GRAFOVÉ ALGORITMY

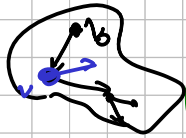
- BFS (prohledávání grafu do šířky)
- DFS (... do hloubky)

DFS2(v):

1. Stav(v) ← Otevřený
   T ← T+1, in(v) ← T
2. Pro hrany vw:

O(1) na hranu

3. Pokud Stav(w) = nenalezen:
4. DFS2(w)
5. Stav(v) ← zavřený, T ← T+1, out(v) ← T



Stav(v): nenalezen, otevřený, zavřený

in(v), out(v)

DFS(v_0): $\forall v$ Stav(v) ← Nenalezen
in(v), out(v) ← ∅
T ← 0
DFS2(v_0)

O(n)

Notace: $G=(V,E)$ → Chceme čas $\Theta(n+m)$

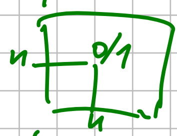
$n := |V|$

$m := |E|$

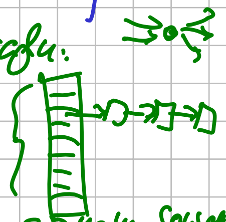
typ. orientovaný v_0

multigraf ?

Reprezentace grafu:



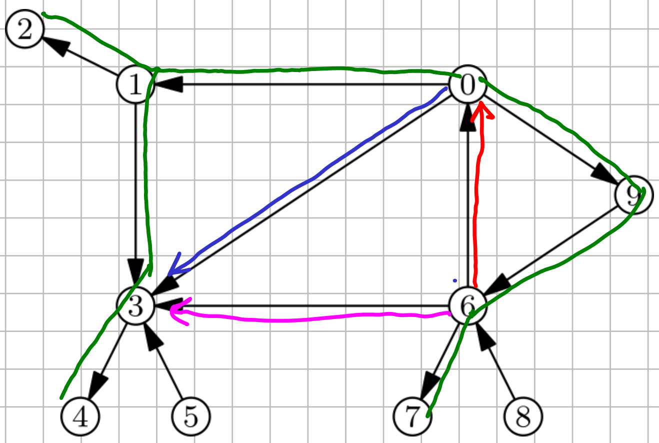
matice sousednosti



seznamy sousedů

Lemma: DFS se zastaví v čase $O(n+m)$.

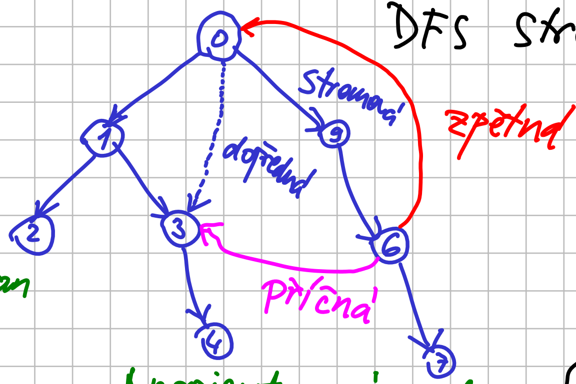
Lemma: Po zastavení DFS je $\forall v$ stav(v) = $\begin{cases} \text{nenalezený} \\ \text{zavřený} \iff v \text{ je dosažitelný z } v_0 \end{cases}$



$(_0\ (_1\ (_2\ )_2\ (_3\ (_4\ )_4\ )_3\ )_1\ (_9\ (_6\ (_7\ )_7\ )_6\ )_9\ )_0$

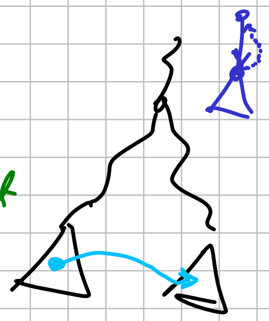
je dobře uzávorkované

DFS strom



Typ hrany $xy \in E$: ← DFS klasifikace hran

$(_x \ldots (_y \ldots)_y \ldots)_x$	stromová	y neobjevený
	dopředná	y uzavřený
$(_y \ldots (_x \ldots)_x \ldots)_y$	zpětná	
$(_y \ldots)_y \ldots (_x \ldots)_x$	příčná	
$(_x \ldots)_x \ldots (_y \ldots)_y$	nenastane!	



Neorientovaný graf

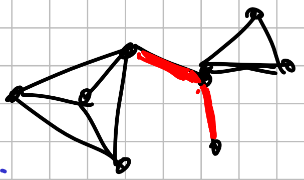
poprvé	podruhé
stromová	zpětná
zpětná	dopředná

Typ hrany zjistíme v čase $O(1)$!

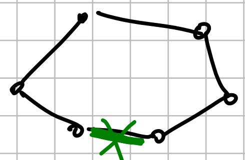
Věta: DFS v čase $\Theta(n+m)$ a prostoru $\Theta(n+m)$ najde dosažitelné vrcholy a klasifikuje dosažitelné hrany.

Mosty v neorientovaných grafech

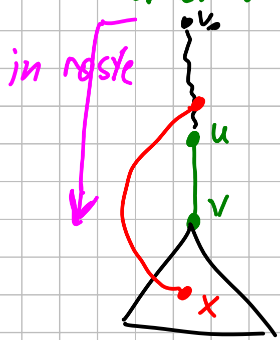
Df: Hrana $e \in E(G)$ je **most** $\equiv$ $G-e$ má více komponent než G.



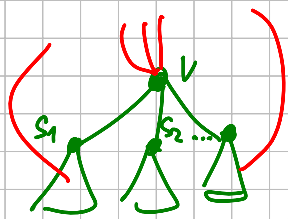
Lemma: Hrana e není most $\iff$ e leží na nějaké kružnici.



👁 zpětná leží na kružnici $\to$ není most
stačí zkoumat stromové:



uv leží na kružnici
$\Updownarrow$
$\exists\, xy \in E$ zpětná t.ž. x je potomkem v, y je předkem v, $in(y) < in(v)$ $\iff$ $low(v) < in(v)$



Df: $low(v) := \min\{ in(y) \mid xy$ je zpětná & x pod $v \}$
může být $+\infty$; neostře

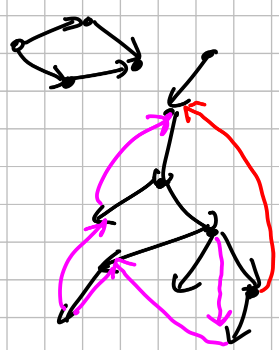
$low(v)$ počítám při odchodu z v:

$$low(v) = \min \begin{cases} low(s) \mid s \text{ syn } v \\ in(y) \mid vy \text{ zpětná hrana} \end{cases}$$

trvá $O(\deg(v))$

Věta: Alg. nalezne všechny mosty v čase a prostoru $\Theta(n+m)$.

Acyklické Orientované Grafy (DAGy)



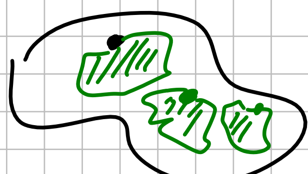
(1) Je graf DAG?

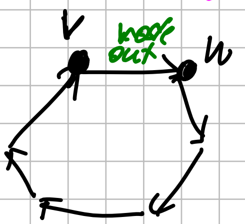
Lemma: $\exists$ dosažitelný cyklus
$\Updownarrow$
DFS najde zpětnou hranu.

$\Uparrow$ triv. $\Downarrow$ nechť $\exists$ cyklus C
zvolme $v \in C$ s min. $out(v)$
$w :=$ následník v na cyklu
na hraně vw roste out
$\downarrow$
zpětná hrana

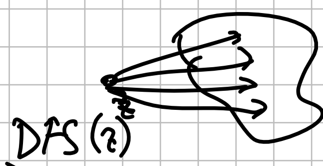
Opakujeme DFS z neobjevených vrcholů

Pro všechny $v \in V$:
Pokud $stav(v) =$ neobjeven:
DFS2(v)





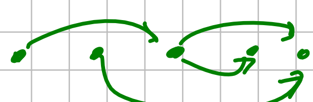
nebo: přidám nový zdroj z



(2) Topologické uspořádání

↳ Df: Lineární uspořádání $\preccurlyeq$ na $V(G)$ t.ž. $\forall\, xy \in E(G)$: $x \prec y$.

↳ alt.: očíslování vrcholů v_1 — v_n t.ž. kdykoliv $v_i v_j \in E$, pak $i<j$.



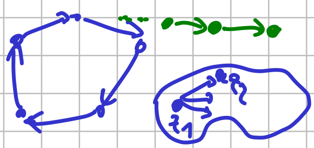
👁 cyklus nemá TU

Věta: G má TU $\iff$ G je DAG.

Dk: Df: $v \in V$ je zdroj $\equiv \deg^{in}(v) = 0$ [stok $\equiv \deg^{out}(v) = 0$]

Lemma: Každý DAG má zdroj.

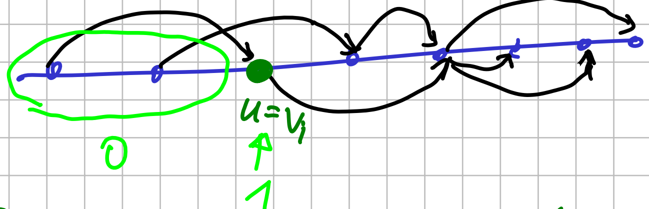
stačí odtrhávat zdroje... $\to$ Cvičení: impl. v $O(n+m)$.



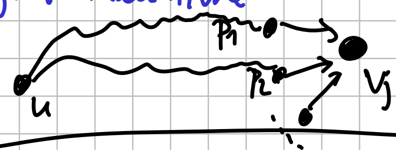
Věta: Pořadí, v němž DFS opouští vrcholy, je opačné topologické.

3. Topologická indukce

Příklad: zvolíme $u \in V$, chceme $\forall v \in V$: $c(v) :=$ # cest z u do v.
↳ pro DAG



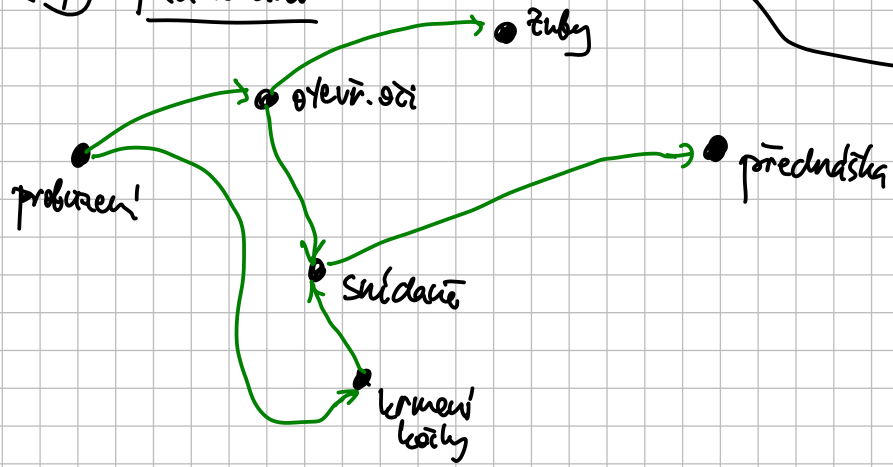
Nechť $v_1 - v_n$ je top. pořadí.
i: $u = v_i$
- Pro $j < i$: $v_j = 0$
- $v_i = 1$
- $j > i$ induktivně

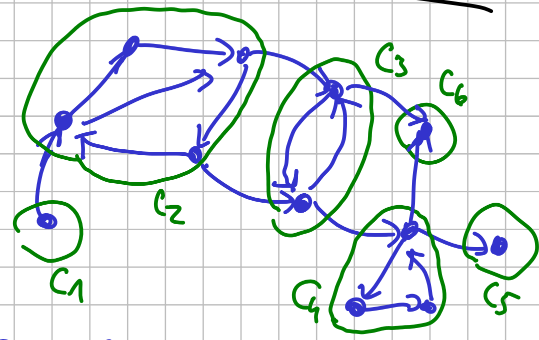


$$c(v_j) = \sum_{p:\ p v_j \in E} c(p)$$

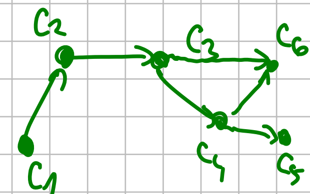
čas: v 1 vrcholu $\Theta(\deg(v))$
⇓
celkem $\Theta(n+m)$.
včetně nalezení TU.

4. Plánovdní





Df: Graf komponent $\mathcal{C}(G)$
vrcholy: komponenty grafu G
hrany: (C_i, C_j): $\exists x \in C_i$, $\exists y \in C_j$, $xy \in E(G)$
neboli kondenzace



5. Silná souvislost

Df: Relace $\rightsquigarrow$ na V: $u \rightsquigarrow v \equiv \exists$ sled z u do v.
$\leftrightsquigarrow$: $u \leftrightsquigarrow v \equiv u \rightsquigarrow v$ & $v \rightsquigarrow u$

$\leftrightsquigarrow$ je ekvivalence
↳ třídy jsou komponenty silné souvislosti
↳ G je silně souvislý ⇔ # komp. s.s. $= 1$

Lemma: $\mathcal{C}(G)$ je DAG.

Dk: kdyby $\exists$ cyklus $C_1, C_2, \dots, C_k, C_1$

$\exists\ u_1 - u_k$: $\forall i$ $u_i, v_i \in C_i$
$v_1 - v_k$ $\quad u_i v_{i+1} \in E$ (cyklicky)

Nyní v_1 (cesta v C_1) $u_1 v_2$ (cesta v C_2) $u_2 v_3 \dots$
$\dots v_k$ (cesta v C_k) $u_k v_1$

je cyklus v G ⇒ spor s růzností $C_1 - C_k$. ↯