3. cvičení z ADS1

5. 3. 2025

Definice. Vrchol $v \in V(g)$ je *artikulace*, jestliže $G - v$ má více komponent než G.

Příklad 1.

Víme, že DFS je rekurzivní, tudíž využívá zásobník na ukládání vrcholů. BFS naopak na ukládání vrcholů využívá frontu. Nabízelo by se tedy zkusit DFS naimplementovat tak, že vezmeme implementaci BFS, ve které nahradíme frontu zásobníkem.

Jak bude vypadat výsledný „DFS strom“ pro tento algoritmus? Bude stejný jako standardní DFS strom?

Příklad 2.

Naimplementujte DFS bez použití rekurze. Pokud lze využít implementaci z předchozího příkladu, odůvodněte proč.

Příklad 3.

Dokažte, že nachází-li se v souvislém grafu na alespoň třech vrcholech most, pak v něm taky najdeme artikulaci. Ukažte, že opačná implikace neplatí.

Příklad 4.

Ve škole se nachází počítačová síť tvořící graf. Vrcholy jsou přirozeně počítače a hrany značí dvojice počítačů, které mohou komunikovat přímo.

Dozvěděli jsme se, že zítra přijdou údržbáři a se sítí provedou nějaké údržbové práce.

K tomu ale potřebují, aby všechny počítače byly vypnuté. V libovolném čase však chceme, aby byly libovolné dva zapnuté počítače mohly navzájem komunikovat. Musíme proto počítače vypínat v takovém pořadí, abychom možnost komunikace zbylých počítačů nikdy nepřerušili. Navrhněte algoritmus, který toto pořadí najde.

Příklad 5.

Jaké vlastnosti musí pro $v \in V(G)$ platit, aby v byla artikulace? Zkuste využít klasifikaci hran. Pomocí této vlastnosti pak navrhněte algoritmus, který artikulace najde.

Příklad 6.

Vraťme se ještě k RAMu s konstantně rychlými neomezeně velkými buňkami. Ukažte, jak reprezentovat vektory, abychom uměli přečíst a zapsat konkrétní složku, sečíst dva vektory nebo násobit skalárem v konstantním čase. Pro zjednodušení předpokládejte, že složka vektoru nikdy nepřesáhne konstantu 2^m.