4. CVIČENÍ Z ADS 1, ČTVRTEK 10:40

DFS: artikulace, DAGy, TU a silná souvislost

1. *Artikulace.* Vrchol v neorientovaného grafu G je artikulace, pokud $G - v$ má více komponent než G. Vymyslete, jak artikulace najít. Nápovědy:

- Na přednášce bylo následující lemma: v *není* artikulace $\Leftrightarrow$ pro každé dva jeho sousedy $x \neq y$ existuje kružnice, která obsahuje hrany vx a vy.
- Kdy je kořen DFS stromu artikulací?
- Kdy je obecný vrchol DFS stromu artikulací?

Dále se budeme zabývat hlavně orientovanými grafy.

2. *Jeznoznačnost TU.* Jakou vlastnost má graf, jehož topologické pořadí (uspořádání) je jednoznačné?

3. *Paralelní plánování.* Máme velký projekt, např. chceme postavit dům. Sestavíme si závislostní graf všech činností, které jsou potřeba. U každé činnosti si poznamenáme, jak dlouho bude (nejspíš) trvat. Máme k dispozici neomezeně mnoho pracovníků, takže můžeme vykonávat libovolně činností najednou, ale musíme dodržet závislosti (tedy než začneme nějakou činnost, musí být hotové všechny činnosti, na kterých závisí). Spočítejte pro každou činnost, kdy s ní začít, abychom projekt dokončili co nejdříve.

4. *Kritické vrcholy.* O činnosti z předchozího příkladu řekneme, že je kritická, pokud by její zpomalení o libovolné $\varepsilon > 0$ způsobilo pozdější dokončení projektu (o ε). Jak najít všechny kritické vrcholy? (Tomuto se někdy říká „metoda kritické cesty".)

5. *Nejdelší cesty.* Na vstupu máme acyklický orientovaný graf (DAG) s délkami hran $\ell : E \to \mathbb{R}^+$. Pro dané dva vrcholy u a v nalezněte počet nejdelší cest z u do v (a jejich délku, definovanou jako součet $\ell(e)$ přes hrany e na cestě).

6. *Silně souvislá orientace.* Ukažte, že v každém neorientovaném grafu *bez mostů* je možné hrany zorientovat tak, aby výsledný graf byl silně souvislý. Dává důkaz i efektivní algoritmus?

Bonusové úlohy:

7. *Zelený cyklus.* Mějme orientovaný graf, v němž jsou nějaké vrcholy obarveny zeleně. Jak zjistit, jestli v něm existuje cyklus, který obsahuje zelený vrchol.

8. *Longest Common Subsequence (LCS).* Pro dvě posloupnosti délek m a n nalezněte nejdelší společnou podposloupnost pomocí hledání nejdelší cesty ve vhodném DAGu. Jakou má algoritmus časovou složitost? (Tento problém lze řešit také pomocí dynamického programování.)

9. *Barvení rovinného grafu.* Mějme rovinný graf (lze ho tedy nakresit na rovinu bez křížení hran). Připomeňme, že takový graf obsahuje vrchol stupně max. pět. Obarvěte vrcholy grafu šesti barvami tak, aby žádné dva sousední vrcholy neměly stejnou barvu.