

## 2. PÁTEČNÍ CVIČENÍ Z DATOVÝCH STRUKTUR 1

Dynamická pole a BVS

1. *Dynamické pole jinak.* Co by se stalo, kdybychom zvyšovali kapacitu jinak, než na dvojnásobek? Uvažte následující možnosti ( $p > 1$  je původní kapacita):

- a)  $p \rightarrow p + k$ , kde  $k$  je konstanta (např.  $k = 1$ ),
- b)  $p \rightarrow p^2$ ,
- c)  $p \rightarrow k \cdot p$  pro libovolnou konstantu  $k > 1$ .

2. *Mazání z dynamického pole.* Jak upravit dynamické pole, abychom z něho mohli i mazat?

3. *Ještě lepší dynamické pole.* Do teď jsme přidávali nové prvky jen na konec pole. Šlo by konstrukci upravit tak, abychom mohli přidávat i na začátek (při zachování složitosti)? A co kdybychom chtěli prvky z konce (začátku) i mazat?

4. *Binární počítadlo s odčítáním.* Připomeňte si důkaz, že binární počítadlo (na začátku nastavené na 0) má amortizovanou konstantní složitost přičítání jedničky. (Kolik takových důkazů znáte?)

Jak se změní amortizovaná složitost, pokud povolíme přičítání i odčítání jedničky? Lze upravit počítadlo, aby zůstala konstantní?

5. *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  $\delta$ . Tato operace má běžet v  $O(h)$ , kde  $h$  je hloubka, takže nemusíme hned provést aktualizaci hodnot všech klíčů v daném intervalu, stačí když  $\text{Find}(k)$  vrátí správnou hodnotu klíče  $k$ .

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.)