Aplikace pro analýzu souvisloti návrh sítí

Příklad 1: Testování: Zopakujte si algoritmus na hledání mostů v neorientovaném grafu pomocí DFS. Připomeňte si co je hodnota "low" a jak se počítá (a že pro orientované grafy nedává smysl kvůli příčným hranám). Nahlédněte při tom do ukázkové implementace BFS (na webu je opravená verze).

Pokud jste na přednášce neviděli algoritmus na hledání artikulací, zkuste ho vymyslet. Připomínka: artikulace je vrchol jehož odstraněním se graf stane nesouvislým (podobně jako most je hrana se stejnou vlastností), speciálně oba konce mostu jsou obvykle artikulace (vyjma listů).

Hinty: Podobně jako most je vždy stromová hrana, vrcholy s více incidentními stromovými hranami jsou podezřelé. Nicméně i vrchol s dvěma stromovými hranami může být artikulací. Pozor na kořen vyhledávání, chová se trochu jinak.

Oba algoritmy najdete v průvodci v kapitole 5.7. Dle sylabu musít znát algoritmus na hledání mostů, a algoritmus na artikulace byste měli být schopni (u zkoušky) alespoň zhruba odvodit s drobnými hinty jako hinty výše.

Příklad 2: 2-souvislost: Graf je hranově 2-souvislý pokud se odebráním žádné (jedné) hrany neporuší jeho souvislost. Komponenta 2-souvislosti je množina vrcholů t.ž. graf indukovaný touto množinou je 2-souvislý.

Rozdělte vrcholy neorientovaného grafu do komponent hranové 2-souvislosti. Zkonstruujte graf komponent hranové 2-souvislosti. Vrcholy představují komponenty původního grafu, a granami jsou spojené vrcholy mezi jejichž komponentami vedly hrany v původním grafu. (Např. pokud máme dvě kružnice spojené hranou, komponenty 2-souvislosti jsou právě kružnice, graf komponent tedy bude obsahovat dva vrcholy představující kružnice a hranu mezi nimi)

(*) Totéž pro vrcholovou 2-souvislost (komponenta vrcholové 2-souvislosti je technicky množina hran t.ž. se smazáním žádného vrcholu neznesouvislí)

Dle sylabu byste měli ovládat techniky pro vytváření grafů různých typů komponent.

Příklad 3: Zodolnění: (obtížnější úloha) Pro (souvislý) graf na vstupu přidejte co nejmenší množství hran tak, aby ve výsledném grafu nebyly žádné mosty. Nebudeme se snažit argumentovat složitost, ale měla by stačit kvadratická (tedy $\mathcal{O}((n+m)^2)$).

Analýza orientovaných grafů

Příklad 4: Topologie grafu: s využitím vhodného prohledávacího algoritmu, navrhněte/připomeňte si algoritmy pro následující

- sestrojení topologického pořadí vrcholů (nebo ukažte, že není DAG)
- test, zda u, v jsou ve stejné komponentě silné souvislosti
- najití zdrojové komponenty silné souvisloti
- (obtížnější) sestrojení grafu silně souvislých komponent v čase $\mathcal{O}(n+m)$

Dle sylabu byste měli být schopni popsat/navrhnout postupy pro všechny tyto procesy. Poslední bod byste měli zvládnout alespoň zhruba.

Příklad 5: Orientování: (Tato úloha je pouze k zamyšlení, nechceme řešit detaily provedení.)

Pro neorientovaný souvislý graf bez mostů navrhněte orientaci všech hran t.ž.

- výsledný graf je silně souvislý
- pro každý vrchol rozdíl vstupních a výstupních stupňů je nejvýše 1

DAGy, aplikace pro plánování a analýzu procesů

Příklad 6: Kritické činnosti: Mějme závislostní graf množiny činností projektu. Vrcholy jsou činnosti, ohodnocené délkou trvání, orientované hrany představují závislosti. Například pokud vykrádáme banku, před opuštěním banky bychom měli naplnit pytle penězi, a než vůbec začneme v vykrádáním, měli bychom si najít únikové vozidlo atd.

Zjistěte, jak dlouho bude minimálně trvat provést všechny činnosti (činnosti můžeme provádět paralelně, mají-li splněné předpoklady). Najděte všechny činnosti jejichž zpoždění by prodloužilo celkový čas.

Dle sylabu byste měli znát souvislosti DAGů a plánování, a být schopni vyřešit různé jednoduché úlohy s tím spojené.

Základy nejkratších cest

Příklad 7: Malé délky: Mějme graf s hranami ohodnocenými délkami $1, \ldots, k$ pro malou konstantu k. Každou hranu podrozdělíme na cestu takové délky, kolik je délka původní hrany. Nahlédněte, že spuštění BFS z některého z původních vrcholů odpovídá spuštění Dijkstrova algoritmu z téhož vrcholu v původním grafu. Dá se říct, že Dijkstra je vlastně jenom BFS udělané "pořádně".

Příklad 8: Délky vrcholů: Mějme graf s hranami i vrcholy ohodnocenými časem průchodu (délky ulic a očekávaná doba čákání na křižovatkách). Jak v takovém grafu hledat nejktratší cesty?

Toto je typický trik potřebný k řešení různých úloh, nejen pro hledání cest.