

2. CVIČENÍ Z DATOVEK 1, ZS23/24

„Óčka“, amortizace a BVS

1. *Asymptotická rozčvička:* Připomeňte si „óčkovou“ notaci O , Ω , Θ , o a ω . Roztřídte následující funkce do skupin stejně rychle rostoucích (pro každé f a g v jedné skupině platí $f = \Theta(g)$): 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 , $n!$, $(n + 1)!$.

2. *Iterovaný následník.* Najdeme v binárním vyhledávacím stromu (BVS) vrchol s minimálním klíčem (jak?) a poté $n - 1$ krát provedeme operaci nalezení následníka. Jaká bude celková časová složitost?

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

- zvětšili velikost pole o konstantní počet prvků?
- zvětšili velikost pole z m na m^2 ?

4. *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?

5. *Binární počítadlo s odčítáním.* Na přednášce byl důkaz, že binární počítadlo přičítá $+1$ (operace INC) v amortizovaně konstantním čase.

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

6. *Intervalový update.* Mějme BVS jako slovník dvojic (klíč, hodnota), přičemž hodnoty jsou číselné. Upravte jej, aby podporoval operaci $\text{add}(x, y, \delta)$, která k hodnotám všech klíčů v intervalu $[x, y]$ přičte δ . Tato operace má běžet v $O(h)$, kde h je hloubka BVS, takže nemusíme hned provést aktualizaci hodnot všech klíčů v daném intervalu, stačí když $\text{Find}(k)$ vrátí správnou hodnotu klíče k .