

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 $h : \mathcal{U} \rightarrow [m]$, 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 prvků, která se zahešuje do jedné přihrádky?

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

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“).