koutecky+ads1@iuuk.mff.cuni.cz

8. cvičení

Součet. Mějme množinu přirozených čísel a číslo x. Chceme zjistit, zda množina obsahuje dvojici prvků se součtem x.

Minimové okénko. Na vstupu postupně přicházejí čísla. Kdykoliv přijde další, vypište minimum z posledních k čísel. Umíte to v $\mathcal{O}(\log n)$? A co teprve v $\mathcal{O}(1)$? To je dost těžké, ale zkuste to amortizovaně v $\mathcal{O}(1)$:-)

Koloniál. Ve Frantově koloniále přicházejí zákazníci a zadávají do fronty objednávky; objednávka je trojice (zboží, množství, jméno zákazníka). Koloniálník Franta by měl rád přehled o tom, zda má na skladě dost příslušného zboží. Navrhněte datovou strukturu pro jeho koloniál, která bude v čase $\mathcal{O}(1)$ vykonávat operace:

(1) `ENQUEUE(`R`)`— zařadí objednávku R
(2) `DEQUEUE()` — vypíše následující objednávku
(3) `QUERY(`P`)`— pro produkt P vypíše celkové objednané množství tohoto produktu.

(Předpokládám, že znáte strukturu pro FIFO frontu.) Máte zaručeno, že ve frontě nebude nikdy více než m objednávek, a víte, že v koloniále je celkem n druhů zboží. Najdete řešení v prostoru $\mathcal{O}(n)$? A co $\mathcal{O}(m)$, pro případ, že $m \ll n$?

Nafukovací pole.

(1) Uvažujme, že bychom pole místo zdvojnásobování nafukovali o konstantu. Dokažte, že se tím pokazí časová složitost.
(2) Co kdybychom m-prvkové pole zvětšovali rovnou na m^2? Počáteční velikost musíme samozřejmě zvětšit na konstantu větší než 1.

Fronta. Ukažte, jak pomocí dvou zásobníků simulovat frontu. Snažte se o amortizovaně konstantní časovou složitost operací (za předpokladu, že zásobník pracuje v konstantním čase).

Čítač s `DEC`. Uvažme n-bitový čítač, který podporuje operace `INC` a `DEC` (zvýšení a snížení o 1) a `TESTZERO` (zjisti, zda je číslo nulové). Modifikujeme ho tak, že každý bit může nabývat hodnot 0, +1 a -1. Ukažte, že v této reprezentaci je amortizovaná složitost obou operací $\mathcal{O}(1)$.