

**Pole**  $\rightarrow$  **BVS**. Navrhněte algoritmus, který ze zadaného setříděného pole vytvoří dokonale vyvážený BVS v lineárním čase.

**BVS\_SPLIT**. Navrhněte operaci **BVS\_SPLIT**, která dostane BVS  $T$  a hodnotu  $s$  a vrátí dva BVS  $T_1, T_2$  takové, že hodnoty v  $T_1$  jsou menší než  $s$  a hodnoty v  $T_2$  jsou  $\geq s$ .

**Sekvence**. Nakreslete, jak se vyvíjí AVL strom, když do něj postupně přidáváme čísla 10, 20, 15, 25, 30, 16, 18, 19. Jak následně vypadá odebrání čísla 30?

**Hloubka**. Vyberte nějakou reprezentaci AVL stromu. Jak nyní spočítat jeho hloubku? Jaké jsou výhody a nevýhody této reprezentace?

**Minimum v intervalu**. Mějme AVL strom použitý jako slovník, tzn. v každém vrcholu je klíč a nějaká celočíselná hodnota. Upravte strom tak, aby uměl rychle odpovídat na dotaz, jaká je nejmenší hodnota mezi klíči v intervalu  $[a, b]$ .

**Datovka 1**. Sestrojte datovou strukturu, která bude umět vykonávat následující operace s příslušnými složitostmi:

- **Init()** – inicializuje strukturu –  $\mathcal{O}(1)$ .
- **INSERT( $X$ )** – vloží prvek  $X$ , pokud ve struktuře ještě není –  $\mathcal{O}(\log n)$ .
- **DELETE( $X$ )** – odstraní ze struktury  $X$ , pokud v ní je –  $\mathcal{O}(\log n)$ .
- **DELETE\_IN\_PLACE( $I$ )** – odstraní ze struktury prvek, který byl v pořadí  $I$ -tý přidaný –  $\mathcal{O}(\log n)$ .
- **GET\_PLACE( $X$ )** – vrátí číslo  $I$  označující jako kolikátý byl prvek  $X$  přidán do struktury –  $\mathcal{O}(\log n)$ .

**Datovka 2**. Elektikář si chce udržovat seznam klientů podle jejich identifikačních čísel (ID) a s údajem, zda se jedná o muže či ženu. Navrhněte datovku, která bude umět následující operace v  $\mathcal{O}(\log n)$ :

- **INSERT( $K, C$ )** – vloží nového klienta  $C$  s  $ID=K$ , označí ho jako ženu.
- **UPDATE( $K$ )** – klienta s  $ID=K$  přeznačí na muže.
- **FINDDIFF( $K$ )** – spočítaj rozdíl počtu mužů a žen mezi klienty s  $ID \leq K$ .

