GRAFOVÉ ALGORITMY - CVIČENÍ 2: PROHLEDÁVÁNÍ DO ŠÍŘKY (BFS)

Převzato z Cormen, Leiserson, Rivest, Stein: Introduction to algorithms. The MIT Press and McGraw-Hill, 2001; kapitola 22.

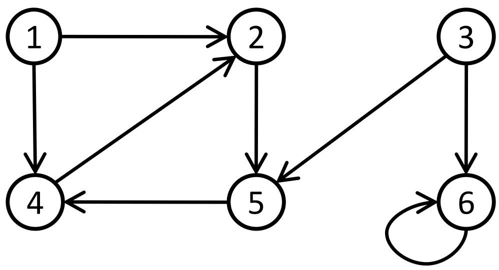
Příklad 1. Demonstrujte běh algoritmu BFS na zadaném

a) orientovaném grafu (viz obrázek 1); začněte z uzlu 3.

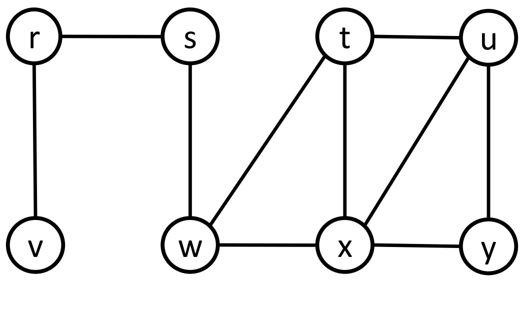
b) neorientovaném grafu (viz obrázek 2); začněte z uzlu u.

Uveďte i historii hodnot polí d a π.

Příklad 2. Určete a porovnejte časovou složitost BFS (a dalších základních operací nad danou reprezentací jako inicializace, hledání zadané hrany, vložení nové hrany nebo určení nulovosti stupně daného uzlu) pro vstupní graf reprezentovaný jako (a) seznam sousedů a (b) maticí sousednosti. Zapište také modifikovaný BFS pro reprezentaci grafu pomocí matice sousednosti.



Obrázek 1: Příklad 1a



Obrázek 2: Příklad 1b

Příklad 3. Závisí u algoritmu BFS hodnoty $d[u]$ na pořadí, ve kterém jsou uzly uloženy v seznamu sousedů? Na příkladu (viz obrázek 2) demonstrujte, že výsledný strom konkrétního prohledávání do šířky může na pořadí uložení uzlů v seznamu sousedů záviset.

Příklad 4. Vymyslete příklad orientovaného grafu $G = (V, E)$ s počátečním uzlem $s \in V$ a množinou stromových hran $E' \subseteq E$ takových, že pro každy uzel $v \in V$ je cesta v grafu $G' = (V, E_\pi)$ z s do v jediná a zároveň nejkratší cestou v G, a přesto G' není možné za žádných okolností vygenerovat jako strom prohledávání do šířky (i když zvolíme jakékoli pořadí uzlů v seznamu sousedů).

Příklad 5. Mějme n zápasníků. Mezi zápasníky jsou rivalové; zapsáno jako r dvojic vyjadřujících vzájemnou rivalitu. Navrhněte algoritmus s časovou složitostí $O(n + r)$, který určí, zda je možné rozdělit zápasníky do dvou skupin dle pravidla „rival mého rivala není můj rival" (tj. rivalita mezi zápasníky v rámci téže skupiny je zakázána). Je-li takové rozdělení možné, bude výstupem tohoto algoritmu.

Příklad 6. *Průměr stromu* $T = (V, E)$ je dán jako $\max_{u,v \in V} \delta(u, v)$; jinými slovy, průměr je nejdelší vzdálenost ze všech nejkratších cest v daném stromě. Navrhněte efektivní algoritmus, který bude zjišťovat průměr stromu, a analyzujte jeho časovou složitost.

Příklad 7. Mějme souvislý neorientovaný graf $G = (V, E)$. Navrhněte algoritmus s časovou složitostí $O(V + E)$, který vypočítá průchod grafem G procházející všemi hranami z E právě jednou v každném směru. Jako doplnění úlohy popište, jak byste hledali cestu ven z bludiště, pokud byste disponovali velkým počtem drobečků.