Pokročilá algoritmizace

Jiří Vyskočil, Marko Genyg-Berezovskyj

2010

Úvod

- stránky předmětu:

  https://cw.felk.cvut.cz/doku.php/courses/a4m33pal/start

- cíle předmětu

  Cílem je samostatná implementace různých variant standardních (základních nebo mírně pokročilých) úloh v několika vybraných aplikačně bohatých partiích informatiky. Důraz je kladen na algoritmický aspekt úloh a efektivitu praktického řešení. Náplní cvičení bude proto převážně rozbor a příprava jednotlivých implementací, přednáška poskytuje předmětu nezbytný teoretický základ.

- přepoklady

  Kurs předpokládá **schopnost programování** v alespoň jednom z jazyků C/C++/Java. Součástí cvičení jsou programovací úlohy na řešení problematiky PAL. Adept musí ovládat základní datové struktury jako pole, seznam, soubor a musí být schopen manipulovat s daty v těchto strukturách. Na začátku semestru kontrolujeme na cvičeních programátorskou úroveň pomocí konkrétních úloh.

Asymptotické odhady

- horní asymptotický odhad:

$$f(n) \in O(g(n))$$

- význam:

f je shora asymptoticky ohraničená funkcí g (až na konstantu)

- definice:

$$(\exists c > 0)(\exists n_0)(\forall n > n_0) : |f(n)| \leq |c \cdot g(n)|$$

Asymptotické odhady

- dolní asymptotický odhad:

$$f(n) \in \Omega(g(n))$$

- význam:

f je zdola asymptoticky ohraničená funkcí g (až na konstantu)

- definice:

$$(\exists c > 0)(\exists n_0)(\forall n > n_0) : \ |c \cdot g(n)| \leq |f(n)|$$

Asymptotické odhady

- optimální asymptotický odhad:

$$f(n) \in \Theta(g(n))$$

- význam:

f je asymptoticky ohraničená funkcí g z obou stran (až na konstantu)

- definice:

$$(\exists c_1, c_2 > 0)(\exists n_0)(\forall n > n_0): |c_1 \cdot g(n)| < |f(n)| < |c_2 \cdot g(n)|$$

Asymptotické odhady

- příklad: Mějme dvojrozměrné pole MxN celých čísel. Jaká je asymptotická složitost problému nalezení největšího čísla v tomto poli?

- horní:
  - $O((M+N)^2)$ ✓
  - $O(\max(M,N)^2)$ ✓
  - $O(N^2)$ ✗
  - $O(M \cdot N)$ ✓
- dolní:
  - $\Omega(1)$ ✓
  - $\Omega(M)$ ✓
  - $\Omega(M \cdot N)$ ✓

- optimální:
  - $\Theta(M \cdot N)$

Grafy

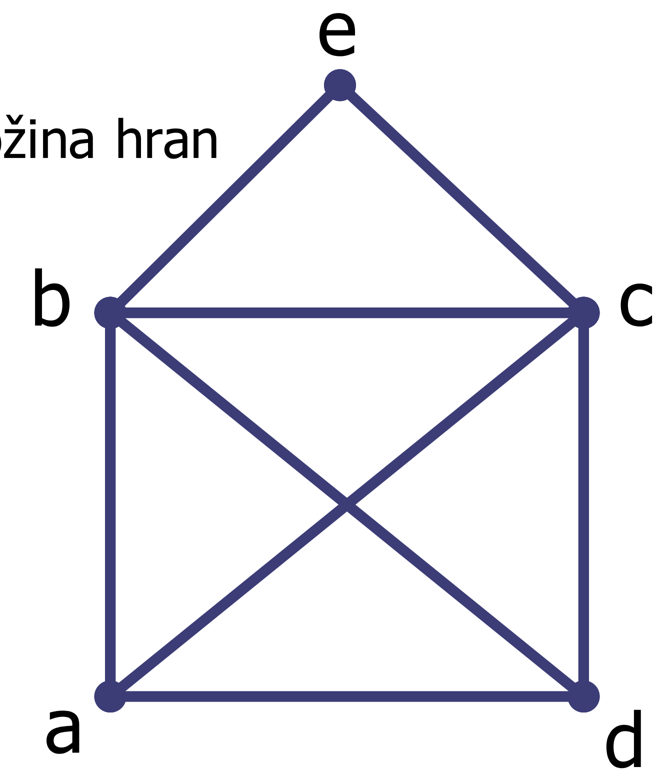
- graf je uspořádaná dvojice množiny vrcholů a množiny hran
- $G = (V, E)$

  kde V je množina vrcholů a E je množina hran

  taková, že platí:

$$E \subseteq \binom{V}{2}$$

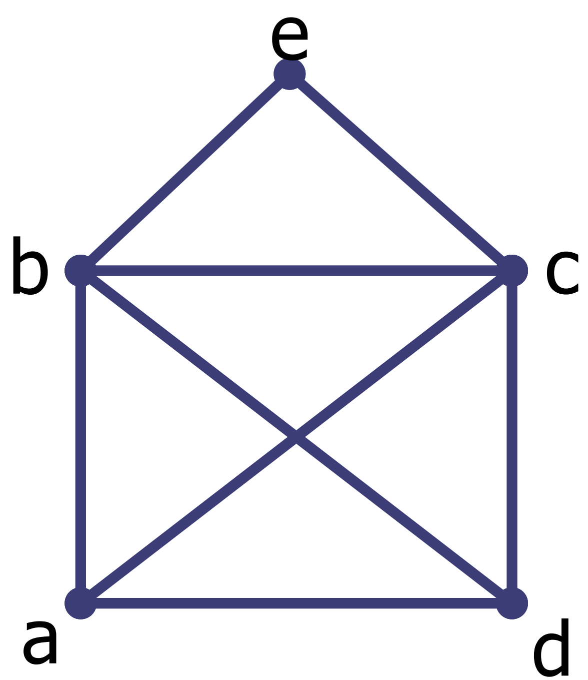
- příklad:
  - V={a,b,c,d,e}
  - E={{a,b},{b,e},{e,c},{c,d},
    {d,a},{a,c},{b,d},{b,c}}



Grafy - orientovanost

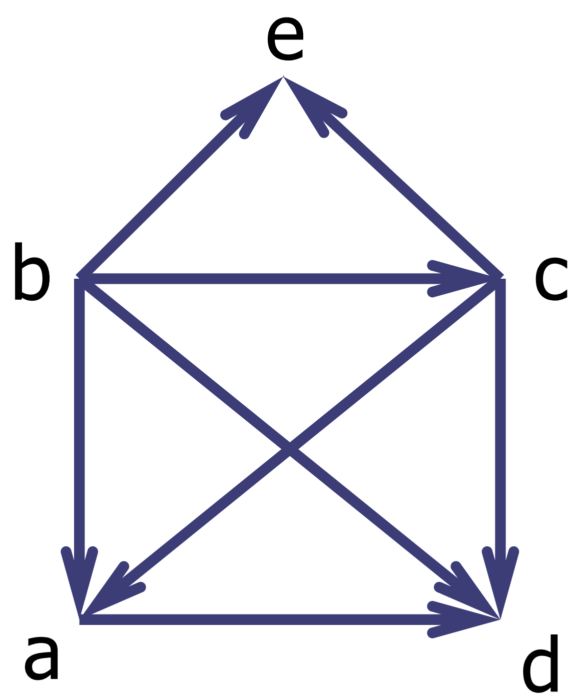
- neorientovaný graf
  - hrana je **neuspořádaná** dvojice vrcholů

E={{a,b},{b,e},{e,c},{c,d},
{d,a},{a,c},{b,d},{b,c}}



- orientovaný graf
  - hrana je **uspořádaná** dvojice vrcholů

E={(b,a),(b,e),(c,e),(c,d),
(a,d),(c,a),(b,d),(b,c)}



Grafy – vážený graf

- vážený graf
  - u každé hrany máme informaci o váze
  - obvykle se váha formalizuje jako váhová funkce:

$$w : E \to \mathbb{R}$$

w({a,b}) = 1.1

w({b,e}) = 2.0

w({e,c}) = 0.3

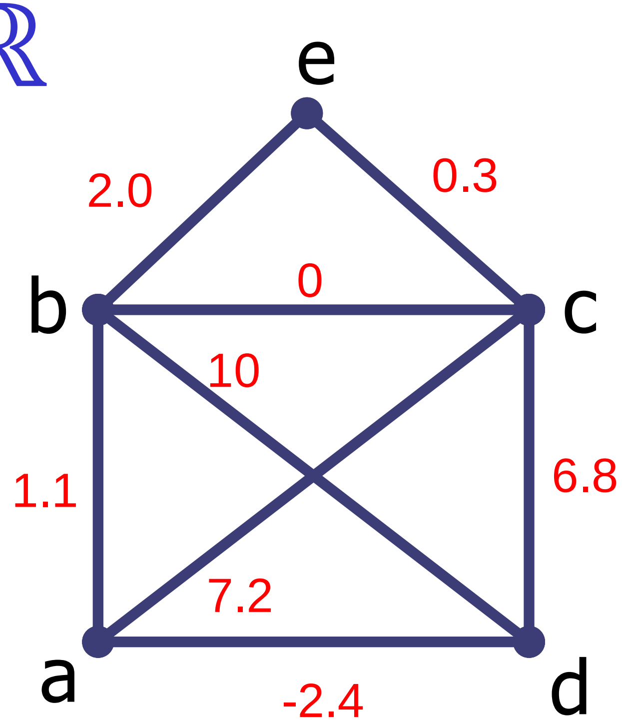
w({c,d}) = 6.8

w({d,a}) = -2.4

w({a,c})= 7.2

w({b,d})= 10

w({b,c})= 0

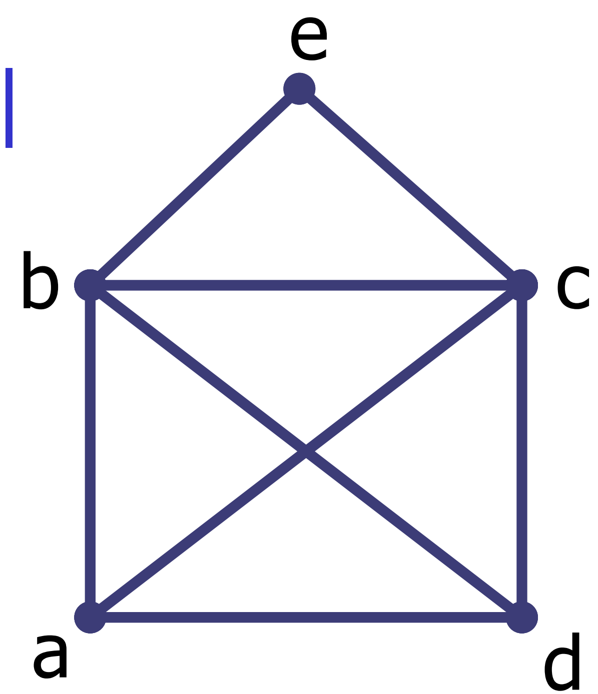


Grafy – stupeň vrcholu

- incidence
  - Pokud jsou dva vrcholy *x*,*y* spojené hranou *e*, říkáme, že vrcholy *x*,*y* jsou incidentní s hranou *e* nebo také, že hrana *e* je incidentní s vrcholy *x*,*y*.
- stupeň vrcholu (pro neorientovaný graf)
  - funkce udávající ke každému vrcholu počet hran s ním incidentních.

$$\deg(u) = |\{e \in E \mid u \in e\}|$$

deg(a)=3
deg(b)=4
deg(c)=4
deg(d)=3
deg(e)=2



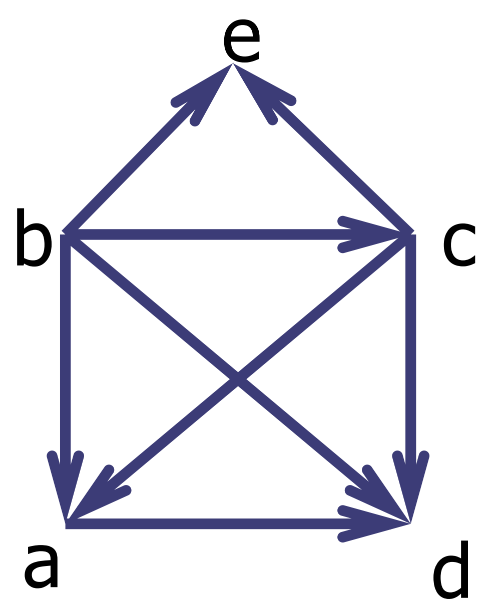
- stupeň vrcholu (pro orientovaný graf)
  - vstupní stupeň

$$deg^+(u) = |\{e \in E \mid (\exists v \in V) : e = (v, u)\}|$$

  - výstupní stupeň

$$deg^-(u) = |\{e \in E \mid (\exists v \in V) : e = (u, v)\}|$$

$deg^+(a)=1$	$deg^-(a)=2$
$deg^+(b)=4$	$deg^-(b)=0$
$deg^+(c)=3$	$deg^-(c)=1$
$deg^+(d)=0$	$deg^-(d)=3$
$deg^+(e)=0$	$deg^-(e)=2$



Grafy – Princip sudosti

- **Princip sudosti** (pro neorientované grafy)

$$\sum_{v \in V} deg(v) = 2|E|$$

- vysvětlení: Každou hranu započítáváme dvakrát - jednou ve vrcholu, kde začíná, podruhé ve vrcholu, kde končí.
- varianta pro orientované grafy

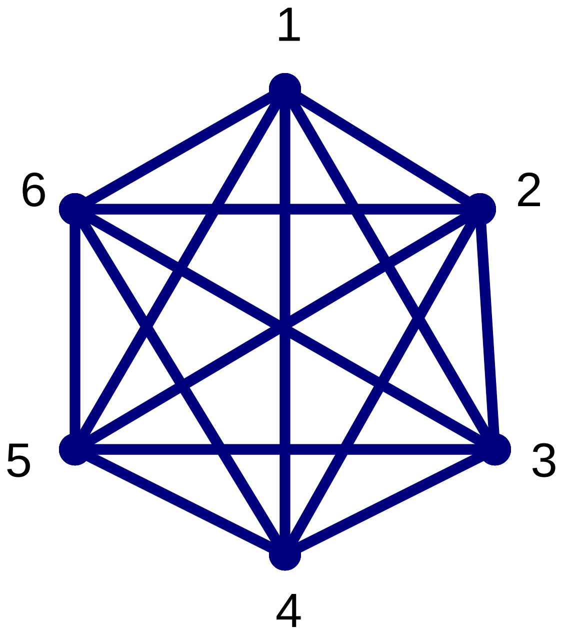
$$\sum_{v \in V} (deg^{+}(v) + deg^{-}(v)) = 2|E|$$

- úplný graf
  - je graf, kde jsou každé dva vrcholy spojené hranou

$$E = \binom{V}{2}$$

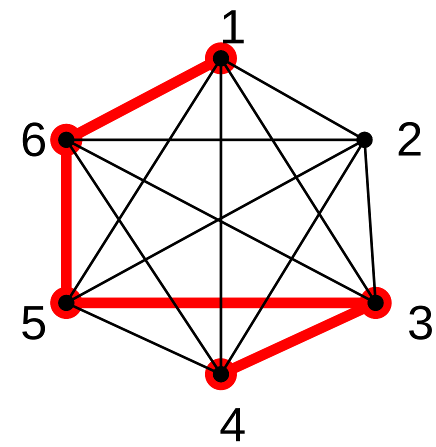
  - důsledkem je

$$(\forall v \in V) : \deg(v) = |V| - 1$$



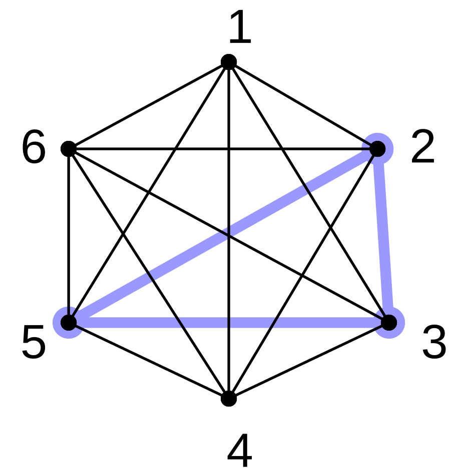
Grafy – cesta, cyklus

- cesta
  - **Cestu** v grafu můžeme chápat jako posloupnost vrcholů a hran $(v_0, e_1, v_1, \ldots, e_t, v_t)$, kde vrcholy $v_0, \ldots, v_t$ jsou *navzájem různé vrcholy* grafu G a pro každé $i = 1,2,\ldots,t$ je $e_i = \{v_{i-1}, v_i\} \in E(G)$.



(1,{1,6},6,{6,5},5,{5,3},3,{3,4},4)

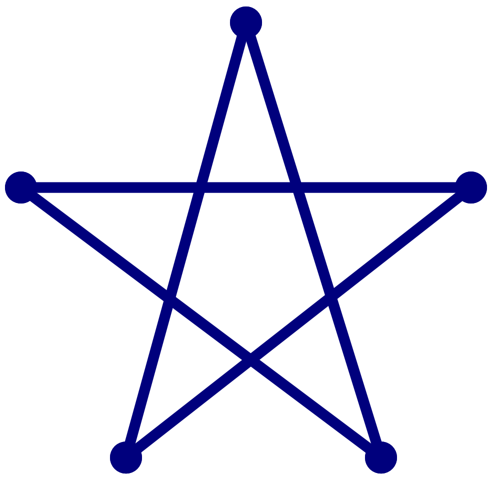
- cyklus
  - **Cyklem** (kružnicí) v grafu rozumíme posloupnost vrcholů a hran $(v_0, e_1, v_1, \ldots, e_t, v_t = v_0)$, kde vrcholy $v_0, \ldots, v_{t-1}$ jsou *navzájem různé vrcholy* grafu G a pro každé $i = 1,2,\ldots,t$ je $e_i = \{v_{i-1}, v_i\} \in E(G)$.



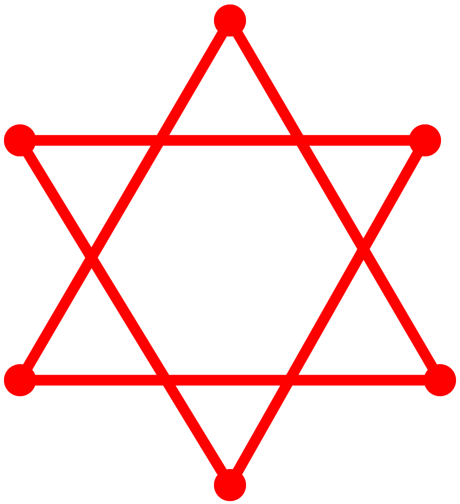
(2,{2,5},5,{5,3},3,{3,2},2)

Grafy - souvislost

- souvislost
  - Řekneme, že graf G je **souvislý**, jestliže pro každé jeho dva vrcholy *x* a *y* existuje v G cesta z *x* do *y*.



souvislý graf



nesouvislý graf

Grafy - stromy

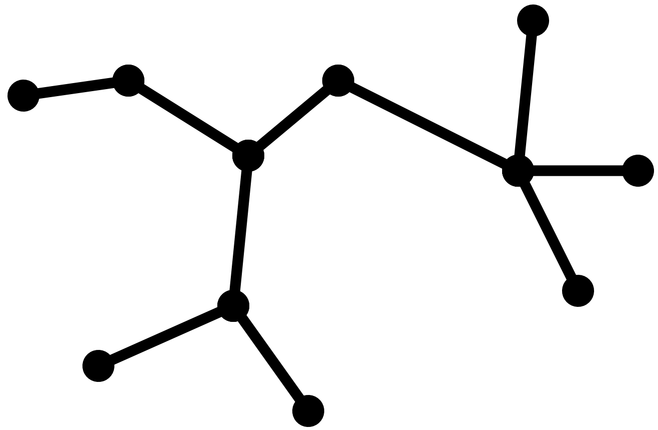
- strom

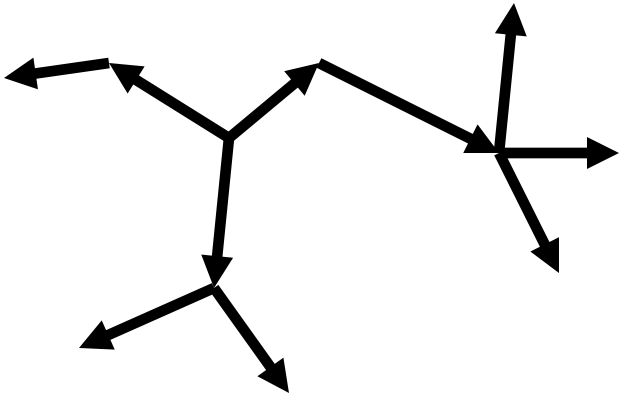
Následující definice stromu jsou ekvivalentní:

- Je souvislý graf bez cyklů.
- Je graf bez cyklů takový, že přidáním libovolné nové hrany vznikne cyklus.
- Je souvislý graf takový, že odebráním libovolné hrany přestane být souvislý.
- Je souvislý graf s |V|-1 hranami.
- Je graf, kde každé dva vrcholy jsou spojeny pouze jednou cestou.

Grafy - stromy

- pro neorientované stromy
  - **List** je vrchol stupně 1.
- pro orientované stromy (někdy bývá orientace opačná!)
  - **List** je vrchol z nějž nevede hrana.
  - **Kořen** je vrchol do nějž nevede hrana.





Grafy – matice sousednosti

- matice sousednosti
  - Nechť G=(V,E) je graf s n vrcholy.

    Označme vrcholy $v_1, \ldots, v_n$ (v nějakém libovolném pořadí). **Matice sousednosti grafu G** je čtvercová matice

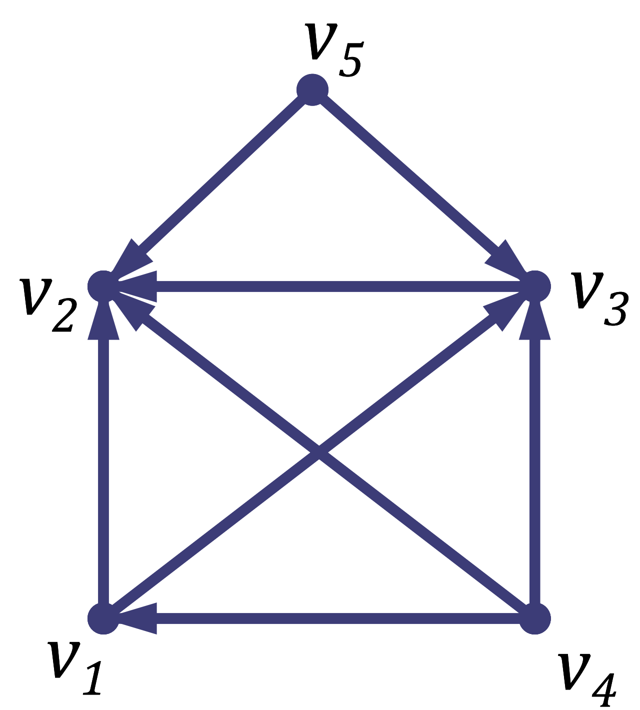
    $$A_{\mathrm{G}} = \left(a_{i,j}\right)_{i,j=1}^{n}$$

    definovaná předpisem

    $$a_{i,j} = \begin{cases} 1 & \text{pro } \{v_i, v_j\} \in E \\ 0 & \text{jinak} \end{cases}$$

Grafy – matice sousednosti (pro orientovaný graf)

	1	2	3	4	5
1	0	1	1	0	0
2	0	0	0	0	0
3	0	1	0	0	0
4	1	1	1	0	0
5	0	1	1	0	0



Grafy – Laplaceova matice

- Laplaceova matice
  - Nechť G=(V,E) je graf s n vrcholy

    Označme vrcholy $v_1, \ldots, v_n$ (v nějakém libovolném pořadí). **Laplaceova matice grafu G** je čtvercová matice

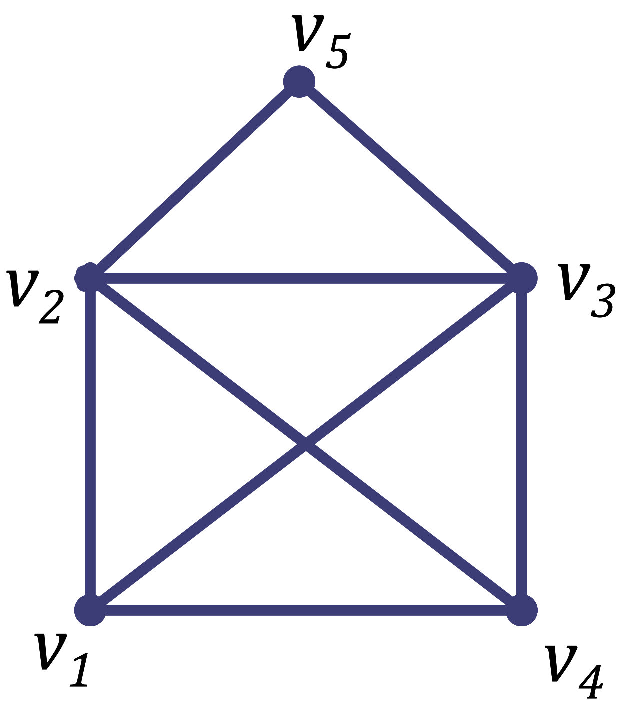
$$L_G = \left(l_{i,j}\right)_{i,j=1}^{n}$$

definovaná předpisem

$$l_{i,j} = \begin{cases} \deg(v_i) & \text{pro } i = j \\ -1 & \text{pro } \{v_i, v_j\} \in E \\ 0 & \text{jinak} \end{cases}$$

Grafy – Laplaceova matice

	1	2	3	4	5
1	3	-1	-1	-1	0
2	-1	4	-1	-1	-1
3	-1	-1	4	-1	-1
4	-1	-1	-1	3	0
5	0	-1	-1	0	2



Grafy – matice vzdáleností

- matice vzdáleností
  - Nechť G=(V,E) je graf s n vrcholy a váhovou funkcí w.

    Označme vrcholy $v_1, \ldots, v_n$ (v nějakém libovolném pořadí). **Matice vzdáleností grafu G** je čtvercová matice

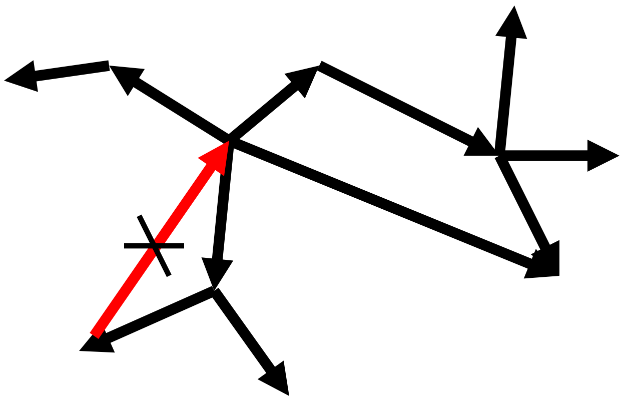
$$A_{\mathrm{G}} = \left(a_{i,j}\right)_{i,j=1}^{n}$$

definovaná předpisem

$$a_{i,j} = \begin{cases} w(\{v_i, v_j\}) & \text{pro } \{v_i, v_j\} \in E \\ 0 & \text{jinak} \end{cases}$$

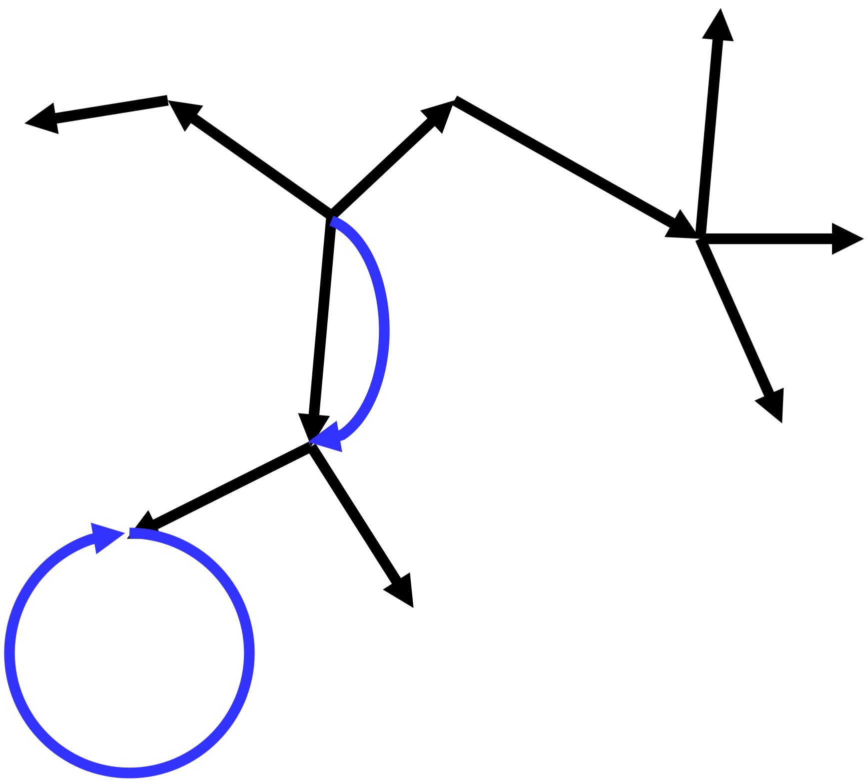
Grafy – DAG

- DAG (Directed Acyclic Graph)
  - DAG je orientovaný graf bez cyklů (=acyklický)



Grafy – multigraf

- multigraf
  - Je graf, ve kterém povolujeme vícečetné hrany a někdy také hrany nad jedním vrcholem.



Grafy – matice incidence

- matice incidence
  - Nechť G=(V,E) je graf kde |V|=n a |E|=m.

    Označme vrcholy $v_1, \ldots, v_n$ (v nějakém libovolném pořadí) a hrany $e_1, \ldots, e_m$ (v nějakém libovolném pořadí). **Matice incidence grafu G** je matice typu

$$\{-1, 0, +1\}^{n \times m}$$

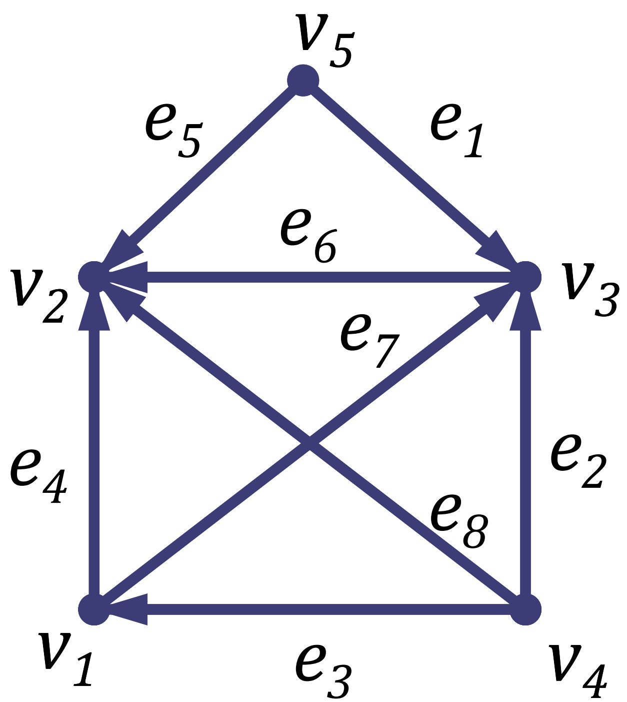
definovaná předpisem

$$(I)_{i,j} = \begin{cases} -1 & \text{pro } e_j = (v_i, *) \\ +1 & \text{pro } e_j = (*, v_i) \\ 0 & \text{jinak} \end{cases}$$

Jinými slovy, každá hrana má -1 u vrcholu, kde začíná a +1 tam, kde končí. U neorientovaných grafů je na obou místech +1.

Grafy – matice incidence

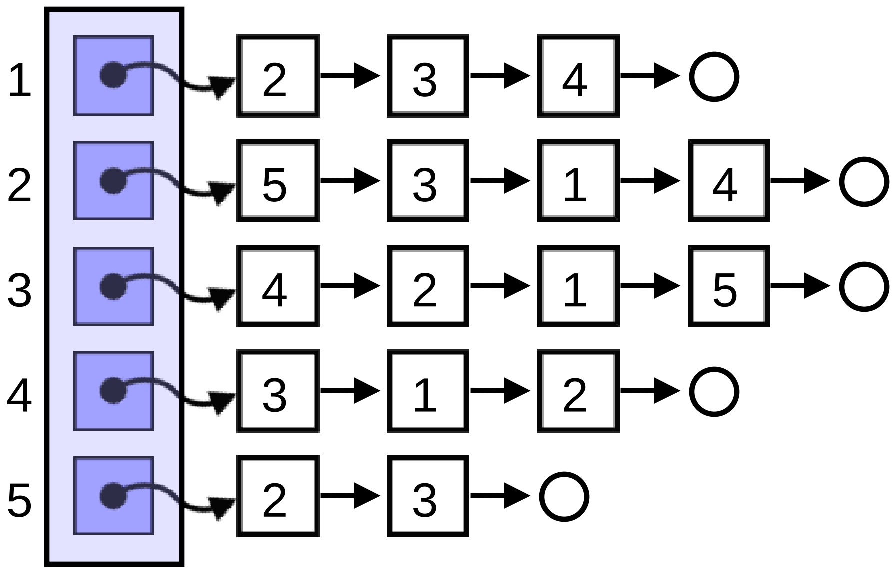
	1	2	3	4	5	6	7	8
1	0	0	1	-1	0	0	-1	0
2	0	0	0	1	1	1	0	1
3	1	1	0	0	0	-1	1	0
4	0	-1	-1	0	0	0	0	-1
5	-1	0	0	0	-1	0	0	0

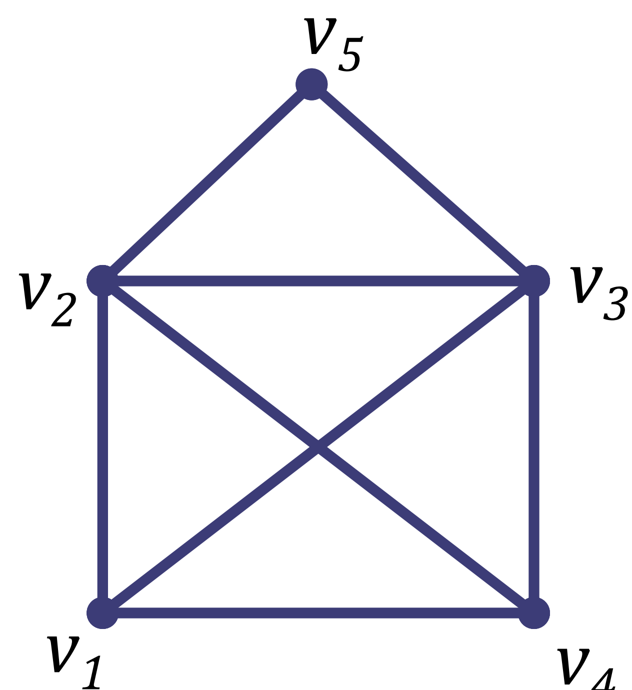


Grafy – seznam sousedů

- seznam sousedů
  - Nechť G=(V,E) je graf s n vrcholy

    Označme vrcholy $v_1, \ldots, v_n$ (v nějakém libovolném pořadí). **Seznam sousedů grafu G** je pole P ukazatelů velikosti n, kde P[i] ukazuje na spojový seznam indexů všech vrcholů, se kterými je vrchol v_i spojen hranou (lze definovat i případě orientovaného grafu).





Grafy - DFS

- DFS (Depth First Search) procházení grafu do hloubky

```
procedure dfs(start_vertex : Vertex)
var     to_visit : Stack = empty;
        visited : Vertices = empty;
{
        to_visit.push(start_vertex);
        while (size(to_visit) != 0) {
                v = to_visit.pop();
                if v not in visited then {
                        visited.add(v);
                        for all x in neighbors of v {
                                to_visit.push(x);
                        }
                }
        }
}
```

- BFS (Breadth First Search) procházení grafu do šířky

```
procedure bfs(start_vertex : Vertex)
var     to_visit : Queue = empty;
        visited : Vertices = empty;
{
        to_visit.push(start_vertex);
        while (size(to_visit) != 0) {
                v = to_visit.pop();
                if v not in visited then {
                        visited.add(v);
                        for all x in neighbors of v {
                                to_visit.push(x);
                        }
                }
        }
}
```

Grafy – prioritní fronta

- prioritní fronta
  - Je fronta s operací **vlož do fronty s prioritou**.
  - V případě, že je priorita nejnižší, chová se jako vkládání (push) do normální fronty.
  - V případě, že je priorita nejvyšší, chová se jako vkládání na zásobník.
  - Pomocí prioritní fronty lze realizovat DFS i BFS pouhou změnou priority vkládání prvku.