7. cvičení z ADS1

1. 4. 2022

Příklad 1.

Roste strom vyhledávací
v paměti obsahující
mnoho čísel a máme k
jehož chceme následníka
následník k je číslo m
větší než k ve stromu našem

Příklad 2.

Představte si, že BVS použijeme na ukládání k-tic přirozených čísel, které porovnáváme lexikograficky. Jaká bude časová složitost jednotlivých operací? Kolik paměti tento strom zabere?

Příklad 3.

Uvažme, že v každém vrhcolu BVS máme kromě klíče uložené číslo. Pak chceme dělat dotaz, který vrátí součet čísel všech vrcholů s klíčem nejvýše k. Navrhněte způsob, jak tento dotaz vyhodnotit v čase $O(h)$, kde h je výška stromu. Strukturu BVS můžete libovolně upravovat.

Příklad 4.

Zkusme uvážit Poměrově vyvažovaný vyhledávací strom, ve kterém pro každý vrhcol v platí $1/2 \leq \frac{|l(v)|}{|r(v)|} \leq 2$. Dokažte, že takový strom má logaritmickou hloubku.

Příklad 5.

Zkusme naimplementovat vkládání vrcholu do Poměrově vyvažovaného BVS následně: Přidáme list a poté, pokud se pro libovolný v poruší vyváženost, tak zvolíme v nejblíže ke kořeni a celý jeho podstrom předěláme na perfektně vyvážený.

Ukažte, že takové vkládání vždy udrží BVS Poměrově vyvážený.

***Příklad 6.**

Ukažte, že vkládání do Poměrově vyváženého stromu popsané v předchozím příkladu má amortizovanou složitost $\mathcal{O}(\log n)$.

Příklad 7.

Navrhněte co nejrychlejší algoritmus, který pro dané k rozdělí BVS na dva tak, aby jeden obsahoval klíče menší než k a ten druhý větší nebo rovno k.

Příklad 8.

Mějme dva BVS velikostí n, m takové, že je chceme spojit do jednoho. Navrhněte algoritmus, který to udělá v čase $\mathcal{O}(n + m)$. Půjde to v čase $\mathcal{O}(\min(n, m))$?

Příklad 9.

Zkuste vymyslet způsob, jak pro určité volby m a n spojit dva stromy v čase lepším, než $\mathcal{O}(n + m)$.