

Algoritmy a grafy 1 (BI-AG1), Cvičení č. 6

Amortizovaná složitost, Binomiální halda, Vyhledávací stromy

Paralelka 104, Úterý 16:15-17:45

Cvičící: Šimon Lomič
lomicsim@fit.cvut.cz

Informace: [lomicsim.github.io](https://github.com/lomicsim)

Fakulta informačních technologií
České vysoké učení technické v Praze
<https://courses.fit.cvut.cz/BI-AG1>



6.1 Amortizovaná složitost

- **Amortizovaná časová složitost** $O^*(f(n))$ operace A v jistém kontextu \Leftrightarrow posloupnost k operací A má celkovou časovou složitost $O(k \cdot f(n))$.

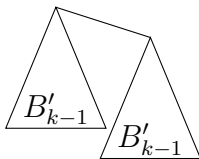
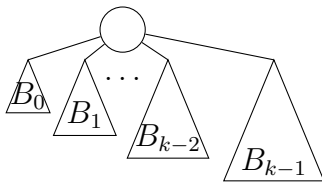
Cvičení:

- Simulujte frontu pomocí dvou zásobníků. Určete amortizovanou složitost.
- Uvažujme nafukovací pole z přednášky, nechť c je konstanta, m aktuální kapacita pole. Jak by se změnila amortizovaná složitost, kdybychom při naplnění pole prováděli realokaci na velikost:
 - ▶ $m \rightarrow c \cdot m$
 - ▶ $m \rightarrow c + m$

6.2 Binomiální strom

Binomiální strom řádu k (značíme B_k) je uspořádaný (záleží na pořadí synů) zakořeněný strom, pro který platí:

- 1 B_0 je tvořen pouze kořenem.
- 2 Pro $k \geq 1$ získáme B_k ze stromů B_0, B_1, \dots, B_{k-1} tak, že přidáme nový kořen a kořeny těchto stromů uděláme (takto popořadě) syny nového kořene.



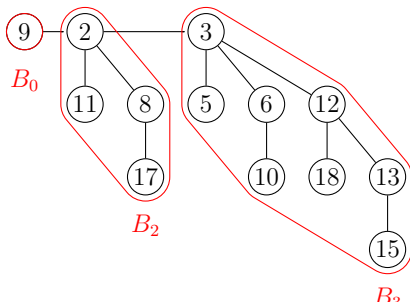
Cvičení:

- (a) Jaký je maximální stupeň B_k ?
- (b) Dokažte, že B_k má 2^k vrcholů.
- (c) Dokažte, že B_k má v i -té vrstvě $\binom{k}{i}$ vrcholů.

6.3 Binomiální halda

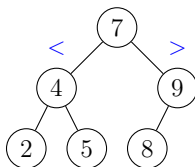
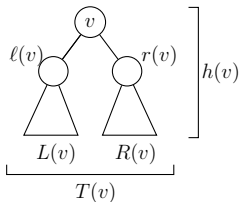
- **Binomiální halda:** soubor binomiálních stromů $\mathcal{T} = T_1, \dots, T_\ell$:
 - ▶ V každém vrcholu v je uložen **prvek** $k(v)$.
 - ▶ **Haldové uspořádání:** pro každý vrchol v a jeho libovolného syna s platí $k(v) \leq k(s)$.
 - ▶ Soubor stromů \mathcal{T} je uspořádán **vzestupně** podle jejich řádů (řády jsou **unikátní**).
 - ▶ Logaritmičké operace GetMin, Insert, ExtractMin a Merge.

Cvičení: Navrhněte operace DecreaseKey, Delete, IncreaseKey.



6.4 Binární vyhledávací strom

- **Binární strom** je zakořeněný strom, kde každý vrchol má nejvýše dva syny, u nichž rozlišujeme, který je levý a který pravý.
- **Binární vyhledávací strom** (BVS) je binární strom, kde v každému vrcholu v je uložen unikátní klíč $k(v)$ tak, aby $\forall a \in L(v) : k(a) < k(v)$ a $\forall b \in R(v) : k(b) > k(v)$.



Cvičení:

- Jaká je minimální a maximální hloubka BVS (délka nejdelší cesty z kořenu do listu).
- Navrhněte algoritmus, který vypíše všechny klíče v BVS v rostoucím pořadí. Lze to provést v $O(1)$ paměti?

6.5 Vyvážený vyhledávací strom

- BVS je **dokonale vyvážený**, pokud pro každý jeho vrchol v platí $||L(v)| - |R(v)|| \leq 1$.
- BVS je **hloubkově vyvážený** pokud pro každý jeho vrchol v platí $|h(\ell(v)) - h(r(v))| \leq 1$ (**AVL strom**).

Cvičení:

- Proč nelze udržovat dokonale vyvážený BVS?
- Jak postavit *dokonale* vyvážený BVS ze setříděného pole?
- Navrhněte algoritmus, který *dokonale* vyváží zadaný BVS.

6.6 Domácí úkol (0.5 b)

Vyřešte následující úlohu: <http://www.hackerrank.com/challenges/jesse-and-cookies/problem>.

Implementaci, která prošla testy na HackerRank, odevzdejte e-mailem na lomicsim@fit.cvut.cz.