

1. *Změna intervalu:* Uvažme obecný BVS uchovávající dvojice (klíč, hodnota) seříděné podle klíče. Se zachováním asymptotické složitosti všech operací naučte BVS operaci přičtení δ ke všem hodnotám v intervalu klíčů $[a, b]$.
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 $b \geq 2a$.
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 $O(\log |X| + \log |Y|)$.
4. *Malé zaplnění:* Nevýhodou (a, b) -stromů je, že plýtvaají paměť – může se stát, že vrcholy jsou zaplněné jen z poloviny. Navrhněte úpravu, která zaručí zaplnění z alespoň $2/3$.
5. *Seznam:* Sestrojte datovou strukturu pro uložení seznamu tak, abychom uměli rychle najít k -tý prvek a přesunout ho na začátek.

6. *(a, b)-split*: Navrhněte operaci $\text{SPLIT}(T, x)$, 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.
7. *Pořadí permutací*: Uspořádejme všechny permutace na množině $\{1, \dots, n\}$ lexikograficky. Vymyslete algoritmus, který pro dané k sestrojí v pořadí k -tou permutaci v čase $O(n \log n)$. Navrhněte též převod permutace na její pořadové číslo.
8. *Pomalé sjednocení*: Dokažte, že budeme-li reprezentovat množiny binárními vyhledávacími stromy, nelze sjednocení provést rychleji než lineárně v nejhorsím případě. Platí to dokonce i tehdy, máme-li na vstupu zaručený dokonale vyvážený strom a výstup může být jakkoliv nevyvážený.