

*Pokud jste nerozuměli rotacím AVL stromu z přednášky, nezapomeňte se ozvat.*

**Příklad 1.**

Implementujte dvojrotace pomocí jednoduchých rotací.

**Příklad 2.**

Ukažte, jak pomocí rotací převést libovolný strom na „levou liánu“ – tedy cestu, kde v kořeni je maximum a list obsahuje minimum. Kolik času a paměti bude potřeba?

**\*Příklad 3.**

Nyní tuto liánu pouze pomocí rotací převedte v lineárním čase na perfektně vyvážený BVS. Z toho odvodte, jak libovolný BVS převést na perfektně vyvážený pouze pomocí rotací.

**Příklad 4.**

Mějme BVS, pro jehož vrcholy máme přímé ukazatele. Vymyslete, jak pro daný vrchol najít jeho následníka, aniž byste museli jít od kořene. Pokud budete potřebovat, strom si můžete upravit.

**Příklad 5.**

Dokažte, že projdeme-li všechny vrcholy BVS postupným hledáním následníků, celkově tím strávíme lineární čas, ať je struktura stromu jakákoliv.

**Příklad 6.**

Mějme AVL strom použitý jako slovník: v každém vrcholu sídlí klíč a nějaká celočíselná hodnota. Upravte jej, aby uměl rychle zjistit největší hodnotu přiřazenou nějakému klíči z intervalu  $[a, b]$ .

**Příklad 7.**

Znovu mějme AVL strom jako slovník. Upravte jej, aby podporoval operaci, která ke všem hodnotám pro klíče z intervalu  $[a, b]$  přičte číslo  $k$ . Tato operace má běžet v čase  $\mathcal{O}(\log n)$ . Přičtení nemusí být okamžité – stačí, aby až FIND vrcholu viděl správnou hodnotu.

**Příklad 8.**

AVL strom pro svou práci potřebuje mít v každém vrcholu značky tří druhů. Ukažte, že ale existuje způsob, jak udržovat AVL strom s jedním bitem na vrchol.