11. CVIČENÍ Z DATOVÝCH STRUKTUR 1, ZS24/25

Bloomovy filtry aneb jak řešit domácí úkol na hledání duplikátů + „rolling hash“

1. *Počítací Bloom filtr.* Uvažme Bloom filtr, který umožňuje mazání tím, že v políčku je místo jednoho bitů b-bitové počítadlo pro malé b, např. $b = 4$. Toto počítadlo ale nesmí přetéct, takže když se dostane na hodnotu $2^b - 1$, zasekne se a už se nemění. Zanalyzujte pravděpodobnost, že jedno konkrétní počítadlo v jedné tabulce se zasekne po vložení n prvků. Můžete předpokládat, že velikost tabulky je $m \geq \ln 2 \cdot n$.

Jak odhadnout pravděpodobnost, že nějaké počítadlo přeteče?

Nyní hurá na řetězcové algoritmy:

2. *Polynomiální hešování řetězců aneb „rolling hash“.* Pro prvočíslo p a délku vektoru (řetězce) d definujme třídu hešovací funkcí $\mathcal{R} = \{h_a : a \in \mathbb{Z}_p\}$, kde $h_a(x_0x_1 \dots x_{d-1}) = \sum_{i=0}^{d-1} x_i a^{d-i-1} \mod p$.

a) Dokažte, že $\mathcal{R}$ je $d-1$-univerzální. (Hint: může se hodit základní věta algebry: kolik kořenů má polynom stupně k?)
b) $\mathcal{R}$ se říká "rolling hash", protože lze snadno spočíst $h_a(x_1 \dots x_d)$, pokud již máme $h_a(x_0x_1 \dots x_{d-1})$. Jak to přesně provést v konstantním čase?
c) Bonus: Jak z $\mathcal{R}$ udělat $(2, c)$-nezávislý systém?

3. *Rabin-Karpův algoritmus na vyhledávání v textu.* Využijte „rolling hash“ $\mathcal{R}$ ke vytvoření jednoduchého algoritmu na nalezení všech výskytů daného řetězce J délky m („jehly“) v textu S délky n („seně“) a to v průměrném čase $O(n + m + k \cdot m)$, kde k je počet výskytů J v S. Jak velké potřebujeme prvočíslo p pro $\mathcal{R}$?

A pokud zbyde čas, můžeme pro změnu hešovat perfektně. Příště už ale nebudeme hešovat a dokonce ani používat pravděpodobnost!