

1. cvičení

Datové struktury I, 2. 10. 2025

<https://iuuk.mff.cuni.cz/~chmel/2526/ds1/>

Úloha 1 (Asymptotika)

Roztříd'te následující funkce do skupin stejně rychle rostoucích funkcí (tj. pro všechny f, g v jedné skupině platí $f = \Theta(g)$) a následně porovnejte tyto skupiny pomocí o a ω : n , $42n + 7$, n^2 , $\log n$, $\log_e n$, $\log(n^2)$, $(\log n)^2$, $\sqrt{n}$, 2^n , 2^{2n} , 4^n , $2^{n \log n}$, $2^{2 \log n}$, n^n , $n!$, $(n+1)!$.

Všechny logaritmy bez explicitního základu mají dvojkový základ.

Úloha 2 (Iterace následníka)

Najdeme v BVS (s n vrcholy) vrchol s minimálním klíčem, a poté $(n-1)$ -krát provedeme operaci nalezení následníka. Jaká bude celková časová složitost?

Úloha 3 (A teď oboustranně a se smrštěním)

Zatím jsme přidávali jenom na konec 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?

Úloha 4 (Natahujeme pole)

Ukázali jsme si, jak implementovat natahovací pole, které funguje v amortizovaně konstantním čase, s tím, že vždycky, když je pole plné, a potřebujeme přidat další prvek, všechno překopírujeme do pole dvojnásobné velikosti.

Co by se stalo, kdybychom místo zdvojnásobování kapacity C zvyšovali jinak?

a) $C \rightsquigarrow C + k$, kde $k \geq 1$ je konstanta,

b) $C \rightsquigarrow C^2$,

c) $C \rightsquigarrow k \cdot C$ pro konstantu $k > 1$.

Bonusové úlohy

Úloha 5 (Pozorné čtení)

Najdete v zadání cvičení Asymptotika formální chybu?

Úloha 6 (Perfectly balanced, as all things should be)

Navrhněte algoritmus, který ze seřazeného pole v lineárním čase vytvoří dokonale vyvážený BVS. (Tedy pro každý vrchol musí platit, že počet vrcholů v levém podstromu se od počtu vrcholů v pravém podstromu smí lišit maximálně o 1.)

Úloha 7 (Intervalový update)

Mějme BVS jako slovník dvojic (klíč, hodnota) s číselnými hodnotami. 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 $\mathcal{O}(h)$, kde h je hloubka stromu. To znamená, že nemusíme hned aktualizovat hodnoty všech klíčů v intervalu. Stačí, když operace $\text{FIND}(k)$ vrátí správnou hodnotu klíče k .