# 3. cvičení z ADS1

18. 3. 2021

**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.

Jaké vlastnosti bude mít tento algoritmus?

**Příklad 2.**

Zkuste DFS naimplementovat bez použití rekurze tak, aby uměl i klasifikovat hrany. Pokud lze využít implementaci z předchozího příkladu, odůvodněte proč.

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

**Příklad 3.**

Jaké vlastnosti musí pro $v \in V(G)$ platit, aby $v$ byla artikulace? Zkuste využít klasifikaci hran. Pomocí této vlastnosti pak upravte DFS tak, aby umělo hledat artikulace $G$.

**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.**

Už víme, že můžeme najít topologické uspořádání DAGu tak, že postupně budeme odtrhávat zdroje. Naimplementujte algoritmus používající tuto techniku tak, aby běžel v čase $\mathcal{O}(n + m)$.

**Příklad 6.**

Uvažme DAG $G$ s hranami, které jsou ohodnocené nezápornými čísly. Najděte algoritmus běžící v čase $\mathcal{O}(n + m)$, který najde nejkratší cestu mezi dvěma vrcholy $u, v$.

Půjde tento algoritmus upravit, aby naopak hledal cestu nejdelší?