Aplikace pro analýzu souvisloti návrh sítí

Příklad 1: Testování: Oba algoritmy najdete v průvodci v kapitole 5.7.

Příklad 2: 2-souvislost: Pro hranovou 2-souvislost platí, že podgraf je komponenta právě když je (co do inkluze) maximální podgraf neobsahující most. Jinými slovy, když smažeme všechny mosty, komponenty zbytku grafu jsou právě komponenty hranové 2-souvislosti původního grafu.

Jak je tedy najít? Nejprve identifikujeme všechny mosty pomocí DFS. Následně můžeme všechny mosty z grafu smazat a prohledat graf znovu. Tentokrát nás bude zajímat rozklad na komponenty souvislosti.

Jak zkonstruovat graf komponent? Vytvoříme nový graf, t.ž. každý jeho vrchol odpovídá jedné komponentě 2-souvislosti. Budeme si udržovat odkazy na vrcholy v poli, indexovaném číslem komponenty. Jaké hrany budou v grafu komponent? Budou odpovídat přesně zmazaným mostům. Projdeme všechny most z původního grafu, a zeptáme se koncových vrcholů, do kterých komponent 2-souvislosti patří. Pomocí pole umíme příslušné vrcholy v grafu komponent najít a propojit je hranou.

Pro vrcholovou 2-souvislost postupujeme obdobně, ale s artikulacemi. Problémem je, především detekovat komponenty, protože se jedná o množiny hran (most je samostatná komponenta). Hrany v grafu komponent pak vzniknou projítím artikulací a spojením všech párů komponent, které artikulace spojují.

Příklad 3: Zodolnění: Sestavme si graf komponent hranové 2-souvislosti. Libovolná hrana, kterou přidáme do grafu spojí několik komponent do jedné, konkrétně všechny komponenty na (unikátní) cestě spojující příslušnou dvojici komponent v grafu komponent (všechny mosty reprezentované hranami na cestě přestávají být mosty, v grafu komponent se celá cesta kontrahuje). Naším cílem je zkolabovat celý strom komponent do jediného vrcholu. Snadno vypozorujeme, že jedna hrana nemůže snížit počet listů o více než dva (vnitřní vrcholy nejsou podstatné, pokud jsou všechny listy zkontrahovány do jednoho vrcholu, platí totéž pro vnitřní vrcholy automaticky). Může se ale stát, že přidáním hrany spojující dvě listové komponenty 2-souvislosti klesne počet listů pouze o 1, např. pokud tyto komponenty jsou reprezentovány dvěma listy, které jsou spojené cestou s jediným vrcholem stupně 3 a ostatními stupně 2. To je ale jediný problém. Stačí se tedy libovolnou stregií těmto situacím vyhnout.

Lze tedy dokázat, že na zodolnění je potřeba přesně $\lceil$počet listových komponent/2$\rceil$ hran a přesně tolik nám bude i stačit.

Analýza orientovaných grafů

Příklad 4: Topologie grafu: Řešení jsou v průvodci v kapitole 5.9.

Příklad 5: Orientování: Zhruba popsáno v hintech.

DAGy, aplikace pro plánování a analýzu procesů

Příklad 6: Kritické činnosti: Pro každý vrchol si budeme počítat čas kdy může činnost nejdříve začít. Pro každou činnost je tento čas maximum ze všech předchůdců plus jejich délka. Počítáním dle topologického pořadí máme všechny potřebné hodnoty vždy dostupné.

Pro najité kritických činností stačí jednoduchý test. Činnost A je kritická tehdy pokud některý z jejích následníků B má nejdřívější čas začátku roven nejdřívějšímu času začítku A plu trvání A a zároveň B je kritická. Z první podmínky platí, že jakékoliv zpoždění na A nutně ovlivní čas B. Druhá podmínka je nutná, protože nás zajímají pouze činnosti, které řetězově ovlivní konec celého projektu. Tyto podmínky lze snadno ověřit projitím činností ve zpětném topologickém pořadí.

Celý postup běží v čase $\mathcal{O}(n+m)$.

Základy nejkratších cest

Příklad 7: Malé délky: Úkolem bylo se zamyslet, řešení netřeba.

Příklad 8: Délky vrcholů: Každý vrchol v rozdělíme na dvojici vrcholu, v_{in} a v_{out}. Přidáme orientovanou hranu z *in* do *out* a její délka bude odpovídat délce vrcholu. Všechny původní hrany nahradíme dvojcemi protějších orientovaných hran (pokud graf již nebyl orientovaný), a všechny hrany napojíme tak, že orientovaná hrana vw je nahrazena za $v_{out}w_{in}$. Snadno nahlédneme, že cesty v původním grafu odpovídají cestám v novém grafu, ale místo každého vrcholu (kromě počátečního a koncového) bude na cestě dvojice *in* a *out* vrcholů a hrana mezi nimi.