12. CVIČENÍ Z ADS 1

Viktor Němeček 13. 5. 2019

https://kam.mff.cuni.cz/~viki/vyuka/ads11819/

Příklady 1 a 2 byly vyřešeny již předminule, jsou zde pouze proto, aby třetí příklad dával smysl.

Příklad 1. Z černé krabičky vede na každou ze dvou stran n drátů. Víme, že uvnitř krabičky je jeden drát z jedné strany vodivě spojený s právě jedním drátem z druhé strany, dráty z téže strany k sobě nikdy připojené nejsou. Povolené operace jsou připojit k danému drátu napětí, odpojit od něj napětí a změřit na něm napětí. Kolik nejméně (asymptoticky) operací potřebujete, abyste zjistili, které dvojice drátů jsou spojené.

∗ **Příklad 2.** Nalezněte neadaptivní řešení na předchozí úlohu (tedy řešení, kde to, které operace provádíte, nezávisí na výsledcích měření.

∗ **Příklad 3.** Dokažte, že předchozí příklad nejde řešit rychleji, než v $\Omega(n \log n)$.

Hashování

Ve všech příkladech budeme předpokládat, že umíte nagenerovat libovolné množství dokonale náhodných funkcí.

Příklad 4. Řekli jsme, že bychom rádi zvolili velikost hashovací tabulky jako $\mathcal{O}(n)$, kde n je počet prvků, které chceme vkládat. Jak se vyrovnat s tím, když n předem neznáme (a přitom si chceme zachovat amortizovaně konstantní čas na operaci)?

Příklad 5. Mějme množinu čísel a číslo x. Zjistěte, zda množina obsahuje dvojici prvků se součtem x.

Příklad 6. Bloomův filtr je struktura na přibližnou reprezentaci množiny. Funguje jako standardní hashovací tabulka, až na to, že jednotlivá políčka mají jen jeden bit, a ukládá se do nich informace o tom, zda v množině je alespoň jeden prvek s danou hodnotou hashovací funkce. Všimneme si, že pokud prvek skutečně patří do naší množiny, struktura vždy odpoví správně, ale pokud ne, může udělat chybu a odpovědět, že ano. S jakou pravděpodobností se taková věc stane (v závislosti na počtu prvků v množině n a velikosti (počtu bitů) Bloomova filtru m)? Předpokládáme, že $m << |\mathcal{U}|$.

Příklad 7. Spolehlivost Bloomova filtru můžeme zvýšit tím, že si těchto filtrů pořídíme místo jednoho k, přičemž každému zvolíme jinou hashovací funkci. Insert pak provádíme do všech filtrů, search vrátí ano, pouze pokud všechny dílčí filtry odpověděly ano (delete tato struktura stejně jako základní bloomovy filtry neumožňuje). Představte si, že chcete ukládat množinu o 10^6 prvcích (z o několik řádů většího univerza) s pravděpodobností chyby nejvýše 10^{-9}. Pro jaké hodnoty k a m dosáhnete nejmenší celkové spotřeby paměti?

Příklad 8. Uvažujme hashování s otevřenou adresací. Použijeme funkci $h(x, i) = (f(x) + c \cdot i) \mod m$, kde c je konstanta nesoudělná s m. Jak se toto hashování chová ve srování s lineárním přidáváním?

Příklad 9. Jak byste pro hashování s otevřenou adresací naimplementovali delete, pokud chcete, aby po nalezení políčka, ze kterého se má mazat, už trval amortizovaně $\mathcal{O}(1)$?