

1. *Změna nafukování:*

- (a) Uvažujme, že bychom pole namísto zdvojnásobování zvětšovali o konstantní počet prvků. Dokažte, se tím pokazí časová složitost.
- (b) Jak by to dopadlo, kdybychom m -prvkové pole rovnou zvětšovali na m^2 -prvkové? Počáteční velikost musíme samozřejmě zvýšit na konstantu větší než 1.

2. *Zásobníková fronta:* K dispozici jsou dva zásobníky, které podporují pouze operace PUSH (přidej na vrchol zásobníku) a POP (odeber z vrcholu zásobníku). Navrhněte algoritmus, který bude pomocí těchto dvou zásobníků simulovat frontu s operacemi ENQUEUE (přidej na konec fronty) a DEQUEUE (odeber z počátku fronty). Kromě zásobníků máte k dispozici pouze konstantní množství paměti. Ukažte, že operace s frontou budou mít amortizovaně konstantní časovou složitost.3. *(Od)čítač:*

- (a) Rozmyslete, že kdyby mělo obyčejné binární počítadlo (z přednášky) podporovat zároveň operace INC a DEC (tedy zvýšení a snížení o 1), operace rozhodně nebudou mít konstantní amortizovanou složitost.
- (b) Navrhněte jinou reprezentaci čísel, v níž bude možné provádět operace INC, DEC a TESTZERO (zjistí, zda číslo je nulové) v amortizovaně konstantním čase. Nápoděda: povolte krom 0,1 i -1.

4. *Pokročilé počítadlo:* Uvažujme místo INC operaci $\text{ADD}(k)$, která k počítadlu přičte číslo k . Dokažte, že amortizovaná složitost této operace je $O(\log k)$.

5. **FIND SUCC**: V binárním vyhledávacím stromu budeme provádět operace **FIND** (nalezení prvku se zadaným klíčem) a **SUCC** (nalezení následníka prvku, který nám vrátila předchozí operace **FIND** nebo **SUCC**). Najděte potenciál, vůči kterému vyjde amortizovaná složitost **FIND** $O(\log n)$ a **SUCC** $O(1)$.
6. *Okénkový medián*: Dokažte, že okénkový medián nejde lépe než v $\Omega(\log k)$ čase pokud si povolíme čísla pouze porovnávat.
7. *Seznamy polí*: Inspirujte se binárním počítadlem k vyřešení následujícího problému. Chceme datovou strukturu, která umí udržovat množinu čísel a operace **INSERT** amortizovaně $O(1)$ a **FIND** ve worst-case čase $O(\log^2 n)$. Náповěda: Udržujte si seznam setříděných polí délky 1, 2, 4, 8, 16, ... (každé buďto plné nebo prázdné), a chovejte se k ni stejně jako kdybyste binárním počítadlem počítali celkový počet prvků ve struktuře.