9. Průchod grafem *(zapsali T. Hubík, V. Kolomičenko, L. Slinták)*

1. do hloubky (DFS)

in(v), *out(v)*

Hrany rozdělíme do nekolika tříd

- *stromové* postupují do neoznačených vrcholů
- *zpětné* vedou do už označených vrcholů
- *dopředné* vedou do vrcholů, odkud již jsme se vrátili
- *příčné* pouze zprava doleva

Pozn.: u neorientovaných grafů mohou být pouze stromové a zpětné hrany

Složitost $\mathcal{O}(m+n)$, n … počet vrcholů, m … počet hran

2. do šířky (BFS)

Složitost $\mathcal{O}(m+n)$

Definice: *Lineární uspořádání* $<$ vrcholů grafu G je topologické $\iff \forall(u,v) \in E(G) : u < v$

Pozorování: Pro cyklické grafy topologické uspořádání neexistuje.

Věta: Libovolný acyklický orientovaný graf (DAG) má topologické uspořádání a lze ho sestrojit v $\mathcal{O}(m+n)$.

Důkaz: Spustíme DFS, $u < v$ právě tehdy, když *out(u)* $>$ *out(v)*

hrany:

- stromová - *out(u)* $>$ *out(v)*
- zpětná - existuje cyklus a to je spor
- dopředná - *out(u)* $>$ *out(v)*
- příčná - *out(u)* $>$ *in(u)* $>$ *out(v)*

Důsledek: Orientovaný graf G je cyklický $\iff$ DFS najde zpětnou hranu, existuje zpětná hrana $\Rightarrow$ existuje cyklus, neexistuje zpětná hrana $\Rightarrow$ existuje topologické uspořádání $\Rightarrow$ neexistuje cyklus. ♡

Nejdelší cesta v DAGU:

(z S do každého vrcholu)

1. Zvolíme topologické uspořádání na G, BÚNO S je minimum.
2. $\forall v : D(v) = -inf; D(S) = 0$
3. Postupně procházíme vrcholy $v \in V(G)$ v topologickém pořadí a pro $\forall v$ spočítáme $D(v)$
4. $D(v) = max(D(n)) + 1$

$u : (u,v) \in E(G)$

Pozorování: Každý DAG obsahuje *zdroj* (vycházejí z něj cesty) a *stok* (cesty do něj vstupují)

Rozkládání orientovaných grafů na komponenty souvislosti

Definice: R bude relace na $V(G)$ tak, že $uRv \iff$ existuje orientovaná cesta v G z u do v a opačne

Definice: G je silně souvislý $\iff \forall(u,v) \in V(G) : uRv$

Pozorování: R je ekvivalence, $u...v$ $v...w$ je u-w sled $\Rightarrow$ existuje u-v cesta

Definice: Komponenty silné souvislosti grafu $G \iff$ ekvivalence třídy relace R

Lemma: $\forall(G)$ a $\forall(u,v) \in V(G)$: existuje u-v sled $\iff$ existuje u-v cesta

Důkaz: Buď S u-v sled, na kterém se opakuje vrchol z. Uvažme sled $u...z$ $z...v$, což je také u-v sled, ale je kratší. Iterujeme $\Rightarrow$ zbude cesta. ♡

Definice: Graf komponent $C(G)$

$V(C(G))$: SSK grafu G

$(c_1, c_2) \in E(C(G)) \iff \exists v_1 \in c_1, v_2 \in c_2 : (v_1, v_2) \in E(G)$

Lemma: $\forall(G) : C(G)$ je DAG.

Důkaz: obrázkem a sporem, existuje cyklus $c_1, c_2, ..., c_k$ v $C(G)$. ♡

Algoritmus hladání silně souvislých komponent grafu v O(n)

Pozorování 1: DFS z vrcholu v projde $\{w, \exists v\text{-}w \text{ cesta}\}$.

Pozorování 2: Je-li C stoková komponenta a $v \in V(C)$, pak DFS z v projde právě C.

Pozorování 3: Spustíme-li DFS na celý G, pak vchol v s maximální *out(v)* leží ve zdrojové SSK.

Pozorování 3:* Nechť C_1, C_2 jsou SSK a $C_1, C_2 \in E(C(G)$, pak $max\{out(u), u \in V(C_1)\} > max\{out(v), v \in V(C_2)\}$.

Pozorování 4: G a G^R mají stejné SSK (G^R má proti G otočené hrany).

Algoritmus:

1. Sestrojíme G^R $\mathcal{O}(m+n)$
2. DFS na $G^R \rightarrow$ *out(v)* $\mathcal{O}(m+n)$
3. T bude uspořádání vrcholů podle klesajících *out(v)* $\mathcal{O}(n)$
4. $\forall v \in V(G)$ v pořadí podle T $\mathcal{O}(m+n)$
5. pokud v ještě nepatří do žádné komponenty, tak
6. spustím DFS z v a dosažené vrcholy prohlásím za další SSK $\mathcal{O}(m+n)$

Věta: Každý orientovaný graf lze rozložit na SSK v čase $\mathcal{O}(m+n)$

Důkaz: Pozorovani 3*:

- 1. DFS vstoupí nedříve do C_1 : Z C_2 vyleze určitě dřív než z C_1 $\Rightarrow$ $out(C_1) > out(C_2)$.
- 2. nejdříve do C_2: Nejdřív se vrátím z celé C_2, až pak někdy zase vlezu do $C_1 \Rightarrow out(C_1) > out(C_2)$.