

Algoritmy a datové struktury I

3. cvičení

čtvrtek 10. 3. 2016 9:00

1. Zkonstruujte dokonale vyvážený BVS ze setříděného pole v lineárním čase.
2. Jak v BVS nalezneme následníka daného prvku? Jak dlouho to trvá a jak dlouho trvá projít tímto způsobem celý celý strom od nejmenšího prvku?
3. Jak rychlá je operace nalezení k -tého nejmenšího prvku v AVL stromu? Umíte AVL strom upravit, aby tato operace byla rychlejší (bez asymptotického zpomalení jiných operací)?
4. Máte zadáno číslo k a na vstup vám přichází postupně prvky posloupnosti x_i . Vaším úkolem je udržovat hodnoty následujících funkcí pro posledních k prvků vstupu:
 - (a) součet
 - (b) maximum
 - (c) medián
5. Mějme AVL strom použitý jako slovník: v každém vrcholu sídlí klíč a nějaká celočíselná hodnota. Upravte strom, aby uměl zjistit nejmenší hodnotu přiřazenou nějakému klíči z intervalu $[a, b]$.
6. Pokračujme v předchozím cvičení: Také chceme, aby strom uměl ve všech vrcholech s klíči v zadaném intervalu $[a, b]$ zvýšit hodnoty o δ .

Domácí úkoly

Úkoly jsou za plný počet bodů 14 dnů od zadání (deadline je počátek cvičení), poté za polovinu bodů. Úkoly mi pošlete na husek+ads@iuuk.mff.cuni.cz.

1. Vymyslete algoritmus, který v lineárním čase přebuduje vyhledávací strom na dokonale vyvážený. Nemáte dost paměti na to, abyste si mohli pořídit ještě jednu kopii prvků – kromě stromu si smíte pamatovat $\mathcal{O}(\log n)$ hodnot. (Další 2 body, pokud si vystačíte s konstantní pamětí.) **[rebal, 10]**
2. Ukažte v příkladu 4c potřebujete alespoň čas $\Omega(\log k)$. **[kmed, 7]**
Hint: Ukažte, že pomocí algoritmu, který udržuje medián k posledních prvků v čase $\mathcal{O}(f(k))$ umíte třídit v čase $\mathcal{O}(n \cdot f(n))$.