orientovaný graf $G = (V, E)$

$$E \subseteq V \times V$$

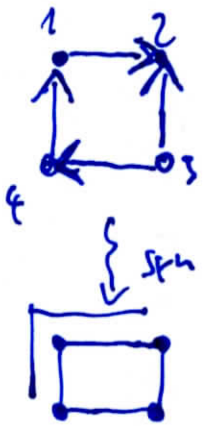
orientovaný sled, tah, cesta, kružnice

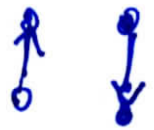
symetrizace grafu G je neorientovaný graf $G' = (V, E')$

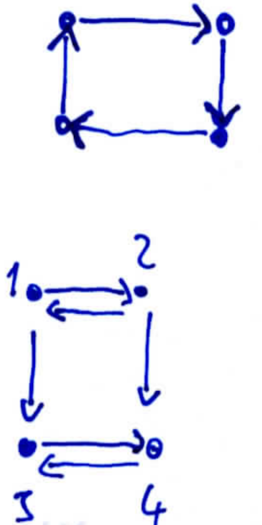
$$\underbrace{E' = \{ \{u, v\} \mid (u, v) \in E \}} \qquad (v, u) \in E$$

- slabá souvislost v G $\Leftrightarrow$ symetrizace je souvislá
- silná souvislost v G $\Leftrightarrow$ $\forall u, v \in V$ $\exists$ cesta (u, v) $\rightsquigarrow$ $\exists$ cesta (v, u)

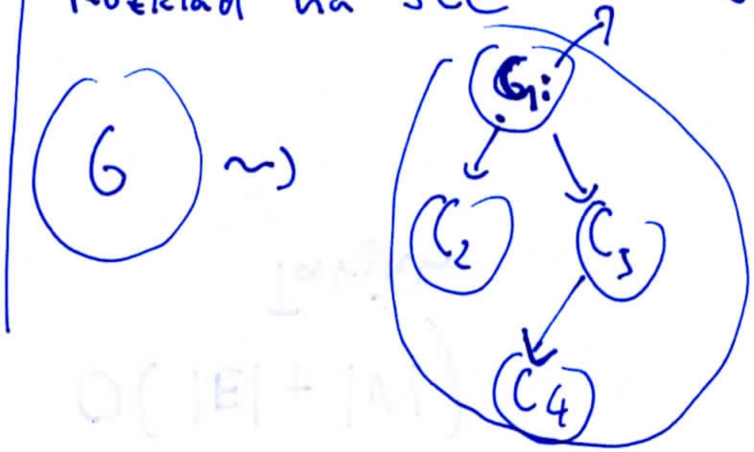
orientovaná







Rozklad na SCC $\qquad C_i \subseteq V$



DFS

Průzkum grafu do hloubky - příklad

barvy

- černou: v byl navštíven a byl prozkoumán
- šedá: v byl navštíven ale nebyl prozkoumán
- bílá: v nebyl navštíven

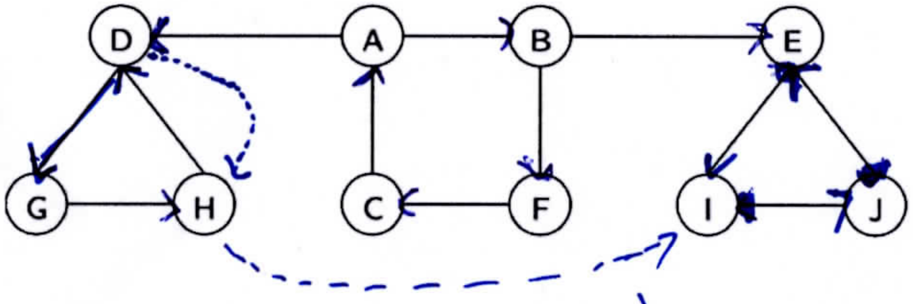
stromové hrany (u,v) (v je bílé)

$u.d < v.d < v.f < u.f$

dopředné hrany —||— (v je šedé)

zpětné hrany (u,v)

$v.d < u.d < u.f < v.f$

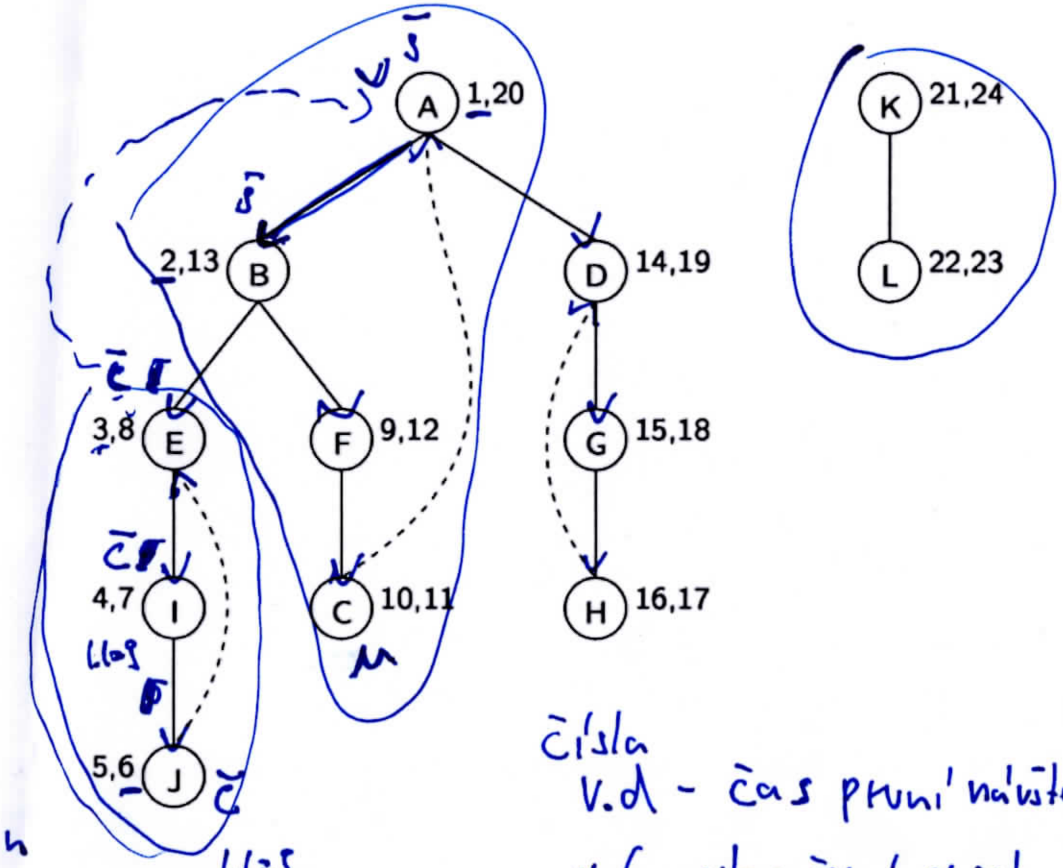


příčné hrany

zásobník

low-link : zpětná hrana s min u.d

v t.z. v.d = low-link ⟶ v je root SCC



čísla

v.d - čas první návštěvy

v.f - ukončení průzkumu

Průzkum do hloubky - iterativní implementace

DFS_Iterative_Visit(G, u)

$S \leftarrow \emptyset$ ⟶ zásobník

$S.push(u)$

$time \leftarrow time + 1;\ u.d \leftarrow time$

$u.color \leftarrow gray$

while $S \neq \emptyset$ **do**

  $u \leftarrow S.pop()$

  if existuje hrana (u, v) taková, že $v.color = white$

    then $S.push(u)$

      $S.push(v)$

      $v.color \leftarrow gray$

      $v.\pi \leftarrow u$

      $time \leftarrow time + 1;\ v.d \leftarrow time$

    else $u.color \leftarrow black$

      $time \leftarrow time + 1;\ u.f \leftarrow time$ **fi od**

$O(|E| + |V|)$

$v \in Adj(u)$

každou hranu v G navštívím jednou (prozkoumám)

Alokace zdrojů

$Z = \{z_1, z_2 \dots z_n\}$

$P = \{p_1, p_2 \dots p_n\}$

Bipartitní grafy

$G = (V_1, V_2, E)$

$V_1 \cap V_2 = \emptyset$

$E = \{(x,y) \mid (x \in V_1 \wedge y \in V_2) \vee (x \in V_2 \wedge y \in V_1)\}$

$\boxed{\text{může dojít k uváznutí} \iff \text{když v } G \text{ je cyklus s hranami } h\ \ v(h) \geq 2}$

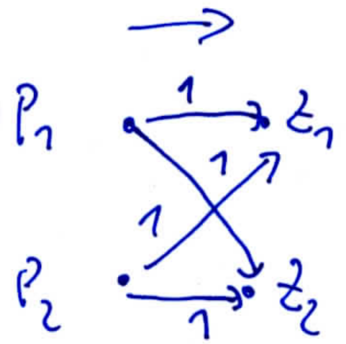
ohodnocení hran

$v: E \to N$

Inicializace

$(u,v) \in E \iff$ proces $\underline{u}$ může žádat zdroj $\underline{v}$

$\wedge\ v((u,v)) = 1$



Proces $\underline{u}$ požádal o zdroj v

$v'((u,v)) = 2$

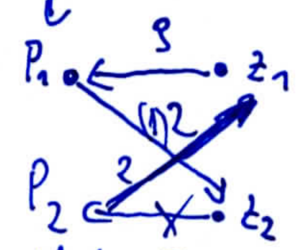
Proces $\underline{u}$ získal zdroj v

$E' = E \setminus \{(u,v)\} \cup \{(v,u)\} \qquad v((v,u)) = 3$

Proces $\underline{u}$ vrátí zdroj $\underline{v}$

$E' = E \setminus \{(v,u)\} \cup \{(u,v)\}$

$v((u,v)) = 1$



P_2 žádá o Z_1

Garbage Collector

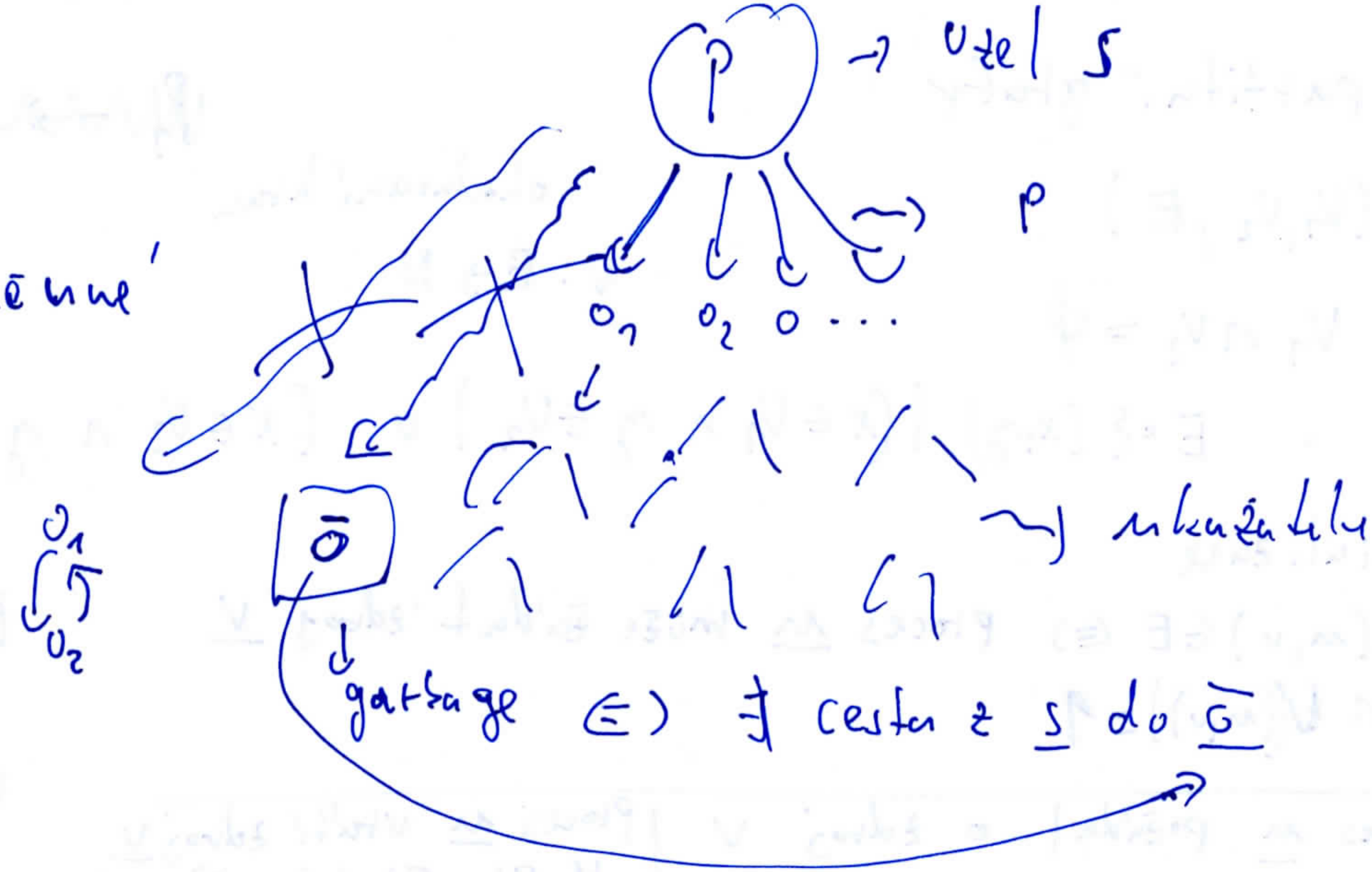
↕

redukce na dosažitelnost v G

O - objekty

E - ukazatele

P - programové proměnné



garbage $\iff$ $\exists$ cesta z $\underline{s}$ do $\underline{\bar{o}}$

Hledání nejkratších cest

$G = (V, E)$

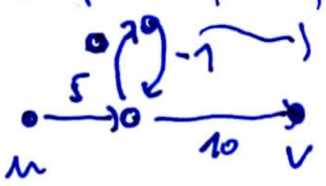
ohodnocení hran $E \to \mathbb{R}$ ⇝ negativní ohodnocení

cesta $p = \langle v_0, v_1, v_2 \cdots v_n \rangle$

$$w(p) = \sum_{i=1}^{n} w(v_{i-1}, v_i)$$

$$\delta(u,v) \stackrel{def}{=} \begin{cases} \min\{w(p) \mid u \stackrel{p}{\leadsto} v\} & \text{existuje cesta z } \underline{u} \text{ do } \underline{v} \\ \infty & \text{jinak} \end{cases}$$

záporný cyklu



-1

- nejkratší cesta mezi $\underline{u}$ a $\underline{v}$
- nejkratší cesta z $\underline{u}$ do všech $\underline{v}$
- nejkratší cesty mezi všemi vrcholy

Belman-Ford → pro obecné grafy

Dijkstra → G bez záporných hran

cena nejkratší cesty z $\underline{s}$ do $\underline{v}$

for each $v \in V$ do
  $d[v] \leftarrow \infty$
  $p[v] \leftarrow nil$
od
$d[s] = 0$

ukládám předchůdce v nejkratší cestě

zdroj

BELLMAN-FORD(G, w, s)
INICIALIZÁCIA(G, s)
for $i = 1$ **to** $|V| - 1$ **do**
  for každú hranu $(u, v) \in H$ **do** RELAXÁCIA(u, v, w) **od od**
for každú hranu $(u, v) \in H$ **do**
  if $d[v] > d[u] + w(u, v)$ **then return** *false* **fi od**
return *true*

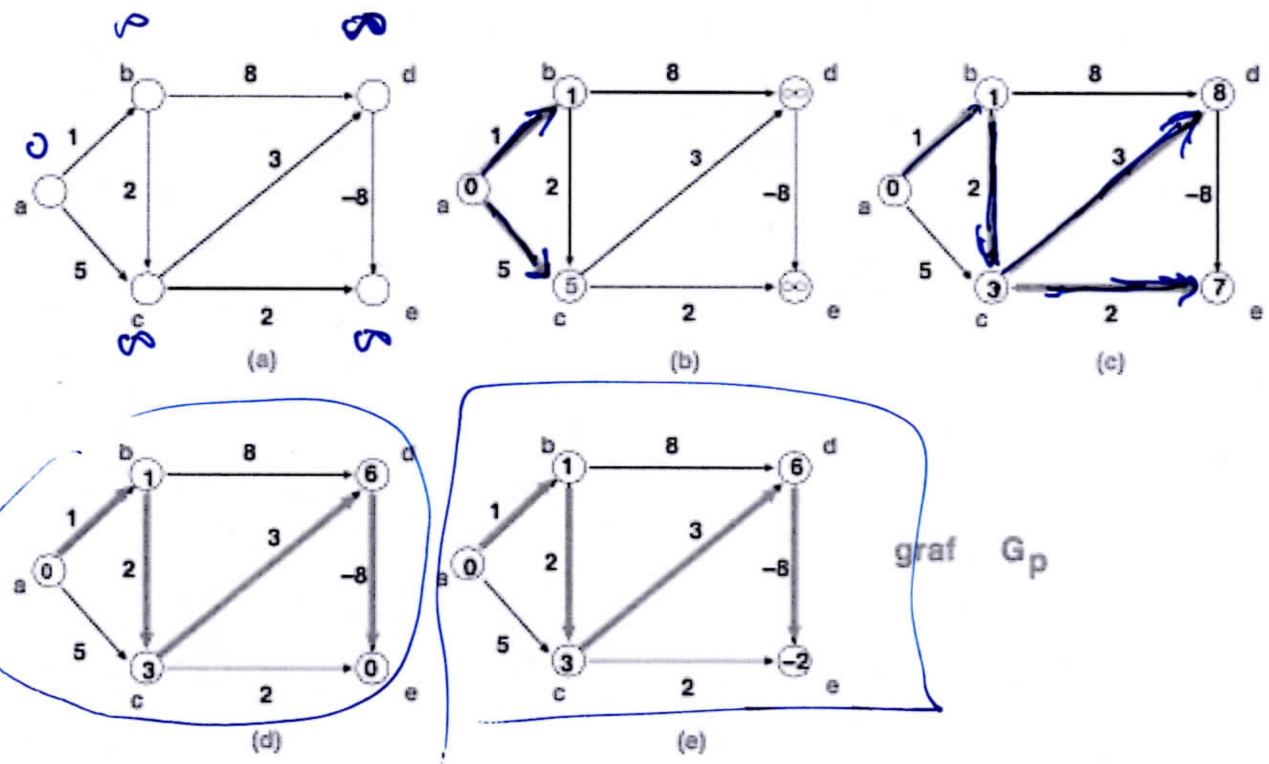
našel jsem záporný cyklus

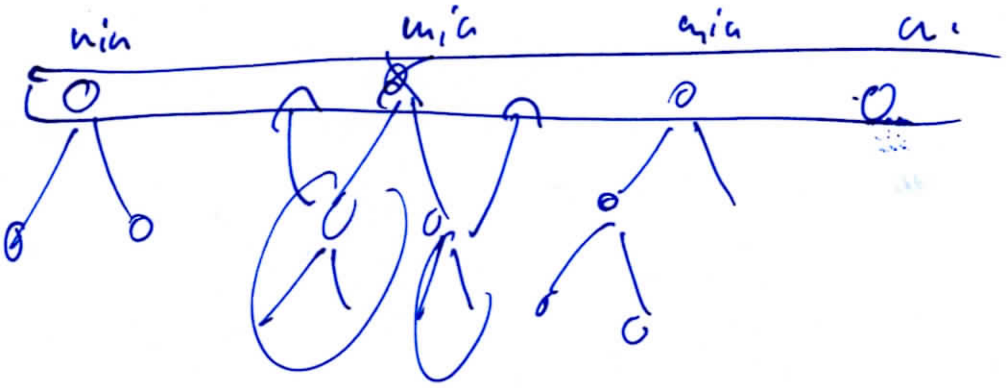
RELAXÁCIA(u, v, w)
if $d[v] > d[u] + w(u, v)$
  then $d[v] \leftarrow d[u] + w(u, v)$
    $p[v] \leftarrow u$
fi

update

$u \xrightarrow{w(u,v)} v$

$O(|V| \cdot |E|)$


a
b
c
d
e
8
1
2
3
−8
5
2
0
1
5
∞
3
6
7
−2
(a)
(b)
(c)
(d)
(e)
graf G_p



Fibonačiho haldu

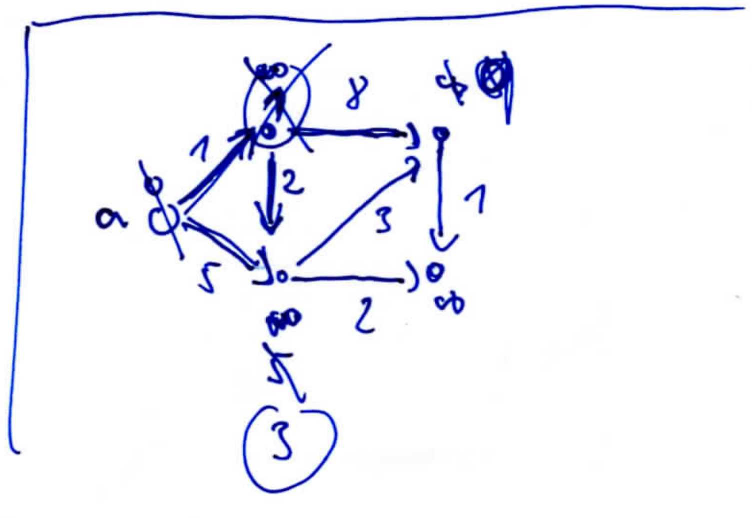
$O(|V| \cdot \log|V| + |E|)$

Extract-min $O(\log|V|)$

$\text{Dijkstra}(G, w, s)$
$\text{Inicializácia}(G, s)$
$S \leftarrow \emptyset$
$Q \leftarrow V$
while $Q \neq \emptyset$ **do**
    $u \leftarrow (u \in Q \wedge d[u] = \min\{d[x] \mid x \in Q\})$
    $S \leftarrow S \cup \{u\}$
    $Q \leftarrow Q \setminus \{u\}$
    for každý vrchol v taký, že $(u, v) \in H$ **do**
        if $d[v] > d[u] + w(u, v)$
            then $d[v] \leftarrow d[u] + w(u, v)$
                $p[v] \leftarrow u$ **fi**
    od od

Decrease-key $O(1)$

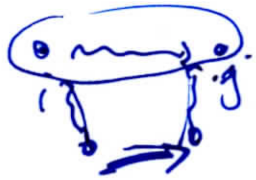
$d[b] = 3$



opakovaný Belman-Ford $O(|V|^2 \cdot |E|)$

—"— Dijkstra $O(|V|^2 \cdot \log(|V|) + |V| \cdot |E|)$

matice sousednosti $W_{ij} = w(v_i, v_j)$



FLOYD-WARSHALL(W)

$D^{(0)} \leftarrow W$

for $k = 1$ **to** n

  do for $i = 1$ **to** n

    do for $j = 1$ **to** n

      do $d_{ij}^{(k)} \leftarrow \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$

    od

  od

od

return $D^{(n)}$

$O(|V|^3)$

časová zložitosť algoritmu je $\mathcal{O}(n^3)$

$$d_{ij}^{(k)} \stackrel{def}{=} \begin{cases} w_{ij} & \text{pokud } k=0 \\ \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) \end{cases}$$

nejkratší cesta z i do j s vnitřními vrcholy z množiny $\{1, \dots, k\}$