Silná souvislost

Graf komponent $\mathcal{C}(G)$

- vrcholy jsou komponenty

① je to DAG

② $\exists$ aspoň 1 zdrojová a aspoň 1 stoková } komp.

③ Je-li C stoková a $v \in V(C)$, pak DFS(v) navštíví právě C.

④ Pokud spustíme opakované DFS, pak vrchol s max. outem leží ve zdrojové komponentě.

⑤ $G^T := (V(G), \{vu \mid uv \in E(G)\})$

G je DAG $\Leftrightarrow$ G^T je DAG

G a G^T mají stejné ekviv. třídy $\Leftrightarrow$

zdroj $\xleftrightarrow{G^T}$ stok

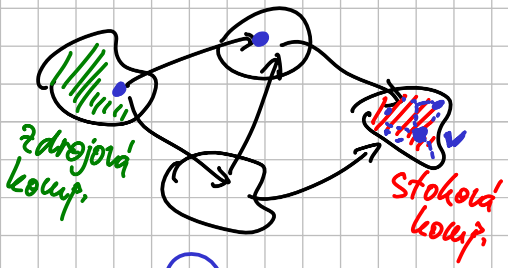
$\mathcal{C}(G^T) \cong \mathcal{C}(G)^T$

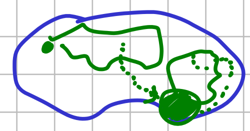
zdrojové komp. $\Leftrightarrow$ stokové komp.

Algoritmus:

1. Sestrojíme G^T. $O(n+m)$
2. $Z \leftarrow$ prázdný zásobník.
3. Opakované DFS v G^T, při opuštění vrcholu přidáme do Z. } $O(n+m)$
4. $\forall v$ komp(v) $\leftarrow \emptyset$] $O(n)$
5. Postupně odebíráme vrcholy ze Z, pro vrchol v: $O(n)$
6. Pokud komp(v) $= \emptyset$:
7. Spustíme DFS v G z vrcholu v, chodíme jen do vrcholů s komp $= \emptyset$ a nastavujeme komp $\leftarrow v$.

$O(n+m)$





⑥ Najdeme v ve zdrojové komp. v G^T

↳ v je ve stokové komp. v G

⑦ Odebereme stokovou komp. a opakujeme...

→ korektní, ale pomalé...

⑧ Procházíme vrcholy v pořadí klesajících outů v G^T, pokud ještě nemají přiřazenou komponentu, spouštím z nich DFS v G a označuji komponenty.

↓ funguje díky

Lemma: Pokud C_1, C_2 jsou komponenty t.ž. $C_1C_2 \in E(\mathcal{C}(H))$, pak:

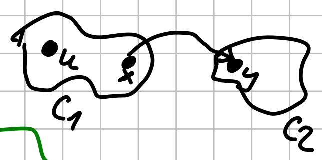
$$\max_{u \in C_1} out(u) > \max_{v \in C_2} out(v).$$

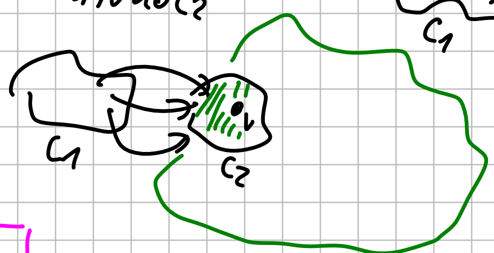
v DFS na H

použijeme na $H = G^T$

Dk: Případy → DFS vstoupí do C_1 před C_2

↓ dřív do C_2





Věta: Algoritmus najde komponenty silné souvislosti v čase a prostoru $\Theta(n+m)$.

Nejkratší cesty

orientovaný graf $G=(V,E)$ s délkami hran $\ell: E \to \mathbb{R}^+_0$.

→ délka uv-cesty P: $\ell(P) := \sum_{e \in P} \ell(e)$.

nebo sledu

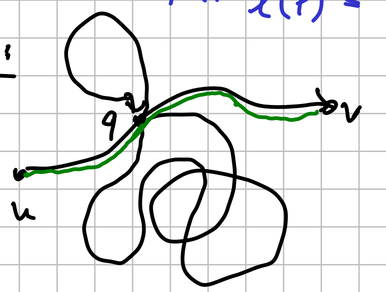
cesta z u do v

vzdálenost $d(u,v) := \min\{\ell(P) \mid P \text{ je } uv\text{-cesta}\}$

může být $+\infty$, pokud $\nexists$ cesta

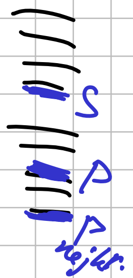
Lemma: Pokud S je uv-sled, pak $\exists P$ uv-cesta t. ž. $\ell(P) \le \ell(S)$.

Dk:



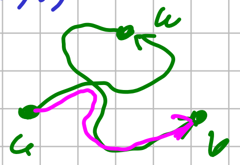
též platí

$$d(u,v) = \min\{\ell(S) \mid S \text{ je } uv\text{-sled}\}$$



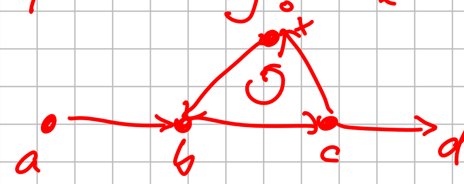
$\triangle$ nerovnost: $\forall u,v,w$

$$d(u,v) \le d(u,w) + d(w,v)$$



záporné hrany!

$\ell = -1$



opatrný kompromis: zakážeme záporné cykly

$d(a,d) = -3$

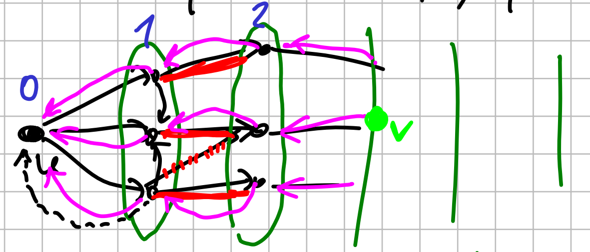
$d(a,x) = -3$

$d(x,d) = -3$

$-3 \not\le \underbrace{(-3) + (-3)}_{-6}$!

nejkratší sled neexistuje, nejkr. cesta je těžká

$\boxed{\ell = 1}$ Chceme-li spočítat $d(u,v)$, spustíme BFS(u).

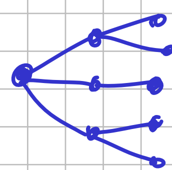


vrstvy ≈ vzdálenosti od u

→ vrcholům přiřadíme čísla vrstev ↪ $d(u,-)$

→ předchůdci $P(v)$ ↪ rekonstrukce cesty

v čase $\Theta(n+m)$

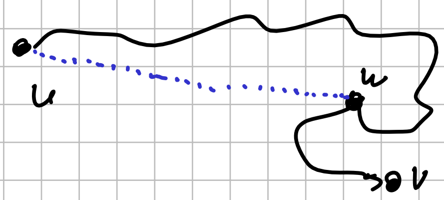


Df: Strom nejkratších cest

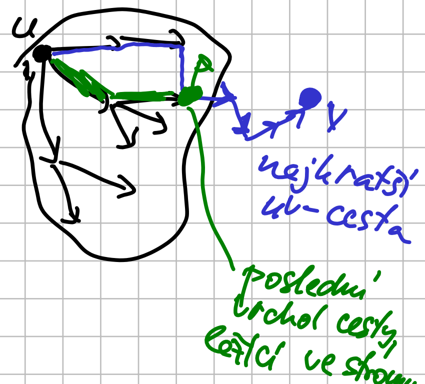
- strom na V (resp. dosaž. části)
- podgraf G
- orientovaný od kořene, kterým je u
- $\forall v \in V$ cesta ve stromu z u do v je jednou z nejkratších uv-cest v G

kompaktní reprezentace nejkr. cest z u kamkoliv

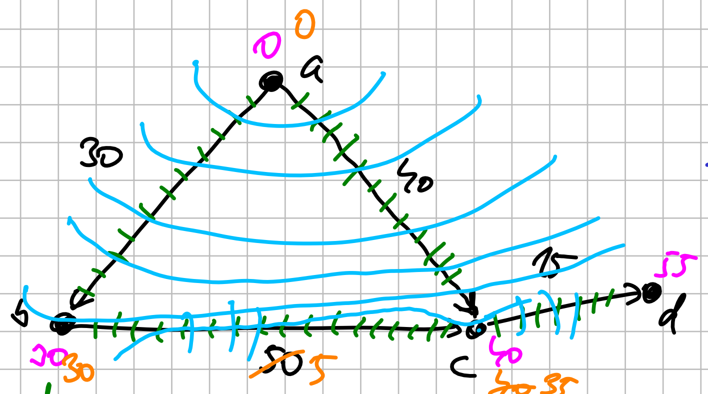
Prefix nejkr. cesty je zase nejkr. cesta



Lemma: Strom nejkr. cest existuje i v ohodnocených grafech.



$\ell(e)$ jsou přirozená čísla



hrany podrozdělíme na jednotkové a pak BFS

↳ nový graf $O(m \cdot L)$ hran!

max. délka

„budíky" pro vrcholy

- kdy se vlna poprvé dostane do vrcholu
- spíme, než nás probudí vnější budík

↳ podle potřeby (pře)nastavím další budíky

Dijkstrův alg.