyní červeným bratrem W. Tuto hranu rotujeme stejně jako v případě 13a; i zde je rotace přípustná a nevytvoří jinou hranu porušující druhou červeno-černou podmínku. Nakonec posunutí jedné černé barvy z nově vzniklé bratrské dvojice V a X na U nejen odstraní dvojnásobně černé barvy vrcholů V a X, ale také začerněním jednoho z vrcholů do té doby druhou podmínku porušující hrany spojující U a W obnoví platnost druhé červeno-černé podmínky.

Scéna: *Červeno-černé vynechávání - případ 14*

Nakonec nejkomplikovanější případ: ve stromu je dvojnásobně černý vrchol V, který má otce U a černého bratra W a tento bratr má červeného syna X na straně přivrácené k vrcholu V. Buď je tedy V levým synem a W pravým synem jejich společného otce U a X je levým synem vrcholu W (to je co vidíte na obrazovce) a nebo je zrcadlově obráceně V pravým a W levým synem a synovec X je pravým synem vrcholu W.

Barva druhého syna Y bratra W a barva otce U nejsou pro sekvenci prováděných operací podstatné; ukážeme si proto čtyři varianty 14a, 14b, 14c, 14d. V prvních dvou je otec U černý a v následujících dvou červený, zatímco syn Y je černý v první a třetí možnosti a červený ve zbývajících dvou.

Podívejme se nyní nejprve na 14a. Jakmile se objeví dvojnásobně černý vrchol V, provedeme rotaci hrany spojující jeho černého bratra s červeným synovcem X přivráceným k vrcholu V. Tím se situace převede na případ 13a a postupuje se, jak bylo ukázáno v minulé scéně.

Stejně postupujeme i v případě 14b. Zde se ale po úvodní rotaci objeví hrana s oběma konci červenými. Je to hrana, spojující původního bratra W se vzdáleným synovcem Y. V následujícím spuštění černé barvy z nového bratra vrcholu V, kterým je jeho původní synovec X, ale horní konec hrany porušující druhou podmínku zčerná a platnost druhé podmínky se obnoví. Pak již operace probíhají stejně jako v případě 14a.

Případy 14c a 14d probíhají analogicky případům 14a, respektive 14b s tím, že rotace je převede na případ 13b. V případě 14d tedy rotací vznikne jedna hrana porušující druhou podmínku (spojuje původního bratra W s původním vzdáleným synovcem Y). Po spuštění černé barvy se tato hrana spraví, ale vznikne nová hrana porušující druhou pormínku (spojuje původního otce U s původním bratrem W). Tato hrana je upravena při závěrečném přesunutí jedné z černých barev dvojnásobně černých vrcholů nahoru.

Scéna: *Operace na červeno-černém stromu*

V této scéně si můžete vyzkoušet libovolné operace na náhodně vytvořeném červeno-černém stromu předem zvolené velikosti.