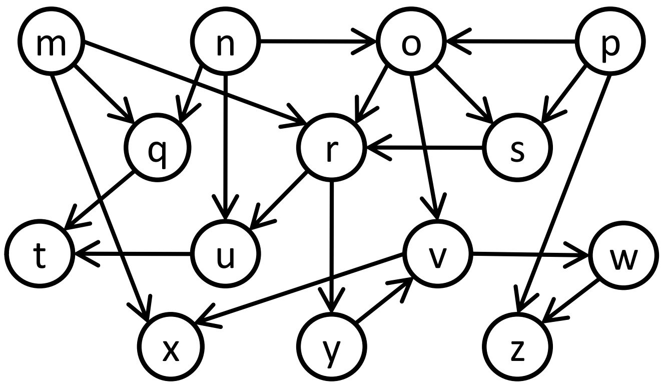
GRAFOVÉ ALGORITMY - CVIČENÍ 4: TOPOLOGICKÉ USPOŘÁDÁNÍ

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

Příklad 1. Ukažte uspořádání uzlů vypočtené algoritmem TOPOLOGICAL-SORT(G), když je spuštěn nad orientovaným acyklickým grafem (zkráceně DAG) G na obrázku 1. Předpokládejte, že uzly jsou vybírány ve vzestupném alfabetickém pořadí a také seznamy sousedů jsou seřazeny vzestupně alfabeticky.



Obrázek 1: Příklad 1

Příklad 2. Navrhněte algoritmus s lineární časovou složitostí, který vezme na vstupu DAG $G = (V, E)$ a dva uzly s a t, a vrátí počet různých cest z uzlu s do uzlu t v G. Například pro DAG na obrázku 1 a uzly p a v existují z p do v čtyři různé cesty: *pov*, *poryv*, *posryv* a *psryv*. Algoritmus však v tomto případě vrátí pouze číslo 4 (nikoli výpis těchto cest).

Příklad 3. Vytvořte algoritmus, který určí, zda daný neorientovaný graf $G = (V, E)$ obsahuje kružnici nebo ne. Tento algoritmus by měl mít časovou složitost $O(V)$, tj. nezávislou na $|E|$.

Příklad 4. Dokažte nebo vyvraťte následující tvrzení: Obsahuje-li orientovaný graf G cykly, tak procedura TOPOLOGICAL-SORT(G) vrací uspořádání uzlů takové, že počet "špatných" hran (porušujících topologičnost generovaného uspořádání) je minimální.

Příklad 5. Jiný způsob získání topologického uspořádání orientovaného acyklického grafu $G = (V, E)$ využívá opakovaného hledání uzlu s nulovým vstupním stupněm, a pak takovýto uzel vypíše, odstraní jej spolu se všemi jeho hranami z G (hrany jsou pouze vycházející z tohoto uzlu, neboť nemá již žádnou vstupní). Vysvětlete, jak je možné tento přístup implementovat s časovou složitostí $O(V + E)$. Co se stane s tímto algoritmem, pokud bude G obsahovat cykly?