[1] Asymptotic complexity recapitulation, Graph representation

- **asymptotická horní mez**
    - hodnota funkce „f“ menší či rovna hodnotě funkce „g“ (s konstantním faktorem)

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

- **asymptotická dolní mez**
    - hodnota funkce „f“ větší či rovna hodnotě funkce „g“ (s konstantním faktorem)

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

- **asymptotická těsná mez**
    - hodnota obou funkcí je stejná (s konstantním faktorem)

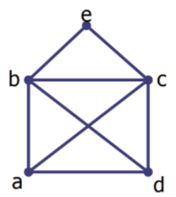
$$f(n) \in \Theta(g(n))$$
$$(\exists c_1, c_2 > 0)(\exists n_0)(\forall n > n_0): c_1 \cdot g(n) < f(n) < c_2 \cdot g(n)$$

Grafy

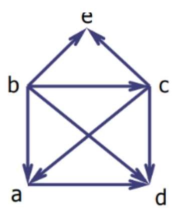
- **graf** = uspořádaná dvojice množiny vrcholů (vertices, nodes) a množiny hran (edges, arcs)
- **G = (V, E)**
    - V = množina vrcholů; E = množina hran

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

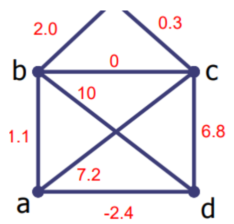
- **příklad**:
    - V = {a,b,c,d,e}
    - E = {{a,b},{b,e},{e,c},{c,d},{d,a},{a,c},{b,d},{b,c}}
- **neorientovaný graf (undirected)**
    - hrany jsou neuspořádané dvojice vrcholů
    - např. E = {{a,b},{b,e},{e,c},{c,d},{d,a},{a,c},{b,d},{b,c}}



- **orientovaný graf (directed, digraph)**
    - hrany jsou uspořádané dvojice vrcholů
    - např. E = {(b,a),(b,e),(c,e),(c,d),(a,d),(c,a),(b,d),(b,c)}



- **ohodnocený graf (weighted graph)**
    - každé hraně je přiřazena číselná hodnota
    - často formalizováno funkcí w: E → $\mathbb{R}$
    - např. w({a,b}) = 1.1, w({b,e}) = 2.0,…



- **stupeň vrcholu =** funkce vracející počet sousedů daného vrcholu (vrcholů spojených hranou)
    - deg(u) = |{e ∈ E | u ∈ e}
    - viz předchozí obrázky – deg(a) = 3, deg(e) = 2,…
    - u orientovaného grafu se dále rozlišuje na **indegree** (vstupní) a **outdegree** (výstupní)

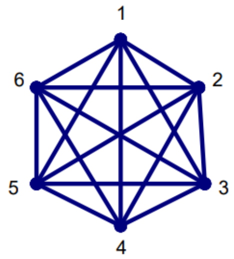
$$deg^{+}(u) = |\{e \in E \mid (\exists v \in V) : e = (v,u)\}|$$
$$deg^{-}(u) = |\{e \in E \mid (\exists v \in V) : e = (u,v)\}|$$

- **handshaking lemma (princip sudosti)**
    - každá hrana přispívá dvakrát – jednak pro výchozí, druhak pro cílový vrchol
    - mírně odlišná varianta pro orientované grafy

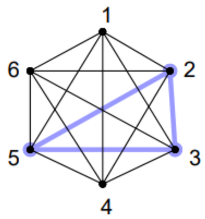
$$\sum_{v \in V} deg(v) = 2|E|$$
$$\sum_{v \in V} (deg^{+}(v) + deg^{-}(v)) = 2|E|$$

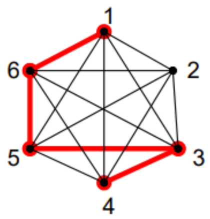
- **úplný graf (complete graph)**
    - každé dva vrcholy jsou spojené hranou

$$|E| = \binom{V}{2}$$
$$(\forall v \in V) : \deg(v) = |V| - 1$$

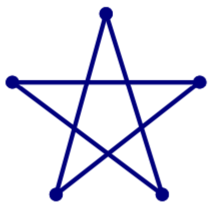


- **cesta (path)**
    - posloupnost vrcholů a hran (v_0, e_1, v_1,…, e_t, v_t), kde se neopakují vrcholy a pro každé i = 1,2,…t platí, že $e_i = \{v_{i-1}, v_i\} \in E(G)$





- **kružnice (circuit) =** uzavřená neorientovaná cesta
- **cyklus** = uzavřená orientovaná cesta
- **souvislý graf (connected graph) =** pro každou dvojici x a y existuje v grafu cesta z x do y



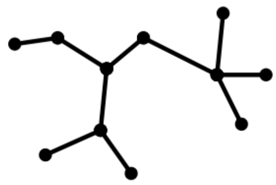
Connected graph

Disconnected graph

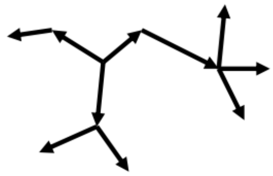
Stromy

- existuje mnoho ekvivalentních definic… G je **strom**, pokud:
    - G je souvislý graf bez cyklů
    - G je souvislý graf, kde se objeví cyklus, přidá-li se jakákoliv hrana
    - G je souvislý graf, který se stane nesouvislým, odebere se kterákoliv hrana
    - G je souvislý graf s |V|-1 hranami
    - G je graf, ve kterém jsou každé dva vrcholy spojeny pouze jednou cestou

- **neorientovaný strom**
    - **list** (**leaf**) je vrchol se stupněm 1



- **orientovaný strom**
    - list je vrchol bez výstupních hran
    - **kořen** (**root**) je vrchol bez vstupních hran



Reprezentace grafů

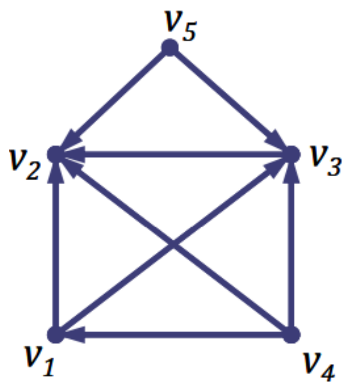
Matice sousednosti (adjacency matrix)

- Nechť G = (V,E) je graf s *n* vrcholy (v_1,… v_n).
- Matice sousednosti je pak čtvercová matice A_G, definovaná následovně:

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

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

	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

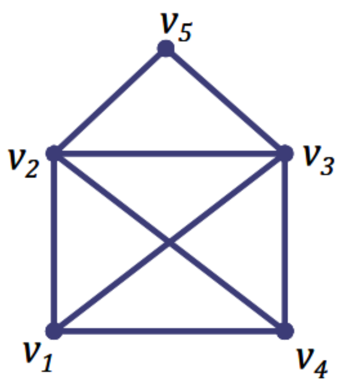


Laplaceova matice

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

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

	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



Matice vzdálenosti (distance matrix)

- Stejné jako u předchozích matic, a dále pak funkce w (ohodnocení grafu).

$$A_G = (a_{i,j})_{i,j=1}^{n}$$

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

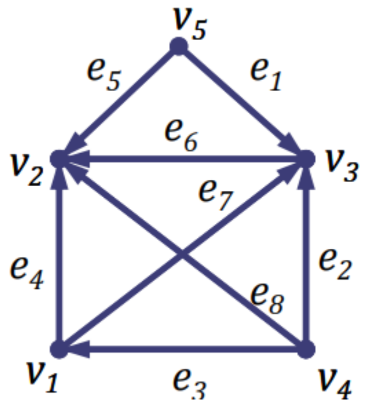
Matice incidence

- Nechť G = (V,E) je graf, kde |V| = n, |E| = m.
- Matice incidence je matice typu $\{-1,0,1\}^{n \times m}$, definovaná takto:

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

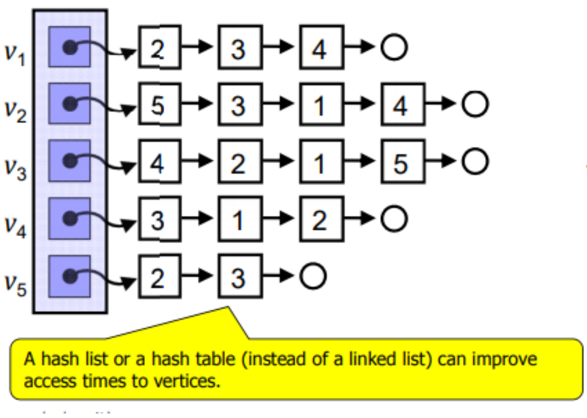
Tj. každá hrana má -1 u výchozího vrcholu a +1 u koncového vrcholu.
U neorientovaného grafu jsou u obou +1.

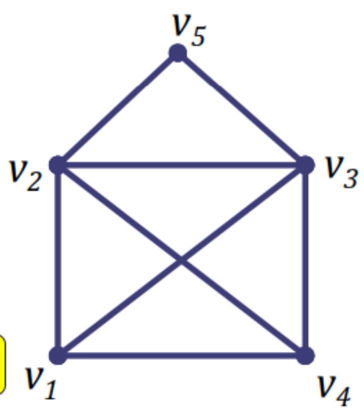
	1	2	3	4	5	6	7	8
1	0	0	1	-1	0	0	-1	0
2	0	0	0	1	1	1	0	1
3	1	1	0	0	0	-1	1	0
4	0	-1	-1	0	0	0	0	-1
5	-1	0	0	0	-1	0	0	0



Seznam sousedů (adjacency list)

- pro každý vrchol grafu si udržujeme seznam jeho sousedů
- mohlo by to být např. pole ukazatelů P o velikosti n, kde P[i] ukazuje na spojový seznam indexů všech vrcholů, které jsou s vrcholem v_i spojeny hranou



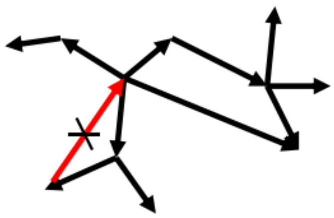


Porovnání složitosti operací v jednotlivých reprezentacích

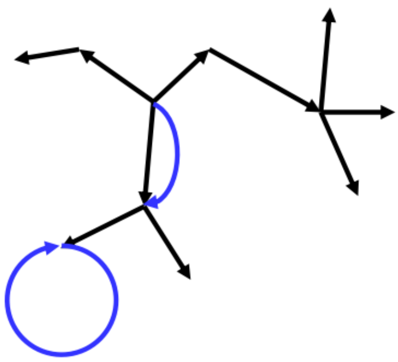
	Adjacency Matrix	Laplacian Matrix	Adjacency List	Incidence Matrix
Storage	$\lvert V\rvert \cdot \lvert V\rvert \in O(\lvert V\rvert^2)$		$O(\lvert V\rvert+\lvert E\rvert)$	$\lvert V\rvert \cdot \lvert E\rvert \in O(\lvert V\rvert \cdot \lvert E\rvert)$
Add vertex	$O(\lvert V\rvert^2)$		$O(\lvert V\rvert)$	$O(\lvert V\rvert \cdot \lvert E\rvert)$
Add edge	$O(1)$			$O(\lvert V\rvert \cdot \lvert E\rvert)$
Remove vertex	$O(\lvert V\rvert^2)$		$O(\lvert E\rvert)$	$O(\lvert V\rvert \cdot \lvert E\rvert)$
Remove edge	$O(1)$		$O(\lvert V\rvert)$	$O(\lvert V\rvert \cdot \lvert E\rvert)$
Query: are vertices u, v adjacent?	$O(1)$		$deg(v) \in O(\lvert V\rvert)$	$O(\lvert E\rvert)$
Query: get node degree of vertex v (=deg(v)), access all vertex neighbours	$\lvert V\rvert \in O(\lvert V\rvert)$	$O(1)$	$deg(v) \in O(\lvert V\rvert)$	$\lvert E\rvert \in O(\lvert E\rvert)$
Remarks	Slow to add or remove vertices, because matrix must be resized/copied		When removing edges or vertices, need to find all vertices or edges	Slow to add or remove vertices and edges, because matrix must be resized/copied

Další grafové pojmy

- **DAG (directed acyclic graph)** = orientovaný graf bez cyklů

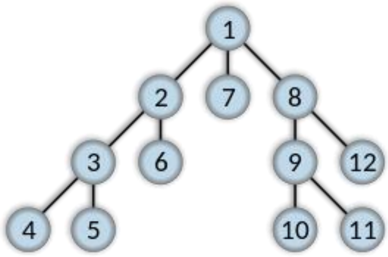


- **multigraf / pseudograf** = graf, v němž může existovat více stejnosměrných (rovnoběžných) hran mezi dvěma uzly, a také hrany, jejichž výchozí a cílový vrchol je tentýž

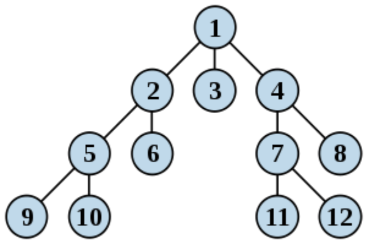


Průchody grafem

DFS – Depth First Search (do hloubky)



BFS – Breadth First Search (do šířky)



Prioritní fronta (priority queue)

- insert operace s prioritou
- je-li priorita nového prvku nejnižší, je operace stejná jako push do klasické fronty
- je-li priorita nejvyšší, operace se chová jako push do zásobníku (stack)
- jak DFS, tak BFS se dají naimplementovat s pomocí prioritní fronty s danou prioritou prvku při jeho vkládání (priorita tedy může např. monotónně klesat)

Zdroj: https://cw.fel.cvut.cz/wiki/_media/courses/b4m33pal/2011pal01.pdf