10. cvičení z ADS1

21. 4. 2022

Příklad 1.

Vymyslete, jak (za předpokladu, že abeceda je konstantně velká) seřadit posloupnost řetězců v čase lineárním vzhledem k součtu jejich délek.

Příklad 2.

Pole je hezká datová struktura – umíme indexovat prvky v konstantním čase. Dokonce i přidávání na konec trvá konstantně dlouho.

Jenže na reálných počítačích má pole omezenou *kapacitu* – potřebujeme, aby blok paměti byl souvislý. Proto se může stát, že budeme muset při přidání prvků celé pole realokovat s větší kapacitou a překopírovat všechny prvky (v lineárním čase).

Nejprve uvažme, že kapacitu při přetečení vždy zvětšíme o konstantu. Pak místo zvětšování o konstantu uvažme, že budeme kapacitu zdvojnásobovat. Jak se tyto dvě strategie chovají? Kolik uděláme celkem práce, pokud na začátku máme kapacitu c a vložíme n prvků?

Příklad 3.

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

Příklad 4.

Mějme hešovací funkci $f : \mathcal{U} \to [m]$, o které nic nevíme. Kolik prvků musí tabulka velikosti m alespoň obsahovat, aby určitě bylo v jedné přihrádce alespoň k prvků? Po kolika prvcích nejméně bude pravděpodobnost takové události nenulová?

Příklad 5.

Jak byste pro hešování s otevřenou adresací naimplementovali operaci Delete, pokud chcete, aby po nalezení políčka, ze kterého se má mazat, už operace trvala amortizovaně $\mathcal{O}(1)$?

Příklad 6.

Nechť $h_{a,b}(x) = ((ax + b) \bmod p) \bmod m)$. Potom systém hešovacích funkcí $\mathcal{L} = \{h_{a,b} \mid a, b \in [p], a \neq 0\}$ je 1-univerzální.

Ukažte, že pokud v systému $\mathcal{L}$ zafixujeme parametr b na nulu, bude 2-univerzální. Podobně ukažte, že pokud dovolíme nulové a, bude systém taky 2-univerzální.

Definice.

Bloomův filtr je datová struktura pro přibližnou reprezentaci množiny. Skládá se z pole bitů $B[1, \ldots, m]$ a hešovací funkce h, která prvkům univerza přiřazuje indexy v poli. Operace Insert(x) nastaví $B[h(x)]$ na 1; Member(x) otestuje, zda $B[h(x)] = 1$.

Příklad 7.

Vložme nyní do Bloomova filtru nějakou n-prvkovou množinu M. Pokud $x \in M$, operace Member(x) vždy odpoví správně. Pokud se ale zeptáme na $x \notin M$, může se stát, že $h(x) = h(y)$ pro nějaké $y \in M$ a dostaneme špatnou odpověď. Spočítejte, s jakou pravděpodobností se to pro dané m a n stane.