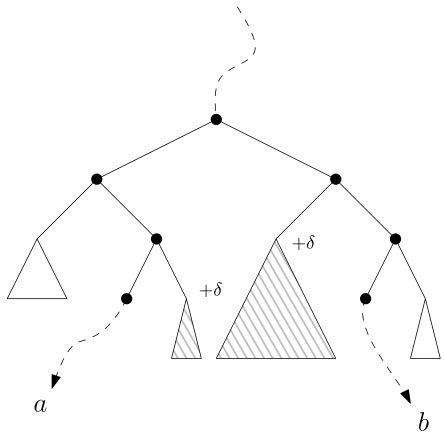
1. *Změna intervalu: Uvažme obecný BVS uchovávající dvojice (klíč, hodnota) setříděné podle klíče. Se zachováním asymptotické složitosti všech operací naučte BVS operaci přičtení $\delta$ ke všem hodnotám v intervalu klíčů $[a, b]$.*

   Řešením této úlohy je použít princip tzv. líného vyhodnocování. Nemáme při provádění této operace čas změnit všechny hodnoty v intervalu klíčů, ale máme čas poznamenat si pro které podstromy máme hodnoty změnit až se do nich někdy v budoucnu budeme zanořovat. Pro každý vrchol $v$ stromu si pořídím hodnotu $zmena(v)$, který bude určovat jakou hodnotu mám přičíst ke všem vrcholům v podstromě $v$. Na začátku $zmena(v) = 0$ pro každý vrchol.

   Operaci přičtení $\delta$ v intervalu $[a, b]$ provedeme následujícím způsobem. Nechť $v$ je nejnižší vrchol, který leží na cestě z kořene při hledání $a$ i $b$. Potom při hledání $a$ z vrcholu $v$ pokaždé když zahneme ve vrcholu $w$ doleva, zvětšíme hodnotu uloženou ve $w$ o $\delta$ a zároveň nastavíme $zmena(p(w)) = zmena(p(w))+\delta$. Stejně tak při hledání $b$ z vrcholu $v$ při každém zahnutí ve vrcholu $w$ doprava, zvětšíme jeho hodnotu o $\delta$ a zároveň $zmena(l(w)) = zmena(l(w)) + \delta$. Pomocí $p(w)$, respektive $l(w)$, značíme pravého, respektive levého, syna vrcholu $w$.

   Při každém jiném průchodu stromem (při operacích Insert, Delete) budeme změny propagovat. Tedy pokud například navštívíme vrchol $v$, upravíme hodnotu uloženou ve $v$ podle $zmena(v)$ a zároveň ke změnám v pravém a levém synovi přičteme $zmena(v)$. Následně hodnotu uloženou ve $v$ vynulujeme.



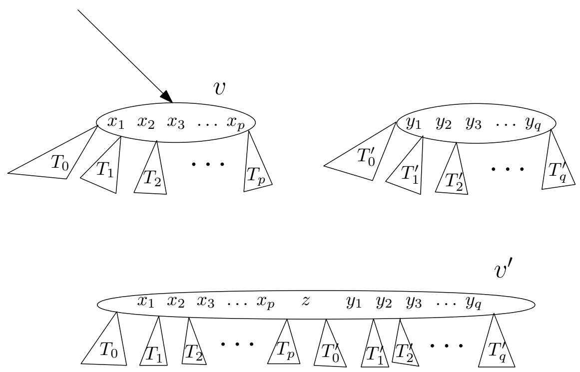

2. *Jednosměrné operace: Navrhněte úpravu operací* Insert *a* Delete *u $(a, b)$-stromů tak, aby stromem procházely pouze směrem dolů. Pro tento příklad předpokládejte, že $b \geq 2a$.*

   První se zaměříme na Insert. Budeme chtít po cestě dolů zařídit, že kdykoliv se zanoříme do potomka vrcholu $v$, pak má $v$ nejvýše $b - 1$ synů (tedy o 1 méně než je povolené maximum). Kdykoliv tedy narazíme na vrchol $w$ s $b$ syny, roztrhneme ho na 2 vrcholy s $\lfloor b/2 \rfloor$, respektive $\lceil b/2 \rceil$ syny. Prostřední klíč v původním vrcholu $w$ musíme vložit do otce. Ten se tam naštěstí vejde bez nutnosti štěpení jelikož jsme po cestě zařídili, že otec má nejvýše $b - 1$ synů. Zároveň pro nově vzniklé vrcholy platí $b - 1 \geq \lceil b/2 \rceil \geq \lceil b/2 \rceil \geq a$.

   Pro Delete postupujeme obdobně. Tentokrát budeme chtít po cestě dolů zařídit, že kdykoliv se zanoříme do potomka vrcholu $v$, pak má $v$ alespoň $a + 1$ synů (tedy o 1 více než je povolené minimum). Rozebereme situaci kdy narazíme na vrchol $w$ s právě $a$ syny. Existuje soused $w$, označme ho $x$. Pokud $x$ má alespoň $a + 1$ synů, půjčíme si jeden jeho podstrom. Jinak má $x$ pouze $a$ synů, a můžeme sloučit $x$ s $w$ do jediného vrcholu s $2a \leq b$ syny. K tomu potřebujeme odstranit jejich oddělovací klíč ve společném otci. Jenže tomu už jsme zařídili $a + 1$ synů a tedy odstraněním jednoho klíče a sloučením dvou jeho synů žádnou podmínku neporušíme[1].

3. $(a, b)$*-join: Navrhněte operaci* JOIN$(X, Y)$*, která dostane dva* $(a, b)$*-stromy* $X$ *a* $Y$ *a sloučí je do jednoho. Může se přitom spolehnout na to, že všechny klíče z* $X$ *jsou menší než všechny z* $Y$*. Zkuste dosáhnout složitosti* $O(\log |X| + \log |Y|)$.

   BÚNO má $Y$ menší hloubku než $X$. Začneme tím že najdeme minimální prvek $z$ ve stromě $Y$ a odstraníme ho. Tím dostaneme strom $Y'$. Nyní se ve stromě $x$ zanoříme po pravých potomcích do vrcholu $v$ ležícího v hloubce $h(X) - h(Y')$. Nyní nám jde o to spojit podstrom $v$ se stromem $Y'$. Nechť jsou $x_1, \ldots, x_p$ klíče uložené ve vrcholu $v$ a $y_1, \ldots, y_q$ klíče uložené v kořeni stromu $Y'$. Nyní vrchol $v$ nahradíme vrcholem $v'$ s klíči $x_1, \ldots, x_p, z, y_1, \ldots, y_q$ a odpovídajícími podstromy. Jediná podmínka, kterou nově vzniklý strom potenciálně nesplňuje, je na počet synů $v'$. Není těžké se rozmyslet, že $v'$ nemůže mít více než $2b$ synů, a tedy ho můžu roztrhnout na dva vrcholy a následně propagovat změnu směrem ke kořeni úplně identicky jako u operace INSERT.




Obrázek 1: Spojení kořene stromu $Y'$ s vrcholem $v$ a klíčem $z$ do jednoho obřího vrcholu, který ale neobsahuje více než $2b$ synů.

4. *Malé zaplnění: 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* $2/3$.

   Pouze lehký náznak, na cvičení jsme nestihli. Při vykonávání operace INSERT můžu aplikovat inverzní operaci k půjčení syna ze souseda - tj. můžu nadbytečný podstrom přesunout do sousedního vrcholu pokud ještě má místo. Pakliže je soused také zaplněn, pak mám dva vrcholy dohromady s $2b + 1$ syny a já je roztrhnu na celkem 3 nové vrcholy. Každý bude mít alespoň $\lfloor 2/3b \rfloor$ synů.

   Operace DELETE je krapet složitější, navíc zaplnění nepůjde garantovat pro syny kořenu (podobně jako kořen je jediný vrchol, který nemusí být zaplněn ani z půlky). Předpokládejme, že vrchol $v$ má dva sourozence. Potom při mazání z vrcholu $v$ si buďto můžeme půjčit z jednoho ze sourozenců, nebo mají všichni už právě $\lfloor 2/3b \rfloor$ synů (jinak bychom půjčením neporušili zaplněnost). Potom tyto tři sourozence můžeme nahradit dvěma novými, téměř zaplněnými, vrcholy. Zůstává otázka jak naložit s vrcholy, které nemají dva sourozence. Předně nemusí nutně jít o nejbližšího pravého a levého sourozence, můžeme klidně uvažovat

[1] Doporučuji konzultovat normální průběh DELETE popsaný v Průvodci.

například levého souseda a levého souseda levého souseda. To nám přesouvání podstromů lehce zkomplikuje, ale rozmyslete si že to stále stihneme v čase $O(1)$. Zůstává tedy vyšetřit situace, kdy má vrchol opravdu jen jednoho sourozence. Tady bohužel musíme začít místo druhého sourozence používat nejbližšího bratrance (vrchol se stejným dědem - tj. společným předchůdcem o 2 hladiny výše). Takový vždy existuje, pokud uvažujeme vrcholy od třetí hladiny.