

1. CVIČENÍ Z DATOVEK 1, ZS25/26

Amortizujeme!

1. *Asymptotická složitost*: Roztříděte následující funkce do skupin stejně rychle rostoucích (pro každé f a g platí $f = \Theta(g)$) a následně porovnejte skupiny pomocí o a ω :

n , $42n + 7$, n^2 , $\log n$, $\log(n^2)$, $(\log n)^2$, $\sqrt{n}$, 2^n , 2^{2n} , 4^n , $2^{n \log n}$, $2^{2 \log n}$, $2^{(\log n)^2}$, n^n , $n!$, $(n + 1)!$.

(Logaritmy mají dvojkový základ. Najdete v zadání formální chybu? :-)

2. *Iterovaný následník*. Najdeme v BVS vrchol s minimálním klíčem (jak?) a poté $n - 1$ krát provedeme operaci nalezení následníka. Jaká bude celková časová složitost? Jinými slovy, jaká bude amortizovaná složitost na operaci následníka?

3. *Nafukování pole jinak*. Na přednášce bylo nafukovací pole (zásobník), u něhož při zaplnění zdvojnásobíme velikost a tím dosáhneme amortizované složitosti $O(1)$ na vložení. Co kdybychom při zaplnění místo toho:

- a) zvětšili velikost pole o konstantní počet prvků?
- b) zvětšili velikost pole z m na m^2 ?
- c) zvětšili velikost pole k krát pro parametr $k > 1$? (Bonus: jak přesně závisí amortizovaná složitost na k ?)

4. *Dynamické pole aneb zásobník polem*. Připomeňte si, jak vytvořit pole, do něhož přidáváme na konec a také z konce prvky odebíráme tak, abychom si udržovali konstantní naplnění (tedy např. 10% kapacity je využito). Analyzujte amortizovanou složitost pomocí vhodného potenciálu.

5. *Fronta pomocí dvou zásobníků*. Mějme dva zásobníky neomezené kapacity (vkládání/odebírání probíhá jen z vrchu zásobníku). Jak pomocí nich implementovat frontu (pouze s konstantní pomocnou pamětí)? Jakou amortizovanou složitost mají operace s frontou?

6. *Dokonalé vyvážení*. Navrhněte algoritmus, který ze setříděného pole vyrobí v lineárním čase dokonale vyvážený BVS. (Pro každý vrchol se počet vrcholů v levém podstromu liší od počtu vrcholů v pravém podstromu maximálně o 1.)

7. *Oboustrané nafukovací/dynamické pole.* Zatím jsme přidávali (nebo odebírali) jenom na konci pole. Šlo by konstrukci upravit tak, abychom mohli přidávat i na začátek a zachovali jsme přitom složitost? A co kdybychom chtěli prvky z konce (či začátku) mazat?

8. *Binární počítadlo s odčítáním.* Na přednášce byly tři důkazy, že binární počítadlo přičítá $+1$ (operace INC) v amortizovaně konstantním čase, kde čas je počet změn bitů.

- a) Ukažte, že pokud bychom povolili odčítání -1 (DEC), amortizovaná časová složitost se zhorší.
- b) Navrhněte jinou reprezentaci celých čísel, v níž bude možné provádět operace INC, DEC a TESTZERO (vrátí, jestli je číslo nulové nebo ne) v amortizovaně konstantním čase.

9. *Víc než jen inkrement.* Uvažujme místo INC operaci $\text{ADD}(k)$, která k počítadlu přičte číslo k . Dokažte, že amortizovaná složitost této operace je $O(\log k)$.

10. *Okénková minima amortizovaně konstantně:* Na vstupu postupně přicházejí čísla v nekonečném proudu. Kdykoliv přijde další, vypište minimum z posledních k čísel. Cílem je dostat amortizovaně konstantní čas na jedno číslo.