Cvičení 10 – Hešování, amortizace

Intervalové stromy

Úloha 1 *(Nejbližší větší napravo)* Naučte intervalový strom zjistit nejbližší prvek, který leží napravo od zadaného listu a obsahuje větší hodnotu. Jinými slovy: na vstupu dostaneme index i v posloupnosti a chceme najít nejmenší $j > i$ takové, že $x_j > x_i$.

Úloha 2 *(Intervalový update)* Naučte (součtový) intervalový strom operaci $\textsc{Update}(i, j, \delta)$, která k hodnotám $x_i, x_{i+1}, \ldots, x_j$ reprezentované posloupnosti přičte δ. Jelikož těchto hodnot může být až $\Omega(n)$ a my chceme logaritmickou časovou složitost, stačí jen upravit strom tak, aby i po provedení této operace vracel správné výsledky pro dotazy na součet intervalu.

Hešování

Úloha 3 Dostali jste hešovací funkci $h : \mathcal{U} \to [m]$. Pokud o této funkci nic dalšího nevíte, kolik nejvýše vyhodnocení funkce potřebujete, abyste našli k-tici prvků, které se všechny zobrazí do téže přihrádky?

Úloha 4 *(Dvojice s daným součtem)* Mějme množinu přirozených čísel a číslo x. Chceme zjistit, zda množina obsahuje dvojici prvků se součtem x.

Úloha 5 *(Úprava lineárního přidávání)* Uvažujme hešování s otevřenou adresací řízené obecnější lineární posloupností $h(x, i) = (f(x) + c \cdot i) \bmod m$, kde c je konstanta nesoudělná s m. Srovnejte jeho chování s obyčejným lineárním přidáváním.

Jakub Komárek
komarek+ads1@iuuk.mff.cuni.cz
https://jakoma02.cz/teaching/ls24/ads1/

Úloha 6* *(Mimozemský narozeninový paradox)* Mějme planetu s m dny v roce a skupinu n obyvatel této planety, kteří mají rovnoměrně náhodně narozeniny v tamním roce. Zkuste co nejlépe spočítat pravděpodobnost, že dva mimozemšťané v této skupině mají narozeniny ve stejný den.

Amortizace

Úloha 7 *(Binární počítadlo)* Mějme číslo n zapsané binárně (jako pole bitů). Jak dlouho trvá operace Inc, která přičte k n jedničku? Jak dlouho trvá n-krát opakovaná operace Inc, pokud začneme s počítadlem nastaveným na 0?

Úloha 8 *(Nafukování pole jinak)* Na přednášce jste měli, že pokud opakovaně přidáváme prvky do hešovací tabulky (nebo jen pole), a vždy když dosáhneme jistého zaplnění, zdvojnásobíme velikost tabulky v čase lineárním s aktuálním počtem prvků, dosáhneme amortizované složitosti $\mathcal{O}(1)$ na vložení. Co kdybychom:

a) zvětšili velikost tabulky o konstantní počet prvků?
b) zvětšili velikost tabulky z m na m^2?

Úloha 9 *(Fronta pomocí zásobníků)* Mějte dva zásobníky neomezené kapacity. Jak pomocí nich implementovat frontu? Pokuste se dosáhnout amortizované složitosti $\mathcal{O}(1)$ na operaci.

Úloha 10* *(Postupné zvětšování)* Vymyslete, jak nafukovací pole upravit tak, aby složitost vložení byla $\mathcal{O}(1)$ v nejhorším případě (tj. neamortizovaně). Předpokládejte, že alokace paměti má konstantní složitost.

Jakub Komárek
komarek+ads1@iuuk.mff.cuni.cz
`https://jakoma02.cz/teaching/ls24/ads1/`