GRAFOVÉ ALGORITMY - CVIČENÍ 1: REPREZENTACE GRAFŮ

Převzato z Cormen, Leiserson, Rivest, Stein: Introduction to algorithms. The MIT Press and McGraw-Hill, 2001; kapitola 22.

Příklad 1. Pro daný seznam sousedů orientovaného grafu navrhněte algoritmus pro výpočet výstupního (d_-) a vstupního (d_+) stupně všech uzlů tohoto grafu.

Příklad 2. Vytvořte příklad grafu binárního stromu se 7 uzly splňující omezení na binární haldu, tj. (a) všechny listy jsou v hloubce n nebo $n-1$ a (b) všichni potomci jsou vůči svému předkovi v relaci globálně dané pro celou haldu. Zapište tento graf pomocí seznamu sousedů a matice sousednosti. Pro ohodnocení uzlů využijte celá čísla od 1 do 7.

Příklad 3. Popište algoritmus, který převede orientovaný graf G na transponovaný orientovaný graf G^T, tj. všechny hrany G mají v G^T opačný směr. Uvažujte varianty algoritmu pro reprezentaci grafu G seznamem sousedů i maticí sousednosti. Analyzujte časovou složitost navržených algoritmů.

Příklad 4. Mějme orientovaný multigraf $G = (V,E)$ zadaný seznamem sousedů. Navrhněte algoritmus s časovou třídou složitostí $O(V+E)$, který transformuje G na „ekvivalentní“ neorientovaný graf $G' = (V,E')$, kde E' obsahuje hrany z E nejvýše jednou a jsou vynechány všechny smyčky (tzv. symetrizace grafu).

Příklad 5. Mějme orientovaný graf $G = (V,E)$ a jeho druhou mocninu definujme jako $G^2 = (V,E^2)$ tak, že $(u,w) \in E^2$ právě když existuje $v \in V$, že platí $(u,v) \in E$ a $(v,w) \in E$. Popište efektivní algoritmus pro výpočet G^2 z G pro obě reprezentace (seznam sousedů i matice sousednosti). Analyzujte doby běhu obou verzí algoritmu.

Příklad 6. Jaká je nejčastější časová složitost algoritmů pracujících s maticí sousednosti? Najděte algoritmus pracující v lineárním čase (patří do $O(V)$), který zjistí, zda orientovaný graf G zadaný maticí sousednosti obsahuje univerzální koncový uzel u (angl. *universal sink*), tj. $d_+(u) = |V| - 1$ a $d_-(u) = 0$.