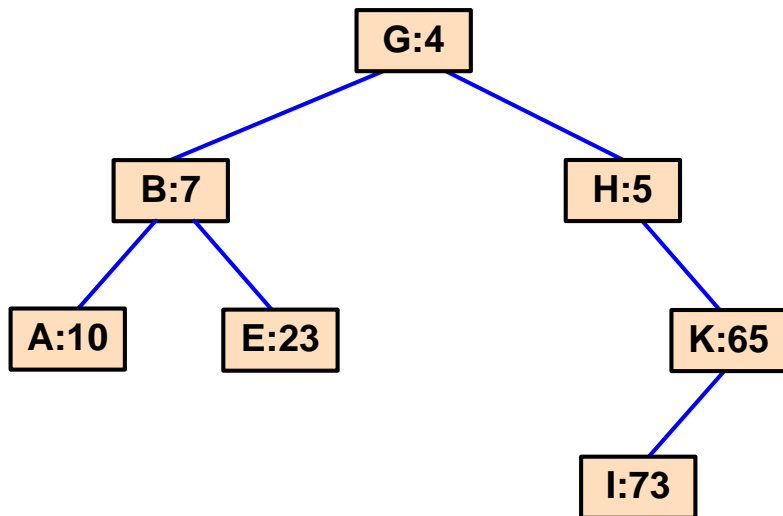


Treap

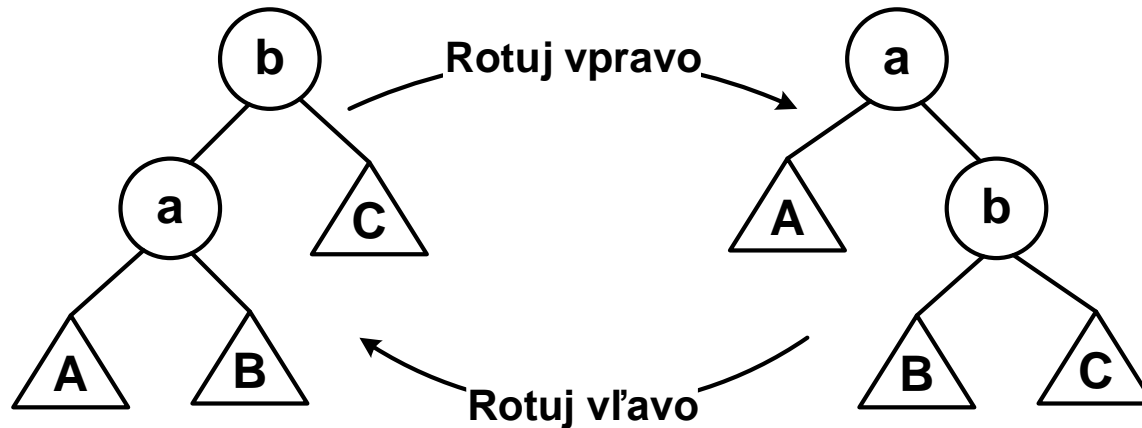
Binárny vyhľadávací strom s haldovým usporiadaním - (random) treap - pozostáva z prvkov, ktoré sú charakterizované jednak položkou **Kľúč** a jednak položkou **Priorita**. Treap vyhovuje aj vlastnostiam BVS, aj vlastnostiam binárnej haldy. Je implementáciou tabuľky. Na zabránenie degenerácie sa využíva náhodná priorita tak, aby bol strom „náhodnejší“. Štruktúra podporuje jednorozmerné intervalové vyhľadávanie.



Find (vyhľadanie) – priemerná zložitosť $O(\log_2 n)$, najhoršia zložitosť $O(n)$

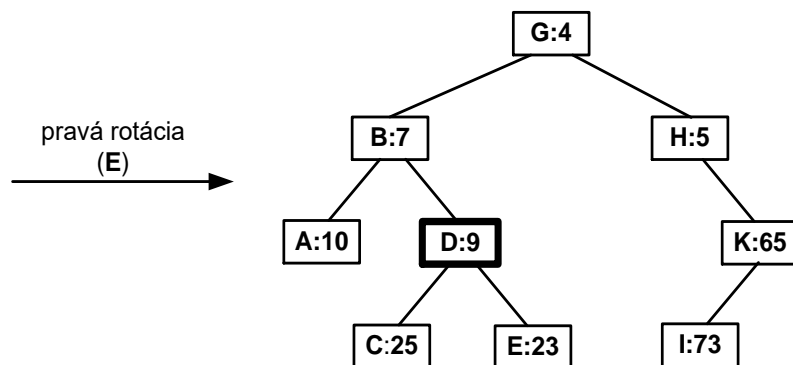
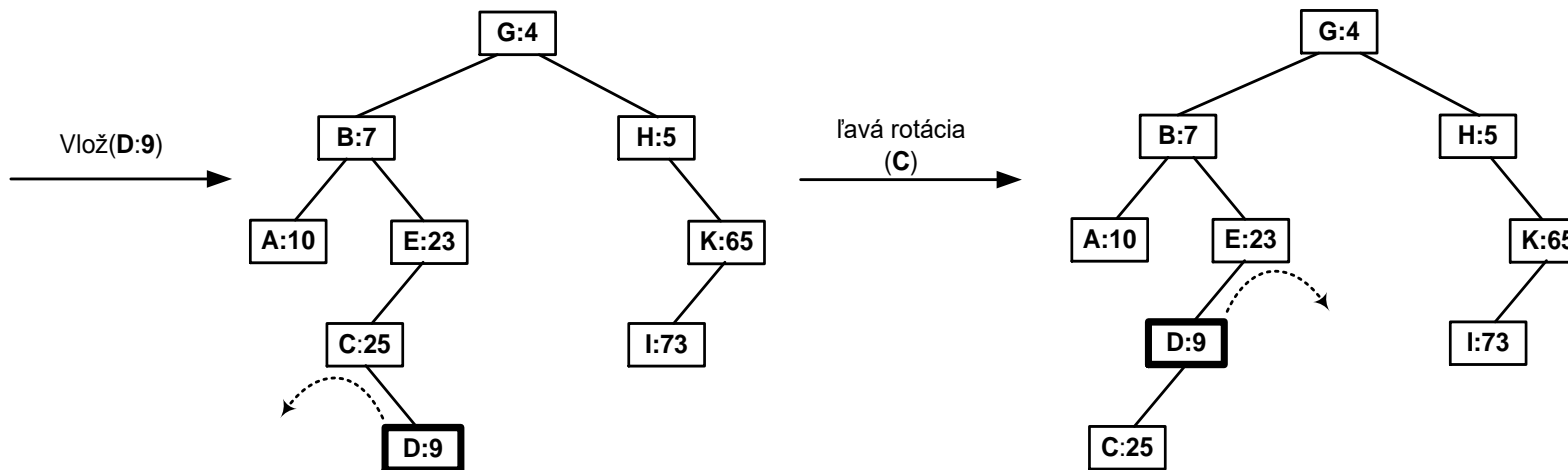
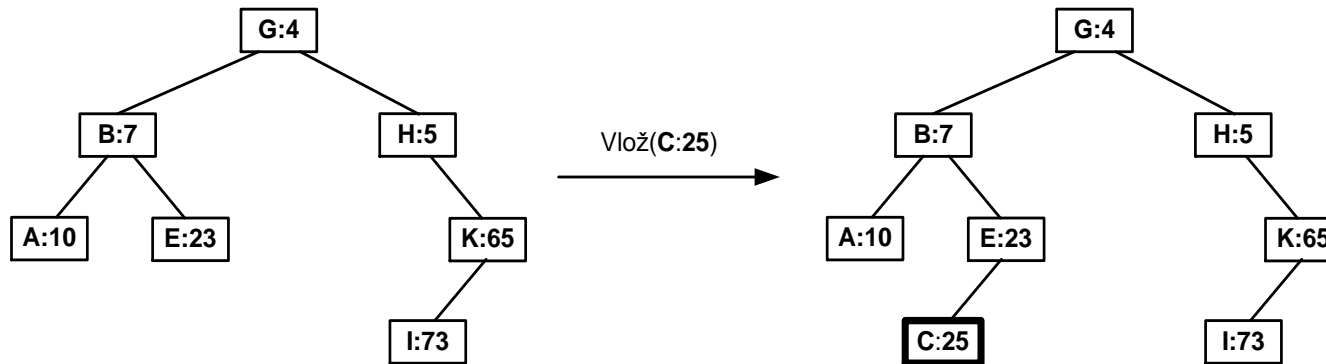
Vyhľadanie elementu **Prvok** podľa položky **Kľúč** využíva rovnakú stratégiu ako „štandardný“ BVS. Priemerná výška štruktúry treap je **$O(\log_2 n)$** , teda rovnakú priemernú zložitosť má i operácia **Nájdí**.

Dynamické operácie (Insert a Delete) využívajú pre konsolidáciu a vyvažovanie rovnaký algoritmus jednoduchšej rotácie jako AVL strom:

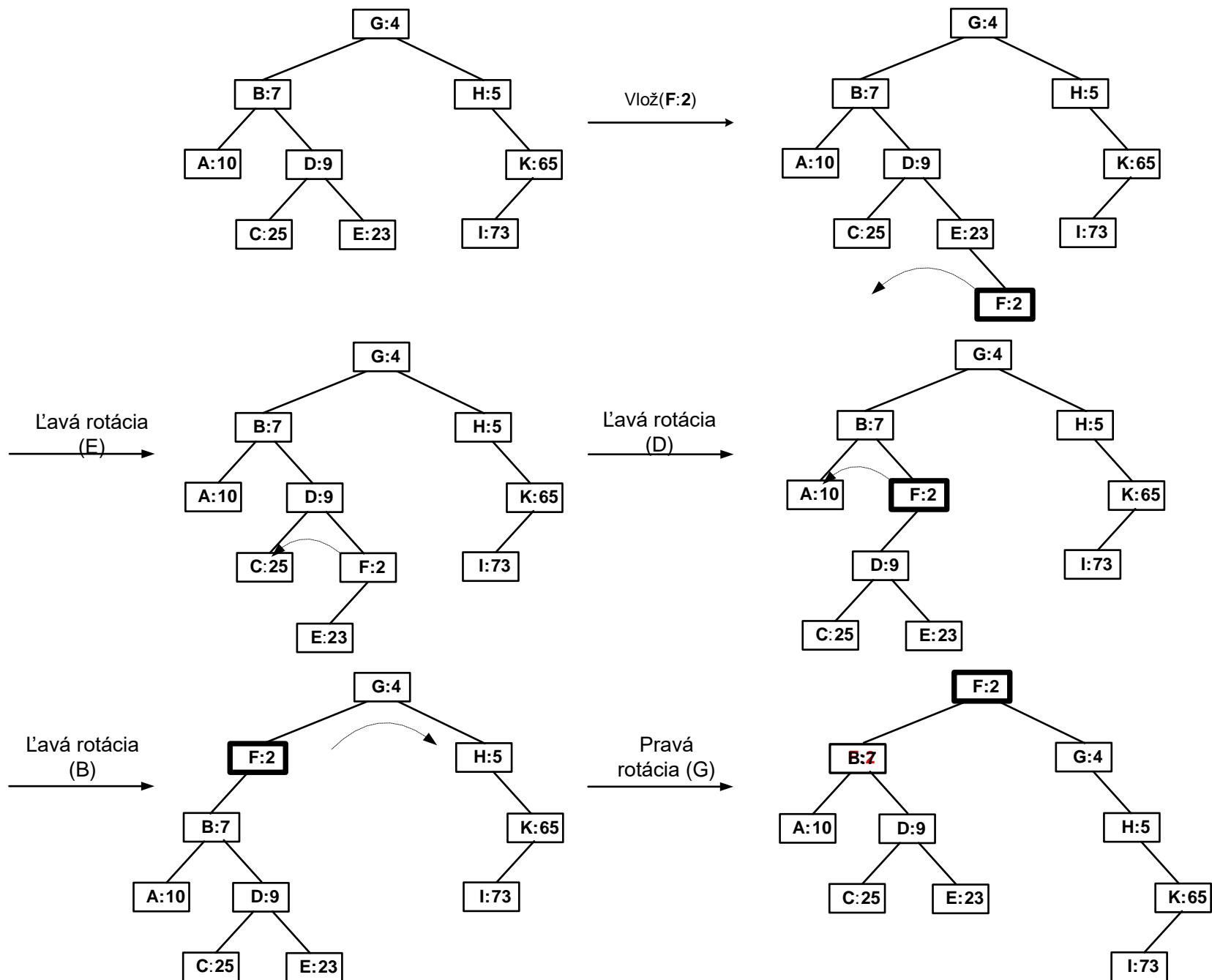


Insert (vkladanie) - priemerná zložitosť $O(\log_2 n)$, najhoršia zložitosť $O(n)$

- Vloženie elementu **Prvok** na pozíciu nového listu (**L**) stromu **S** – rovnaká stratégia ako pri BVS.
- Priradenie náhodnej priority prvku **Prvok** (randomized algorithm) a prvotná kontrola korektnej pozície prvku **Prvok** podľa pravidiel haldovej štruktúry.
- V prípade, že element **Prvok** nie je na korektnej „haldovej“ pozícii je postupne pomocou jednoduchých rotácií premiestňovaný smerom ku koreňu, až kým treap nevykazuje korektné haldové usporiadanie.



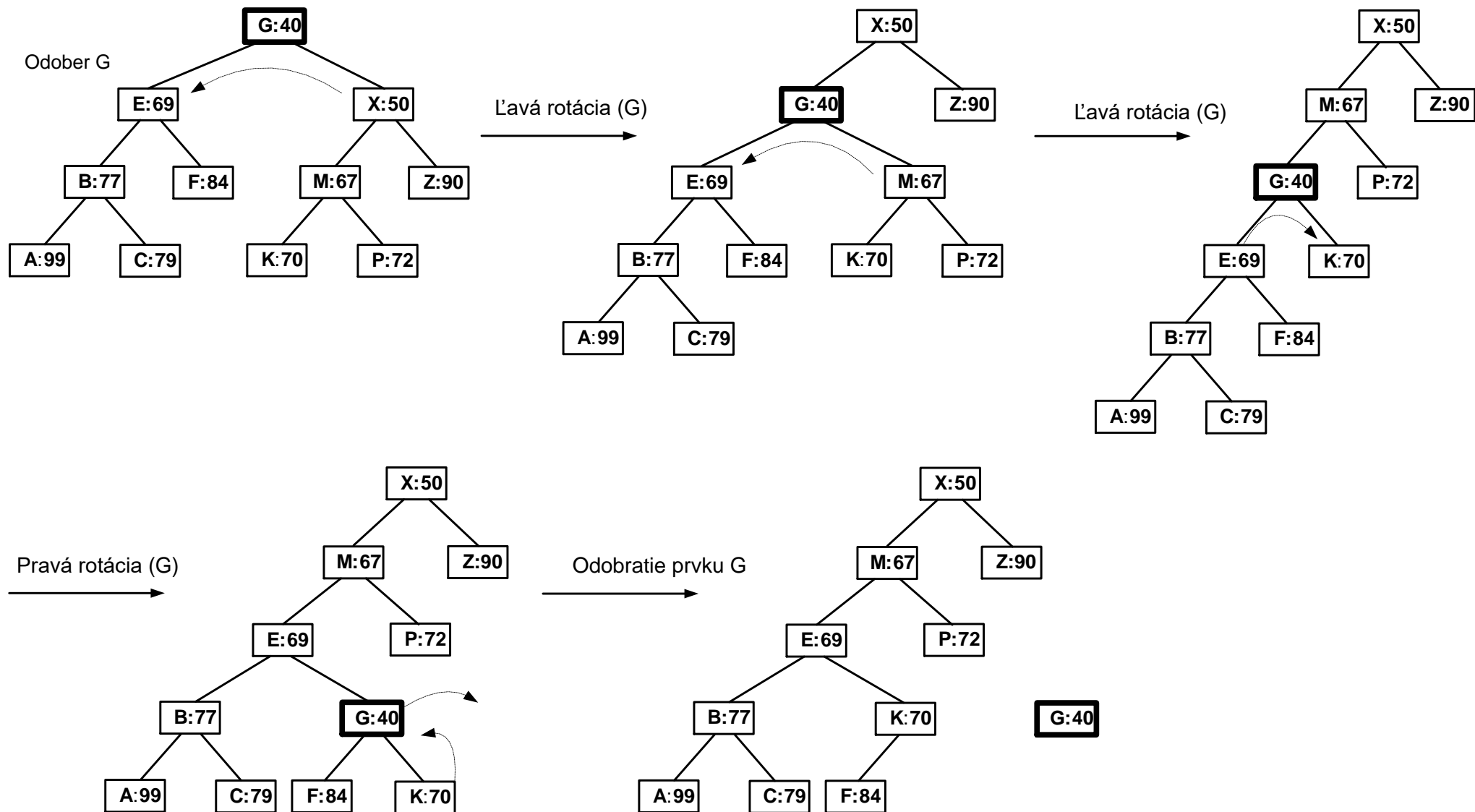
Materiál slúži výlučne pre študentov FRI ŽU, nie je dovolené ho upravovať, prípadne ďalej šíriť.



Delete (mazanie) - priemerná zložitosť $O(\log_2 n)$, najhoršia zložitosť $O(n)$

- a.) Ak je element **Prvok** listom stromu, tak ho odober bez následkov.
- b.) Ak sú synovia elementu **Prvok** listami stromu, tak sa **Prvok** odoberie a nahradí synom s vyššou prioritou.
- c.) Inak je element **Prvok** postupne premiestňovaný pomocou jednoduchých rotácií smerom od koreňa k listom, pokiaľ nie sú synovia elementu **Prvok** listami stromu - potom je aplikované reorganizačné opatrenie z prípadu b.).

Poznámka: Jednoduchá rotácia elementu **Prvok** smerom od koreňa k listu je aplikovaná tak (ľavá alebo pravá), aby sa po jej vykonaní stal otcom elementu **Prvok** ten z jeho synov, ktorý má vyššiu prioritu.



Zhrnutie:

- **Každý prvok** štruktúry treap je charakterizovaný **unikátnym kľúčom a prioritou**.
- Aj keď prvok treapu obsahuje prioritu, treap je implementáciou **tabuľky** (prístup podľa kľúča).
- Aby pravdepodobnosť opakovania hodnoty priority bola nulová, používame napr. v praxi generátor rovnomerne rozdelených náhodných (reálnych) čísel na veľkom intervale, napr. $(0, 2^{-31})$, nie je nutné zabezpečiť, aby bola priorita unikátna.
- Kvalita generátora je zárukou vyváženosti .
- Priradenie **náhodnej priority** vkladanému prvku do štruktúry treap radí stratégie príslušných treap-operácií do kategórie **algoritmov s náhodnými premennými (randomized algorithms)**.

Literatúra:

<http://compgeom.cs.uiuc.edu/~jeffe/teaching/373/notes/04-treaps.pdf>