Binární vyhledávací strom (BVS): Strom s vrcholy z nějaké uspořádané množiny X, kde platí, že $\forall\, a \in L(v)$: $a < v$ a $\forall\, a \in R(v)$: $v < a$
$\boldsymbol{T(v)}$ – podstrom vrcholu v
$\boldsymbol{l(v), r(v)}$ – levý/pravý syn vrcholu v
$\boldsymbol{L(v), R(v)}$ – levý/pravý podstrom vrcholu v
$\boldsymbol{h(v)}$ – hloubka podstromu v
Strom je **dokonale vyvážený**: $\forall\, v$ vrchol: $||L(v)| - |R(v)|| \leq 1$
Strom je **hloubkově vyvážený**: $\forall\, v$ vrchol: $|h(l(v)) - h(r(v))| \leq 1$

AVL strom: BVS udržující hloubkové vyvážení pomocí rotací.

1. Popište algoritmus, který ze setříděného pole vytvoří dokonale vyvážený BVS v lineárním čase.
2. Popište, jak v BVS najít nejbližší větší hodnotu y pro zadané x.
3. Ukažte, jak z jednoduchých rotací na AVL stromě složit dvojitou.
4. Ukažte, že v AVL stromě nikdy opakované mazání a přidávání téhož prvku nezpůsobí víc jak konstantní počet rotací celkem.
5. Navrhněte algoritmus, který v (modifikovaném) AVL stromě pro $a, b \in X$ najde počet vrcholů v intervalu $[a, b]$ v čase $\mathcal{O}(\log n)$.
6. Uvažme volnější podmínku hloubkového vyvážení stromu, která dovoluje rozdíl hloubek až H. Jak by vypadala varianta AVL stromu, která toto využívá, a jakou by měly časovou složitost její operace?
7. Mějme AVL strom s řetězci délky nanejvýš L řazenými lexikograficky. Jaká je asymptotická složitost jeho operací? Porovnejte jej s trií (písmenkovým stromem).

