

# 8. CVIČENÍ Z ADS 1

Viktor Němeček 8. 4. 2019

<https://kam.mff.cuni.cz/~viki/vyuka/ads11819/>

**Definice.** *Velikostí stromu* rozumíme počet jeho vrcholů. *Hloubkou* zakořeněného stromu délku nejdelší cesty z kořene do listu.

**Definice.** *Pravým, resp. levým podstromem* vrcholu  $V$  myslíme podstrom, jehož kořen je pravý, resp. levý syn vrcholu  $V$ .

**Definice.** *Perfektně vyvážený binární strom* je binární strom, kde pro každý vrchol platí, že velikost jeho pravého a levého podstromu se liší nejvýše o jedna.

**Příklad 1.** Dokažte, že perfektně vyvážený strom má všechny hladiny až na poslední zaplněné. Rozmyslete si, zda může ta poslední už vypadat jakkoli, či zd ajsou na ní nějaké další požadavky.

**Příklad 2.** Navrhněte algoritmus, který (v čase  $\mathcal{O}(n)$ ) spočítá, kolika způsoby lze postavit perfektně vyvážený strom na  $n$  prvcích. Uměli byste to i v polylogaritmickeém čase (tj. v čase  $\log^c n$  pro nějakou konstantu  $c$ )? Můžete počítat s tím, že s čísly o velikosti výsledku tohoto příkladu stále ještě umíte počítat v konstantním čase.

**Příklad 3.** Dokažte, že pro libovolný (i nevyvážený) BVS umíte udělat inorder průchod v čase  $\mathcal{O}(n)$ . Inorder průchod spočívá v tom, že navštívíte všechny vrcholy podle rostoucího pořadí klíčů

**Příklad 4.** Převedte libovolný strom na perfektně vyvážený v čase  $\mathcal{O}(n)$

\* **Příklad 5.** Vyřešte předchozí příklad pouze pomocí rotací a v paměti  $\mathcal{O}(1)$  (mimo samotného stromu). Můžete počítat s tím, že  $n = 2^k - 1$  pro nějaké  $k \in \mathbb{N}$ .

\*\* **Příklad 6.** Mějme model, v němž je pointer absolutně amorfní, tedy nelze s ním dělat nic jiného, než dereferencovat (například nemůžete vzít dva pointery a jejich adresy porovnávat menšítkem). V tomto modelu navrhněte datovou strukturu binárního stromu, tak, aby každý vrchol obsahoval nejvýše dva pointery, ale abyste v konstantním čase pro daný vrchol aniž byste využili klíčů:

- Zjistili, zda se jedná o kořen.
- Zjistili, zda má jednoho syna, dva či nula.
- Má-li dva, našli pravého a levého (musíte být schopní poznat, který je který).
- Má-li jednoho syna, našli tohoto syna (nemusíte poznat, zda je pravý či levý).
- Není-li vrchol kořen, našli jeho otce.

**Příklad 7.** Dokažte, že merge dvou BVS o velikostech  $n$  a  $m$  nemusí jít provést rychleji, než v čase  $\Theta(\min(m, n))$  (pro libovolně vysokou hodnotu  $\min(m, n)$ ), a to ani v případě, že vstupní stromy jsou dokonale vyvážené a výstupní být nemusí. Ukažte, že v  $\mathcal{O}(n + m)$  to naopak možné je vždy.

**Příklad 8.** Navrhněte co nejrychlejší algoritmus, který z jednoho BVS udělá dva tak, že v jednom z nich budou všechny hodnoty menší než nějaká konstanta, a ve druhém ty větší nebo rovný.

**Příklad 9.** V AVL stromu máme standardně v každém vrcholu uloženou informaci o vyvážení, která může nabývat tří různých hodnot. Ukažte, jak si vystačit pouze s jedním bitem v každém vrcholu.