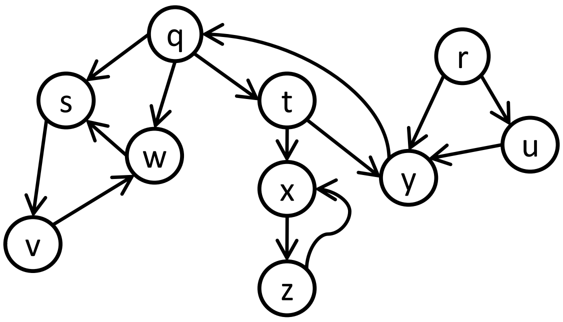
GRAFOVÉ ALGORITMY - CVIČENÍ 3: PROHLEDÁVÁNÍ DO HLOUBKY (DFS)

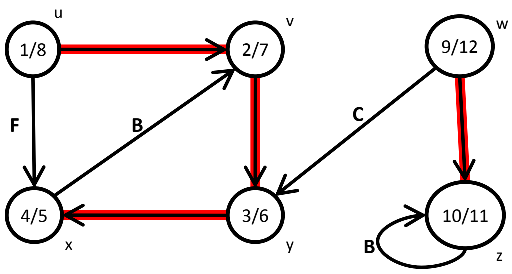
Převzato z Cormen, Leiserson, Rivest, Stein: Introduction to algorithms. The MIT Press and McGraw-Hill, 2001; kapitola 22.

Příklad 1. Vytvořte tabulku 3x3 s řádky a sloupci označenými barvami *bílá*, *šedá*, *černá*. V každé buňce na souřadnicích (i, j) označte, zda v nějakém okamžiku při prohledávání do hloubky v orientovaném grafu může existovat hrana z uzlu barvy i do uzlu barvy j. Pro každou možnou hranu označte, kterého typu (T pro stromovou, F pro dopřednou, B pro zpětnou, C pro křížící) může být. Vytvořte druhou takovou tabulku pro prohledávání do hloubky v neorientovaném grafu.

Příklad 2. Demonstrujte práci algoritmu prohledávání do hloubky na grafu na obrázku 1. Předpokládejte, že cykly procedury DFS vybírají uzly v alfabetickém vzestupném pořadí a že seznamy sousedů jsou také řazeny (a procházeny) vzestupně alfabeticky. Ukažte časy d (angl. *discovery*) a f (angl. *finishing*) a typ (T, F, B, C) pro každý uzel.



Obrázek 1: Příklad 2



Obrázek 2: Příklad 3

Příklad 3. Ukažte závorkový výraz (tak, aby levá závorka specifikovala otevření uzlu (přiřazení hodnoty d), byla následována tímto uzlem; při uzavírání uzlu (přiřazení hodnoty f) by byl opět vypsán tento uzel a doplněna pravá závorka) aplikovaného prohledávání do hloubky zachyceného na obrázku 2. Jaký má vztah struktura závorek k typům jednotlivých hran.

Příklad 4. Dokažte, že hrana (u,v) v libovolném grafu je:

a) stromová nebo dopředná hrana, právě když $d[u] < d[v] < f[v] < f[u]$.

b) zpětná hrana, právě když $d[v] \leq d[u] < f[u] \leq f[v]$.

c) křížící hrana, právě když $d[v] < f[v] < d[u] < f[u]$.

Příklad 5. Ukažte, že v neorientovaném grafu je klasifikace hrany (u,v) jako stromové nebo zpětné, podle toho kterou reprezentaci hrany (tj. (u,v) nebo (v,u)) prozkoumá prohledávání do hloubky jako první, ekvivalentní klasifikaci této hrany podle priority typů v klasifikačním schématu (tj. stromová, zpětná, dopředná a nakonec křížící hrana).

Příklad 6. Modifikujte algoritmus DFS, aby používal zásobník místo rekurze. Detailně analyzujte jeho složitost, zda nedošlo ke změně oproti klasické rekurzivní verzi algoritmu.

Příklad 7. Vymyslete protipříklad následujících hypotéz:

a) Existuje-li cesta z u do v v orientovaném grafu G taková, že při prohledávání do hloubky je $d[u] < d[v]$, pak v je následníkem u v lese prohledávání do hloubky.

b) Existuje-li cesta z u do v v orientovaném grafu G, pak jakékoli prohledávání do hloubky musí skončit s nerovností $d[v] \leq f[u]$.

Příklad 8. Modifikujte algoritmus DFS tak, aby vytiskl během průchodu každou zkoumanou hranu a její typ. Vypracujte i alternativu pro neorientovaný graf.

Příklad 9. Vysvětlete, kdy může dojít k situaci, že DFS strom procházení grafu G obsahuje izolovaný uzel u, přestože u má vstupní i výstupní hrany v G.

Příklad 10. Ukažte, že prohledávání do hloubky v neorientovaném grafu G může být použito pro nalezení souvislých komponent v G a že DFS les obsahuje tolik stromů, jako má G souvislých komponent. Přesněji řečeno, ukažte změnu algoritmu DFS takovou, že každý uzel dostane celočíselné ohodnocení od 1 do k např. v poli cc, kde k je počet souvislých komponent grafu G a navíc $cc[u] = cc[v]$, právě když jsou u a v ve stejné souvislé komponentě.

Příklad 11. Mějme graf $G = (V,E)$, který nazveme *jednoduše souvislý*, pokud $u \rightsquigarrow v$ implikuje existenci nejvýše jedné cesty z uzlu u do v pro všechny uzly $u,v \in V$. Navrhněte efektivní algoritmus na určení, zda je nebo není daný orientovaný graf jednoduše souvislý.