9. CVIČENÍ Z DATOVÝCH STRUKTUR 1, ZS23/24

Opět hešování (hurá!): řešení kolizí lineárním přidáváním a kukačkou

1. *Rozcvička s černou skříňkou.* Dostali jste hešovací funkci funkci $h : \mathcal{U} \to [m]$, o které nic nevíte, pouze můžete nechat vyhodnotit $h(x)$ pro zadané $x \in \mathcal{U}$. Kolik vyhodnocení funkce potřebujete, abyste zaručeně našli k-tici různých prvků, která se zahešuje do jedné přihrádky?

Řešení. $m \cdot (k-1)+1$ vyhodnocení, neboť všechny buňky mohou být zaplněny $k-1$ prvky než najdeme požadovanou k-tici

2. *Lineární přidávání s většími skoky.* Uvažujme hešování s otevřenou adresací řízené obecnou lineární posloupností $h(x,i) = (f(x) + c \cdot i) \mod m$, kde c je konstanta nesoudělná s velikostí tabulky m. Srovnejte jeho chování s obyčejným lineárním přidáváním.

Řešení. Pro libovolné c nesoudělné s m se chová stejně jako běžné lineární přidávání se skokem 1 — také se tvoří srůstající řetězce, jen pro permutaci indexů danou přičítáním c. Pro $c = 1$ dostáváme ostatně běžné lineární přidávání. Pro $c > 1$ se však tento přístup chová hůře ke keším.

3. *Opravdové mazání pro lineární přidávání.* Na přednášce bylo řešení kolizí pomocí lineárního přidávání s tím, že pokud prvek mažeme, pouze ho označíme za smazaný („postavíme tam pomníček“). Zkuste domyslet detaily a alternativní řešení:

a) Pokud provedeme hodně operací mazání, bude hešovací tabulka obsahovat více označených (tj. smazaných) prvků než nesmazaných. Co provést v takovém případě? Jak zajistit konstantní amortizovanou časovou složitost (ve střední hodnotě) pro libovolnou sekvenci operací Insert a Delete? (Můžete předpokládat větu z přednášky o střední hodnotě délky řetízků.)

b) Alternativní způsob: mazaný prvek skutečně smažeme a poté vhodně přesuneme nějaké prvky. Vymyslete, jak to přesně udělat, abychom pak mohli vyhledat všechny nesmazané prvky (tedy abychom tabulku „nerozbili“).

Řešení.

a) Pokud tabulka obsahuje (např.) polovinu prvků, které jsou již označené za smazané, tak vše přehešujeme. To má pořád amortizovanou konstantní složitost (v průměru a za předpokladu, že srůstající řetízky jsou v průměru konstantně dlouhé).

b) Po smazání prvku nám může vzniknout mezera v řetízku a pak bychom nemohli vyhledat prvky, které jsou v řetízku za mezerou. Cílem je posunout celý řetízek od smazaného prvku dál o jedno doleva (indexy modulo m), je akorát potřeba dávat pozor na to, že nesmíme posunout prvek před pozici, kam se zahešuje.