Diskrétní struktury 1

Grafy

Radim Bělohlávek

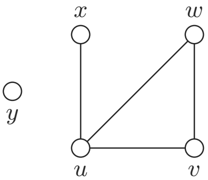
KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

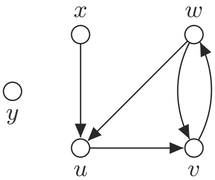
Co jsou grafy

- graf = (místa, spojení mezi nimi)
- (města, silnice/železnice), (webové stránky, odkazy mezi stránkami), (lidé,vztahy mezi nimi), ...
- úlohy:
  - najít nejkratší cestu mezi dvěma městy
  - vybudovat nejlevnější síť propojující všechna místa
  - projít všechna místa a urazit při tom nejmenší vzdálenost
  - najít velkou skupinu navzájem propojených lidí (komunitu)
- teorie grafů = jedna z nejpoužívanějších oblastí diskrétní matematiky

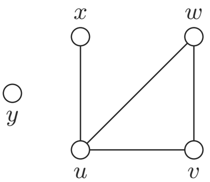
Definice grafu

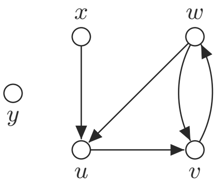
Příklad: neorientovaný (vlevo) a orientovaný graf:





Definice Neorientovaný graf je dvojice $G = \langle V, E \rangle$, kde V je neprázdná množina tzv. vrcholů (někdy také uzlů) a $E \subseteq \{\{u, v\} \mid u, v \in V,\ u \neq v\}$ je množina dvouprvkových množin vrcholů, tzv. (neorientovaných) hran.
Orientovaný graf je dvojice $G = \langle V, E \rangle$, kde V je neprázdná množina tzv. vrcholů (uzlů) a $E \subseteq V \times V$ je množina uspořádaných dvojic vrcholů, tzv. (orientovaných) hran.





Neorientovaný graf $G_1 = \langle V, E_1 \rangle$ vlevo:
$V_1 = \{u, v, w, x, y\}$ $E_1 = \{\{u, v\}, \{u, w\}, \{u, x\}, \{v, w\}\}$

Orientovaný graf $G_2 = \langle V, E_2 \rangle$ vpravo:
$V_2 = \{u, v, w, x, y\}$, $E_2 = \{\langle u, v\rangle, \langle v, w\rangle, \langle w, u\rangle, \langle w, v\rangle, \langle x, u\rangle\}$.

Příklady grafů $G = \langle V, E\rangle$:

- V = křižovatky v daném městě, $\langle k_1, k_2\rangle \in E$ ... existuje ulice z k_1 do k_2
- V = množina lidí, $\{c, d\} \in E$ ... c a d se znají.

  Neorientovaný graf „známosti“ (sociální síť).

  Velikost: 7,8 mld vrcholů (2020). Počet hran = ?

  (Dunbarovo číslo: člověk je schopen udržovat aktivní vztah s max. 150 lidmi.)
- V = množina webových stránek, $\langle p_1, p_2\rangle \in E$ ... z p_1 je odkaz (hyperlink) na p_2

  Orientovaný graf „web“ (tzv. webgraph).

  Velikost: 1,74 mld. vrcholů (InternetLiveStats 2020, aktivních asi 25%).

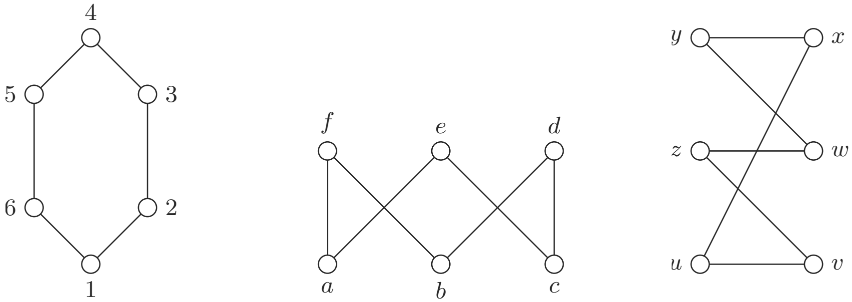
  Jiné zdroje: 5.49 mld. (WorldWideWebSize.com 2020).

  Počet hran: např. 2014 Common Crawl Corpus: 1,72 mld stránek, 64 mld odkazů.
- V = množina neuronů v mozku, $\langle n_1, n_2\rangle \in E$ ... z neuronu n_1 vede vlákno do n_2

  Velikost: 10 mld. neuronů, jeden neuron propojen s cca 10^4 dalšími neurony.

Izomorfní grafy

- Dva různé obrázky mohou mohou popisovat v zásadě stejné grafy.
- „v zásadě stejné“ = mají stejnou strukturu; říkáme, že jsou izomorfní.
- Jsou následující grafy izomorfní?



Definice Nechť $G_1 = \langle V_1, E_1 \rangle$ a $G_2 = \langle V_2, E_2 \rangle$ jsou neorientované grafy. Bijekce $h : V_1 \to V_2$ se nazývá izomorfismus G_1 do G_2, pokud pro každé vrcholy $u, v \in V_1$ je

$$\{u, v\} \in E_1 \quad \text{právě když} \quad \{h(u), h(v)\} \in E_2.$$

Nechť $G_1 = \langle V_1, E_1 \rangle$ a $G_2 = \langle V_2, E_2 \rangle$ jsou orientované grafy. Bijekce $h : V_1 \to V_2$ se nazývá izomorfismus G_1 do G_2, pokud pro každé vrcholy $u, v \in V_1$ je

$$\langle u, v \rangle \in E_1 \quad \text{právě když} \quad \langle h(u), h(v) \rangle \in E_2.$$

Pokud izomorfismus G_1 do G_2 existuje, nazývají se G_1 a G_2 izomorfní. V takovém případě píšeme $G_1 \cong G_2$.

Poznámka:
(a) Tedy izomorfní grafy se liší jen „přejmenováním vrcholů“. Přejmenování = zobrazení h.
(b) Př. $h(1) = a$, $h(2) = e$, $h(3) = c$, $h(4) = d$, $h(5) = b$, $h(6) = f$ na předchozím slajdu.
(c) Relace být izomorfní je ekvivalence na třídě všech grafů (ověřte).

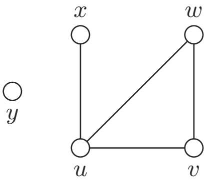
Podgrafy

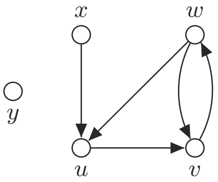
Definice (Orientovaný nebo neorientovaný) graf $\langle V_1, E_1 \rangle$ je podgrafem grafu $\langle V_2, E_2 \rangle$, právě když $V_1 \subseteq V_2$ a $E_1 \subseteq E_2$.

Podgraf $\langle V_1, E_1 \rangle$ grafu $\langle V_2, E_2 \rangle$ se nazývá indukovaný, právě když E_1 obsahuje každou hranu z E_2, jejíž oba koncové vrcholy patří do V_1.

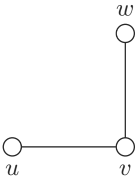
Poznámka: Podgrafy $=$ smysluplné části grafů.

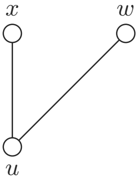
Dané grafy:

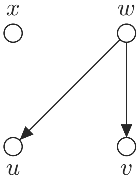




První dva následující grafy jsou podgrafy grafu vlevo nahoře.
První z nich není indukovaný (chybí v něm hrana $\{u, w\}$), druhý ano.
Třetí graf je podgrafem grafu vpravo nahoře.







Cestování v grafech

- vrcholy jako místa a hrany jako spojnice mezi místy: problémy cestování
- efektivní algoritmy: jak se dostat nejkratší cestou z u do v
- různé interpretace: graf $=$ počítačová síť, cestující $=$ blok informací (paket) v síti.

Definice Sled v (neorientovaném nebo orientovaném) grafu $G = \langle V, E\rangle$ je posloupnost

$$v_0, e_1, v_1, e_2, v_2, \ldots, e_n, v_n,$$

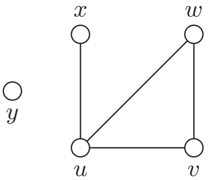
kde $v_i \in V$ jsou vrcholy, $e_j \in E$ jsou hrany a platí, že

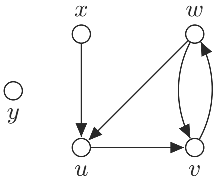
- $e_i = \{v_{i-1}, v_i\}$ pro $i = 1, \ldots, n$, je-li G neorientovaný,
- $e_i = \langle v_{i-1}, v_i\rangle$ pro $i = 1, \ldots, n$, je-li G orientovaný.

Číslo n se nazývá délka sledu. Sled $v_0, e_1, v_1, e_2, v_2, \ldots, e_n, v_n$, se nazývá

- uzavřený, je-li $v_0 = v_n$,
- tah, neopakuje-li se v něm žádná hrana (tj. pro $i \neq j$ je $e_i \neq e_j$),
- cesta, neopakuje-li se v něm žádný vrchol (tj. pro $i \neq j$ je $v_i \neq v_j$),
- kružnice, je-li tahem, $v_0 = v_n$ a s výjimkou vrcholů v_0 a v_n jsou každé dva vrcholy různé.

Vzdálenost z vrcholu u do vrcholu v je délka cesty z u do v, která má ze všech cest z u do v délku nejmenší.





V grafu vlevo:

- $u, \{u, w\}, w, \{w, v\}, v, \{v, u\}, u, \{u, w\}, w$ je sled, který není tahem (a tedy ani cestou), protože se v něm opakuje hrana $\{u, w\}$.
- Sled $u, \{u, w\}, w, \{w, v\}, v, \{v, u\}, u, \{u, x\}, x$ je tah, který není cestou, protože se v něm opakuje vrchol u.
- Sled $x, \{x, u\}, u, \{u, w\}, w, \{w, v\}, v, \{v, u\}, u, \{u, x\}, x$ je sice uzavřený, ale není to kružnice, protože se v něm opakuje vrchol u.
- Sled $u, \{u, w\}, w, \{w, v\}, v, \{v, u\}, u,$ je kružnice.

Věta Existuje-li v grafu sled z vrcholu u do vrcholu v, existuje také cesta z u do v.

Důkaz Opakuje-li se ve sledu $u, \dots, v$ nějaký vrchol w, tj. má-li sled tvar $u, \dots, w, \dots, w, \dots, v$, vynecháme posloupnost $w, \dots,$.

Dostaneme $u, \dots, w, \dots, v$, což je také sled z u do v.

Pokud je už cestou, jsme hotovi. Pokud ne, opět vynecháme posloupnost mezi opakujícími se vrcholy. Protože je sled konečný, po konečném počtu kroků takto skončíme u cesty z u do v. $\square$

Všimněme si:

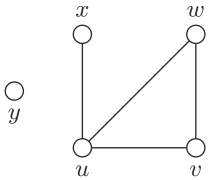
- Sled $=$ cestování bez omezení.
- Najít vhodný tah se bude snažit např. poštovní doručovatelka.
- Hledání cest: ve spedičních firmách při rozvozu zboží; projet místem vícekrát je nehospodárné.

Definice Neorientovaný graf $G = \langle V, E\rangle$ se nazývá souvislý, právě když pro každé dva vrcholy $u, v \in V$ existuje sled z u do v. Komponenta neorientovaného grafu je každý jeho maximální souvislý podgraf.

Tedy:

(a) Komponenta grafu $G = \langle V, E\rangle$ je jeho podgraf $G' = \langle V', E'\rangle$, který je souvislý a pro který platí, že je-li $G'' = \langle V'', E''\rangle$ souvislý podgraf grafu G, pro který $V' \subseteq V''$ a $E' \subseteq E''$, pak $V' = V''$ a $E' = E''$.

(b) Komponenta grafu G je jeho podgraf $G' = \langle V', E'\rangle$, který je indukovaný množinou vrcholů $V' \subseteq V$ takovou, že každé dva vrcholy z V' lze spojit cestou a že k V' není možné přidat další vrchol z V, aby to stále platilo.



Graf není souvislý (vrcholy x a y nejsou spojeny sledem).

Jeho podgraf indukovaný vrcholy u, v a w je souvislý, ale není to komponenta, protože není maximální souvislý.

Komponenty v tomto grafu jsou dvě. První je podgraf indukovaný vrcholy u, v, w, x, druhá je podgraf indukovaný vrcholem y.

I obecně platí, že komponenty tvoří „rozklad grafu“.

Věta Nechť $G_1 = \langle V_1, E_1 \rangle$, ..., $G_n = \langle V_n, E_n \rangle$ jsou všechny komponenty grafu $G = \langle V, E \rangle$. Pak každý vrchol $v \in V$ patří právě do jedné V_i a každá hrana $e \in E$ patří právě do jedné E_i.

Důkaz Vezměme vrchol $v \in V$. Podgraf indukovaný $\{v\}$ je zřejmě souvislý. Proto je podgrafem nějakého maximálního souvislého podgrafu grafu G, tj. komponenty G_i. Proto $v \in V_i$, tj. v patří aspoň do jedné z množin $V_1, \dots, V_n$.

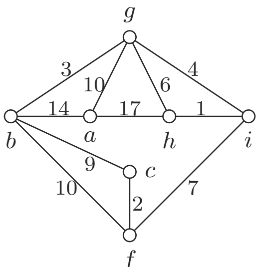
Dále se ukáže, že v nemůže patřit do různých $V_i \neq V_j$.

Že každá hrana $e \in E$ patří právě do jedné E_i dostaneme podobnou úvahou, když si uvědomíme, že pro $e = \{u, v\}$ je podgraf indukovaný množinou vrcholů $\{u, v\}$ je souvislý. □

Ohodnocené grafy

Definice Hranové ohodnocení grafu $\langle V, E\rangle$ s množinou hodnot D je funkce $w : E \to D$.
Vrcholové ohodnocení grafu $\langle V, E\rangle$ s množinou hodnot D je funkce $w : V \to D$.

Hranově ohodnocený graf, kde $w(\{a, b\}) = 14, w(\{a, g\}) = 10, \ldots, w(\{h, i\}) = 1$.



Délka sledu $v_0, e_1, \ldots, e_n, v_n$ v hranově ohodnoceném grafu je číslo

$$w(e_1) + \cdots + w(e_n).$$

Např. délka sledu $b, \{b, f\}, f, \{f, i\}, i, \{i, h\}, h$ v grafu na obrázku je 18.
Jako v neohodnoceném: vzdálenost z u do v = délka nejkratší cesty.

Hledání nejkratší cesty

Problém:

- vstup: neorientovaný graf $G = \langle V, E\rangle$, jeho hranové ohodnocení $w : E \to \mathbb{R}^+$ (každé hraně je přiřazena její délka), a vrchol $s \in V$.
- výstup: pro každý vrchol $v \in V$ číslo $d(v)$, které je vzdáleností z s do v.

Ukážeme tzv. Dijkstrův algoritmus.

Proměnné:

- $d(v)$ = délka nejkratší zatím nalezené cesty z s do v:
  - na začátku $d(s) = 0$ a $d(v) = \infty$ pro ostatní vrcholy $v \neq s$,
  - $d(v) = \infty$ znamená, že žádná cesta do v zatím nebyla nalezena,
  - hodnota $d(v)$ se v průběhu výpočtu zmenšuje; na konci délka nejkratší cesty z s do v, u vrcholů v, do kterých cesta z s nevede, zůstává $d(v) = \infty$.
- A = množina vrcholů, pro které zatím nebyla stanovena konečná hodnota $d(v)$:
  - na začátku $A = V$,
  - opakovaně následující krok: nové A = aktuální A - N, kde N = vrcholy z aktuální A s nejmenší hodnotou $d(v)$
- N = nově stanovené vrcholy, pro něž byla nalezena nejkratší cesta z s
  - vrcholy v z N jsou také považovány za vrcholy, přes které může do zbývajících vrcholů u z A vést kratší než dosud nalezená cesta,
  - možné zlepšení se ověří a změní se hodnoty $d(u)$ se aktualizují,
  - pokud existuje $v \in A$ s $d(v) \neq \infty$, pokračuje se výpočtem nových N, A a aktualizací $d(u)$ jako výše

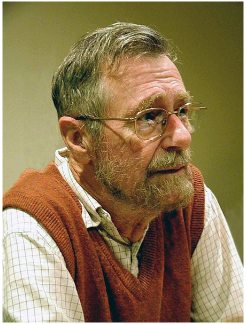
Dijsktrův algoritmus

Vstup: graf $G = \langle V, E\rangle$, vrchol $s \in V$,
hranové ohodnocení $w : E \to \mathbb{R}^+$,

Výstup: hodnota $d(v)$ pro každý $v \in V$, $d(v)$ je délka nejkratší cesty z s do v

Proměnné: funkce $d : V \to \mathbb{R}^+$, číslo $m \in \mathbb{R}^+$,
množiny $A, N \subseteq V$

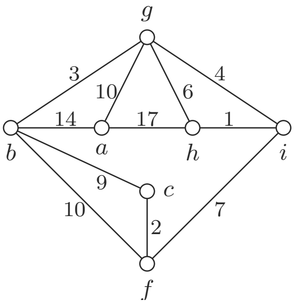
1. $d(s) := 0$; pro každý $v \in V - \{s\}$: $d(v) := \infty$; $A := V$;
2. pokud neexistuje $v \in A$ takový, že $d(v) \neq \infty$, skonči;
3. $m := \min\{d(v) \mid v \in A\}$; $N := \{v \in A \mid d(v) = m\}$; $A := A - N$;
4. pro všechny $v \in N$, $u \in A$ takové, že $\{v, u\} \in E$: jestliže $d(v) + e(\{v, u\}) < d(u)$, pak $d(u) := d(v) + w(\{v, u\})$; pokračuj krokem 2.



Edsger W. Dijkstra (1930–2002)

- nizozemský informatik a matematik
- průkopník v různých oblastech: algoritmy, programovací jazyky, operační systémy, ...
- Turingova cena 1972
- uvedený algoritmus 1956, publikován jako: Dijkstra, E. W. 1959. A note on two problems in connexion with graphs. Numerische Mathematik. 1: 269–271.

Příklad Běh Dijkstrova algoritmu pro následující graf a počáteční vrchol $s = h$.



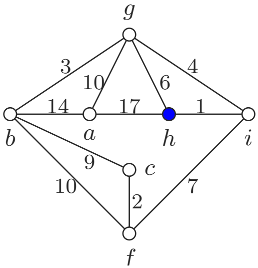
Krok 1:

A	a	b	c	f	g	h	i
$d(v)$	∞	∞	∞	∞	∞	0	∞

Krok 3: $m := 0$, $N := \{h\}$, $A := \{a, b, c, f, g, i\}$.
Krok 4:. $d(a) = d(h) + w(\{h, a\}) = 0 + 17 = 17$, $d(g) = d(h) + w(\{h, g\}) = 0 + 6 = 6$, $d(i) = d(h) + w(\{h, i\}) = 0 + 1 = 1$

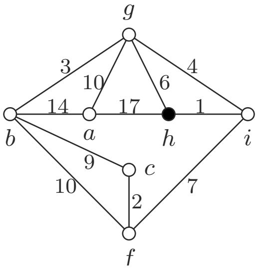
před uvedenými kroky

A	a	b	c	f	g	h	i
$d(v)$	∞	∞	∞	∞	∞	0	∞
✓							
$d(v)$							



po uvedených krocích

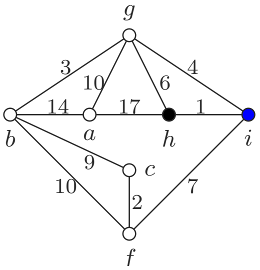
A	a	b	c	f	g		i
$d(v)$	17	∞	∞	∞	6		1
✓						$\boldsymbol{h}$	
$d(v)$						$\mathbf{0}$	



Krok 3: Nastaví se $m := 1$, $N := \{i\}$, $A := \{a, b, c, f, g\}$.
Krok 4: $d(f) = d(i) + w(\{i, f\}) = 1 + 7 = 8$, $d(g) = d(i) + w(\{i, g\}) = 1 + 4 = 5$.

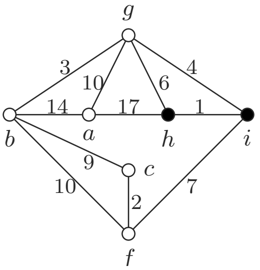
před uvedenými kroky

A	a	b	c	f	g		i
$d(v)$	17	∞	∞	∞	6		1
✓						$\boldsymbol{h}$	
$d(v)$						**0**	



po uvedených krocích

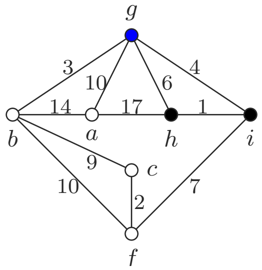
A	a	b	c	f	g		
$d(v)$	17	∞	∞	8	5		
✓						$\boldsymbol{h}$	$\boldsymbol{i}$
$d(v)$						**0**	**1**



Krok 3: $m := 5$, $N := \{g\}$, $A := \{a, b, c, f\}$.
Krok 4: $d(a) = d(g) + w(\{g, a\}) = 5 + 10 = 15$, $d(b) = d(g) + w(\{g, b\}) = 5 + 3 = 8$.

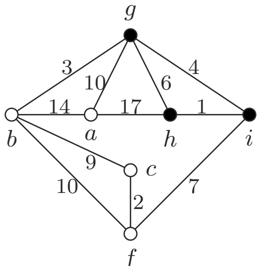
před uvedenými kroky

A	a	b	c	f	g
$d(v)$	17	∞	∞	8	5
✓					$\boldsymbol{h}$ $\boldsymbol{i}$
$d(v)$					**0** **1**



po uvedených krocích

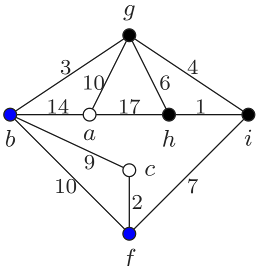
A	a	b	c	f
$d(v)$	15	8	∞	8
✓				$\boldsymbol{g}$ $\boldsymbol{h}$ $\boldsymbol{i}$
$d(v)$				**5** **0** **1**



Krok 3: Nastaví se $m := 8$, $N := \{b, f\}$, $A := \{a, c\}$.
Krok 4: Upraví se $d(c) = d(f) + w(\{f, c\}) = 10$.

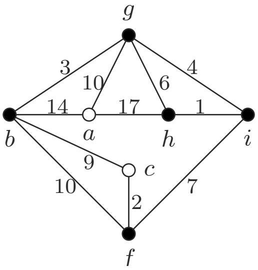
před uvedenými kroky

A	a	b	c	f			
$d(v)$	15	8	∞	8			
✓					$\boldsymbol{g}$	$\boldsymbol{h}$	$\boldsymbol{i}$
$d(v)$					**5**	**0**	**1**



po uvedených krocích

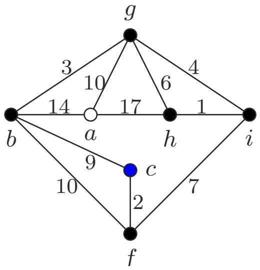
A	a		c				
$d(v)$	15		10				
✓		$\boldsymbol{b}$		$\boldsymbol{f}$	$\boldsymbol{g}$	$\boldsymbol{h}$	$\boldsymbol{i}$
$d(v)$		**8**		**8**	**5**	**0**	**1**



Krok 3: Nastaví se $m := 10$, $N := \{c\}$, $A := \{a\}$.
Krok 4: d se neupravuje.

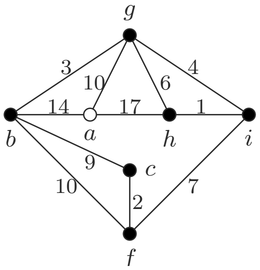
před uvedenými kroky

A	a		c				
$d(v)$	15		10				
✓		$\boldsymbol{b}$		$\boldsymbol{f}$	$\boldsymbol{g}$	$\boldsymbol{h}$	$\boldsymbol{i}$
$d(v)$		**8**		**8**	**5**	**0**	**1**



po uvedených krocích

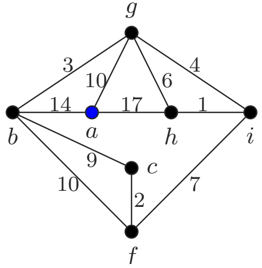
A	a						
$d(v)$	15						
✓		$\boldsymbol{b}$	$\boldsymbol{c}$	$\boldsymbol{f}$	$\boldsymbol{g}$	$\boldsymbol{h}$	$\boldsymbol{i}$
$d(v)$		**8**	**10**	**8**	**5**	**0**	**1**



Krok 3: Nastaví se $m := 15$, $N := \{a\}$, $A := \emptyset$.
Krok 4: d se neupravuje.

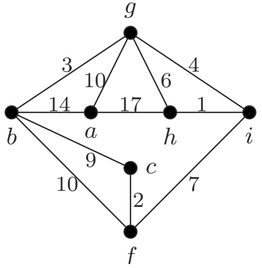
před uvedenými kroky

A	a						
$d(v)$	15						
✓		$\boldsymbol{b}$	$\boldsymbol{c}$	$\boldsymbol{f}$	$\boldsymbol{g}$	$\boldsymbol{h}$	$\boldsymbol{i}$
$d(v)$		**8**	**10**	**8**	**5**	**0**	**1**



po uvedených krocích

A							
$d(v)$							
✓	$\boldsymbol{a}$	$\boldsymbol{b}$	$\boldsymbol{c}$	$\boldsymbol{f}$	$\boldsymbol{g}$	$\boldsymbol{h}$	$\boldsymbol{i}$
$d(v)$	**15**	**8**	**10**	**8**	**5**	**0**	**1**



Krok 2: neexistuje $v \in A$ s $d(v) \neq \infty$, algoritmus tedy skončí.

Poslední tabulka udává vzdálenosti k vrcholům grafu z vrcholu h, tedy:

A $d(v)$	
✓ $d(v)$	$\boldsymbol{a}$ $\boldsymbol{b}$ $\boldsymbol{c}$ $\boldsymbol{f}$ $\boldsymbol{g}$ $\boldsymbol{h}$ $\boldsymbol{i}$ **15 8 10 8 5 0 1**

Kudy vede nejkratší cesta?

- Z výsledku není vidět, kudy vedou k vrcholům u nejkratší cesty z s.
- Ty zjistíme snadnou úpravou algoritmu:
  - ke každému vrcholu u udržujeme množinu vrcholů $\mathrm{pred}(u)$, které se na některé z nejkratších cest z s do u nachází těsně před u,
  - na začátku $\mathrm{pred}(u) := \emptyset$
  - určí-li se v kroku 4 nová hodnota hodnota $d(u)$, provedeme

$$\mathrm{pred}(u) := \{v \in N \mid d(u) = d(v) + w(\{v, u\})\}\,.$$

- Příklad: Pro $s = h$ je po skončení:

$$\mathrm{pred}(c) = \{f\},\ \mathrm{pred}(f) = \{i\},\ \mathrm{pred}(i) = \{h\},\ \mathrm{pred}(h) = \emptyset.$$

tedy (v tomto případě jediná) nejkratší cesta do c je

$$h, \{h, i\}, i, \{i, f\}, f, \{f, c\}, c$$

Ručním ověřením zjistíme, že výše vypočítané vzdálenosti jsou správné. Je to tak vždy?

Věta Dijkstrův algoritmus správný, tedy pro každý graf G s ohodnocením $w : E \to \mathbb{R}^+$ a vrchol $s \in V$ je po skončení výpočtu pro každý vrchol v hodnota $d(v)$ délka nejkratší cesty z s do v.

Důkaz správnosti Dijkstrova algoritmu
Přednáška; podrobně viz [DS1].

Časová složitost Dijkstrova algoritmu v nejhorším případě

Lze ukázat, že pro $n = |V|$ a $m = |E|$ pro tuto časovou složitost $T(n,m)$ je

$$T(n,m) = O(n \log n + m).$$

Později.

Stupně vrcholů

Dále jen neorientované grafy.

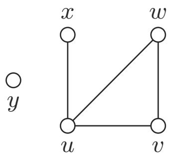
Definice Stupeň vrcholu $v \in V$ grafu $\langle V, E\rangle$ je počet hran, pro které je v koncovým vrcholem, a značí se $\deg(v)$.

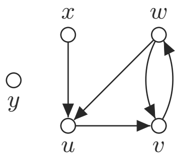
- Pro orientované: vstupní $\deg_{\mathrm{i}}(v)$ a výstupní $\deg_{\mathrm{o}}(v)$, $\deg(v) = \deg_{\mathrm{i}}(v) + \deg_{\mathrm{o}}(v)$.
- $\deg(v)$ snadno vidět.
- Pro graf níže vlevo je

$$\deg(u) = 3,\ \deg(v) = 2,\ \deg(w) = 2,\ \deg(x) = 1,\ \deg(y) = 0.$$

  Pro graf vpravo je

$$\deg(u) = 3,\ \deg(v) = 3,\ \deg(w) = 3,\ \deg(x) = 1,\ \deg(y) = 0.$$





Věta V grafu $G = \langle V, E \rangle$ je $\sum_{v \in V} \deg(v) = 2|E|$.

Důkaz Každá hrana $e \in E$ má dva vrcholy, u a v.
Hrana e přispívá jedničkou do $\deg(u)$ (je jednou z hran, jejichž počet je roven $\deg(u)$), jedničkou do $\deg(v)$ a do stupně žádného jiného vrcholu nepřispívá.
Hrana e tedy přispívá právě počtem 2 do $\sum_{v \in V} \deg(v)$. To platí pro každou hranu. Proto $\sum_{v \in V} \deg(v) = 2|E|$. $\square$

Věta Počet vrcholů lichého stupně je v libovolném grafu sudý.

Důkaz $\sum_{v \in V} \deg(v) = \sum_{v \in S} \deg(v) + \sum_{v \in L} \deg(v)$,
kde S a L jsou množiny vrcholů, které mají sudý a lichý stupeň.

Je jasné, že $\sum_{v \in S} \deg(v)$ je sudé číslo.
Dle věty výše je $\sum_{v \in V} \deg(v)$ sudé. Proto i $\sum_{v \in L} \deg(v)$ musí být sudé.
Kdyby byl počet vrcholů s lichým stupněm lichý, byl by $\sum_{v \in L} \deg(v)$ liché číslo, což není možné. $\square$

Skóre grafu * (nepovinné)

Definice Pro graf $G = \langle V, E\rangle$ s vrcholy $v_1, \dots, v_n$ (v tomto pořadí) se posloupnost

$$\deg(v_1), \dots, \deg(v_n)$$

nazývá skóre grafu G.

Jakou nese o grafu informaci? Přitom dvě skóre považujeme za stejná, liší-li se jen pořadím prvků.

- Dva izomorfní grafy mají stejná skóre (snadno se vidí).
- $1, 1, 1, 1, 1, 1$ je skóre grafu; např. $\langle V, E\rangle$, kde $V = \{a, b, c, d, e, f\}$ a $E = \{\{a, b\}, \{c, d\}, \{e, f\}\}$;
  každé dva grafy s tímto skóre jsou izomorfní.
- Skóre $2, 2, 2, 2, 2, 2$ však graf jednoznačně (až na izomorfismus) neurčuje:
  graf 1: $E_1 = \{\{a, b\}, \{b, c\}, \{a, c\}, \{d, e\}, \{e, f\}, \{d, f\}\}$ (dva trojúhelníky),
  graf 2: $E_2 = \{\{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}, \{e, f\}, \{a, f\}\}$ (šestiúhelník).

Ne každá posloupnost čísel je skóre nějakého grafu.
Např. $3, 4, 2, 0$ není skóre grafu (jeden lichý stupeň).
$6, 2, 2, 0$ nemá lichý stupeň, ale také to není skóre.

Jednoduché kritérium:

Věta Nechť $d_1 \geq d_2 \geq d_3 \geq \cdots \geq d_n$ jsou nezáporná celá čísla a $1 \leq d_1 \leq n - 1$. Pak

$$d_1, d_2, d_3, \ldots, d_n$$

je skóre nějakého grafu, právě když

$$d_2 - 1, d_3 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n$$

je skóre nějakého grafu.

Důkaz Přednáška; detaily v [DS1].

Algoritmus testující, zda posloupnost je skóre grafu:

Vstup: $n \in \mathbb{N}$, $\langle d_1, \dots, d_n \rangle$, kde $d_1 \geq \cdots \geq d_n \geq 0$ jsou celá čísla;
Výstup: ANO, pokud $\langle d_1, \dots, d_n \rangle$ je skóre, NE v opačném případě;

1. je-li $n = 1$ a $d_1 = 0$, odpověz ANO a skonči;
2. je-li $d_1 > n - 1$, odpověz NE a skonči;
3. vypočítej novou posloupnost
   $\langle d'_1, \dots, d'_{n-1} \rangle = \langle d_2 - 1, d_3 - 1, \dots, d_{d_1+1} - 1, d_{d_1+2}, \dots, d_n \rangle$;
4. je-li některé d'_i záporné, odpověz NE a skonči;
5. uspořádej $d'_1, \dots, d'_{n-1}$ sestupně, přiřaď takto uspořádané hodnoty do $d_1, \dots, d_{n-1}$, sniž n o 1 (tj. $n := n - 1$) a pokračuj bodem 1.

Příklad Je $6, 6, 5, 4, 3, 3, 3, 3, 2, 1, 0$ skóre nějakého grafu?
Algoritmus postupně generuje:

$$6, 6, 5, 4, 3, 3, 3, 3, 2, 1, 0 \longrightarrow 5, 4, 3, 3, 2, 2, 2, 2, 1, 0 \longrightarrow 3, 2, 2, 2, 2, 1, 1, 1, 0$$
$$\longrightarrow 2, 1, 1, 1, 1, 1, 1, 0 \longrightarrow 1, 1, 1, 1, 0, 0, 0 \longrightarrow 1, 1, 0, 0, 0, 0$$
$$\longrightarrow 0, 0, 0, 0, 0 \longrightarrow 0, 0, 0, 0 \longrightarrow 0, 0, 0 \longrightarrow 0, 0 \longrightarrow 0$$

Odpověď je tedy ANO.

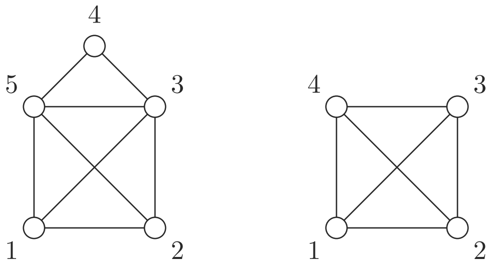
Příklad Je $4, 3, 2, 1, 1$ skóre nějakého grafu?
Algoritmus postupně generuje:

$$4, 3, 2, 1, 1 \longrightarrow 2, 1, 0, 0 \longrightarrow 1, -1, 0$$

Odpověď je tedy NE.
(Že $4, 3, 2, 1, 1$ není skóre, můžeme poznat také přímo podle toho, že $4, 3, 2, 1, 1$ má lichý počet lichých stupňů.)

Eulerovské grafy (jednotažky) * (nepovinné)

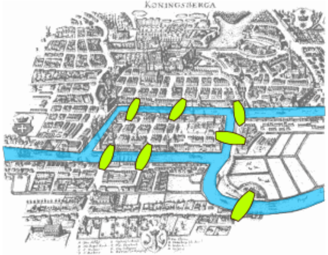
Nakreslete obrázek jedním tahem s tím, že žádnou hranu nesmíte nakreslit dvakrát a nesmíte zvednout tužku z papíru.



U levého obrázku to jde, u pravého ne (zdůvodněte).

Definice Eulerovský tah je tah, který obsahuje všechny vrcholy grafu a ve kterém se každá hrana vyskytuje právě jednou. Je-li navíc uzavřený, nazývá se uzavřený eulerovský tah (nebo eulerovská kružnice).

Zřejmě: Eulerovský tah představuje kreslení „jedním tahem“. Chceme-li navíc při kreslení vyjít i skončit v jednom místě, musíme najít uzavřený eulerovský tah.



Leonhard Euler (1707–1783) a sedm mostů v Královci

- jeden z nejvýznamnějších matematiků
- 1735 problém 7 mostů v Královci (Prusko, dnes Kaliningrad, Rusko): Lze vyjít z jednoho místa, projít před každý most právě jednou a skončit ve stejném místě?
- další známý výsledek v teorii grafů: $v - e + f = 2$ (počty vrcholů, hran a stěn konvexního mnohostěnu)
- další významné objevy (mat. analýza, Eulerovo číslo e, ...).

Věta

(a) V neorientovaném grafu existuje uzavřený eulerovský tah, právě když je souvislý a každý vrchol má sudý stupeň.

(b) V neorientovaném grafu existuje neuzavřený eulerovský tah, právě když je souvislý a má právě dva vrcholy lichého stupně.

Důkaz Přednáška a [DS1].

Příklad Použijte větu ke zjištění, zda lze výše uvedené grafy nakreslit jedním tahem.

Eulerovská kružnice vs hamiltonovská kružnice

Problém rozhodnout, zda v grafu existuje eulerovská kružnice, je snadný. Dle věty stačí zjistit, zda každý vrchol má sudý stupeň.

Podobný je problém rozhodnout, zda v grafu existuje hamiltonovské kružnice.
Hamiltonovská kružnice = kružnice, která obsahuje všechny vrcholy grafu.

Není znám rychlý algoritmus, který by zjistil, zda graf má hamiltonovskou kružnici.
Navíc je pravděpodobné, že takový algoritmus ani neexistuje (zjistit existenci hamiltonovské kružnice je totiž tzv. NP-úplný problém).

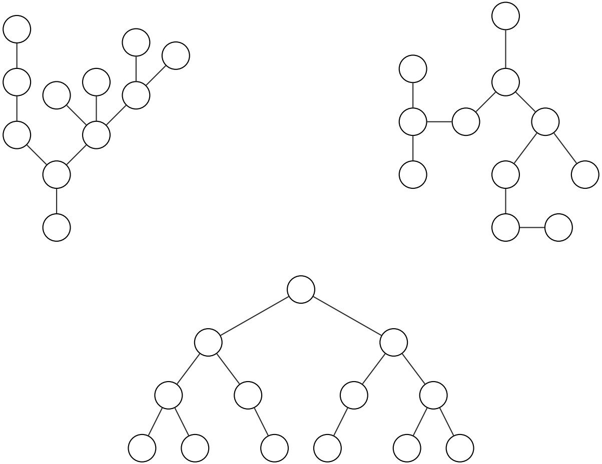
William R. Hamilton (1805–1865), irský matematik.

Stromy

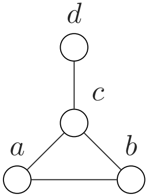
- grafy připomínající vzhledem stromy: větvení
- široké použití
  - různá členění, katalogy, adresáře v počítači apod. mají strukturu stromu
  - zásadní význam pro ukládání a vyhledávání dat

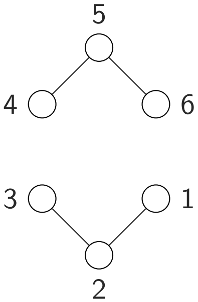
Definice Strom je neorientovaný souvislý graf bez kružnic.

Následující grafy jsou stromy:



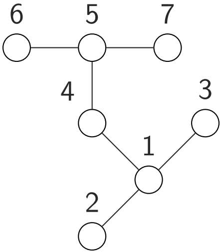
Následující grafy nejsou stromy:





Definice Vrchol se stupněm 1 se nazývá koncový. Koncový vrchol stromu se nazývá list.

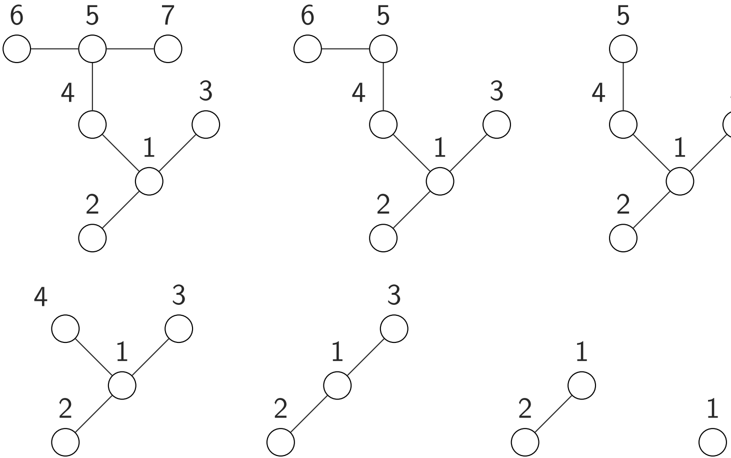
Uvažujme strom



Listy tohoto stromu jsou vrcholy 2, 3, 6 a 7.

Odebereme-li z tohoto stromu nějaký list a hranu, která do něj vede, vznikne opět strom.

Z něho můžeme opět odebrat nějaký list s hranou atd., až z tohoto stromu zbyde jediný uzel. Popsaný proces je vidět zde:



Lze i obráceně: Vyjít z jednoho vrcholu a postupně k němu připojovat vrcholy (listy).

Zmiňovanou vlastnost mají všechny stromy. $\longrightarrow$ konstruktivní pohled

Vlastnost dokážeme. Nejdřív pomocné tvrzení.

Věta V každém stromu s alespoň dvěma vrcholy existují aspoň dva listy.

Důkaz Uvažujme cestu $v_0, e_1, \dots, e_n, v_n$, která má ze všech cest největší délku. Tvrdíme, že v_0 i v_n jsou listy. Dokážeme to sporem.

Kdyby v_0 nebyl list, existovala by hrana $e = \{v, v_0\}$, která je různá od e_1 (v_0 není list). Vrchol v musí být různý od každého v_i ($i = 1, \dots, n$). Kdyby totiž $v = v_i$ pro nějaké $i = 1, \dots, n$, pak by $v, e, v_0, \dots, v_i$ byla kružnice, což není možné, protože náš graf je strom. Pak je ale posloupnost $v, e, v_0, e_1, \dots, v_n$ cesta, která je delší než $v_0, e_1, \dots, v_n$, což je spor s předpokladem, že $v_0, e_1, \dots, v_n$ je ze všech cest nejdelší. Tedy v_0 je list.

Podobně se ukáže, že v_n je list. $\square$

Nyní hlavní tvrzení:

Věta Pro graf G a jeho koncový vrchol v jsou následující tvrzení ekvivalentní.

1. G je strom.
2. $G - v$ je strom. ($G - v$ vznikne odstraněním v a hrany z v.)

Důkaz „1.⇒2.“: Máme ukázat, že $G - v$ je souvislý a neobsahuje kružnici.

Souvislost: Vezměme libovolné dva vrcholy $u \neq w$ grafu $G - v$. Protože G je souvislý, existuje v G cesta $u, e, \dots, w$. Protože v je koncový vrchol grafu G, musí být každý vrchol této cesty různý od v. Cesta $u, e, \dots, w$ je tedy cestou v grafu $G - v$. Proto je G souvislý.

Kružnice: Kdyby $G - v$ obsahoval kružnici, byla by to zřejmě i kružnice v grafu G.

„2.⇒1.“: Předpokládejme naopak, že $G - v$ je strom. Ukážeme, že G je strom.

Souvislost: Vezměme vrcholy $u \neq w$ v G. Když $u \neq v \neq w$, pak protože $G - v$ je strom, existuje v $G - v$ cesta z u do w. Ta je zřejmě i cestou v G. Když je $u = v$ (pro $w = v$ je to podobné), uvažujme vrchol x, ke kterému byl v v grafu G připojen (tj. $G - v$ vznikl odebráním vrcholu v a hrany $\{v, x\}$). Ze souvislosti $G - v$ plyne, že v něm existuje cesta $x, \ldots, w$. Je zřejmé, že pak je $u = v, \{v, x\}, x \ldots, w$ cesta v G, která spojuje u a w. G je tedy souvislý.

Kružnice: Pokud by $v_0, \ldots, v_n$ byla kružnice v G, pak by neobsahovala v, protože v je koncový vrchol (vrcholy kružnice mají stupeň aspoň 2). Tato kružnice by tedy byla i kružnicí v $G - v$, což není možné, protože předpokládáme, že $G - v$ je strom.

Důležité charakterizace stromů:

Věta Pro neorientovaný graf $G = \langle V, E \rangle$ jsou následující tvrzení ekvivalentní.

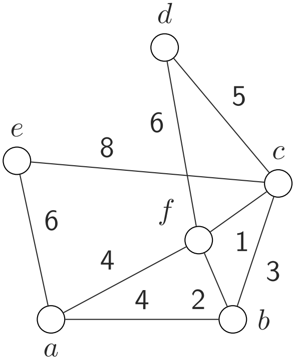
1. G je strom.
2. Mezi každými dvěma vrcholy grafu G existuje právě jedna cesta.
3. G je souvislý, ale vynecháním libovolné hrany vznikne nesouvislý graf.
4. G neobsahuje kružnice, ale přidáním jakékoli hrany vznikne graf s kružnicí.
5. G neobsahuje kružnice a $|V| = |E| + 1$.
6. G je souvislý a $|V| = |E| + 1$.

Důkaz Přednáška; podrobně viz [DS1].

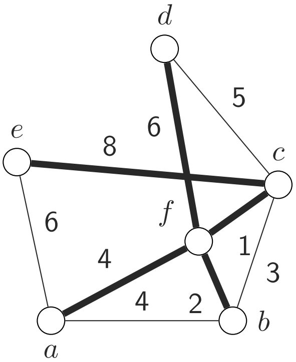
Minimální kostra grafu

Úloha: Propojit města $v_1, \ldots, v_n$ elektrickým vedením co nejlevněji.
Tedy rozhodnout, mezi kterými městy máme natáhnout elektrické dráty.
Mezi některými dvojicemi měst přímé propojení postavit nelze (hory, přehrady apod.).

Možné zadání (čísla udávají náklady w_{xy} na výstavbu propojení x a y):



Možné řešení:



Celkové náklady jsou

$$w_{af} + w_{bf} + w_{ce} + w_{cf} + w_{df} = 4 + 2 + 8 + 1 + 6 = 21.$$

Jak uvidíme, řešení je zbytečně nákladné. Jak najít nejlepší řešení?

V řeči grafů:

- Dán souvislý neorientovaný graf $G = \langle V, E \rangle$ a hranové ohodnocení $w : E \to \mathbb{R}^+$.
- Chceme vybrat podgraf, který obsahuje všechny vrcholy a je souvislý, tedy podgraf $G' = \langle V, E' \rangle$.
- $G' = \langle V, E' \rangle$ nesmí obsahovat kružnice (odstranění hrany by podgraf zůstal souvislý, ale byl by levnější).
- Součet délek hran z E' má být nejmenší možný.
- Chceme vybrat tzv. minimální kostru grafu G.

Definice Kostra neorientovaného grafu G je jeho podgraf G', který je stromem a obsahuje všechny vrcholy grafu G.
Je-li $w : E \to \mathbb{R}^+$ hranové ohodnocení grafu $G = \langle V, E\rangle$, nazývá se kostra $G' = \langle V, E'\rangle$ minimální kostra, pokud má ze všech koster grafu G nejmenší hodnotu $w(G')$, kde

$$w(G') = \sum_{\{u,v\}\in E'} w(\{u, v\}).$$

- Tučně vyznačený podgraf výše je tedy kostrou, ale ne minimální kostrou. (Najděte kostru s menší hodnotou.)
- Snadno se vidí, že graf má kostru, právě když je souvislý.
- V grafu může existovat více koster s nejmenší hodnotou.

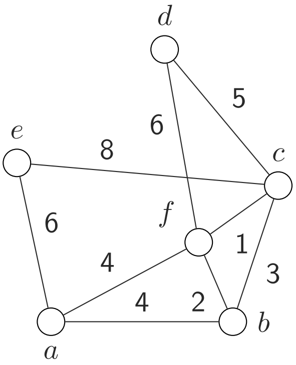
Kruskalův algoritmus hledání minimální kostry

Vstup: souvislý neorientovaný graf $G = \langle V, E\rangle$ s n vrcholy a m hranami, ohodnocení $w : E \to \mathbb{R}^+$

Výstup: množina hran $E' \subseteq E$ taková, že $G' = \langle V, E'\rangle$ je minimální kostra grafu G

1. setřiď hrany vzestupně podle ohodnocení, tj. utvoř posloupnost $e_1, \ldots, e_m$ všech hran z E tak, že
$$w(e_1) \leq \cdots \leq w(e_m);$$
2. $E_0 = \emptyset$; $i := 0$;
3. dokud E_i neobsahuje $n-1$ hran, prováděj:
$$i := i + 1;$$
$$E_i := \begin{cases} E_{i-1} \cup \{e_i\} & \text{pokud } \langle V, E_{i-1} \cup \{e_i\}\rangle \\ & \text{neobsahuje kružnici,} \\ E_{i-1} & \text{v opačném případě;} \end{cases}$$
4. $E' := E_i$.

Příklad Dán ohodnocený graf:



Krok 1: Setřídění hran

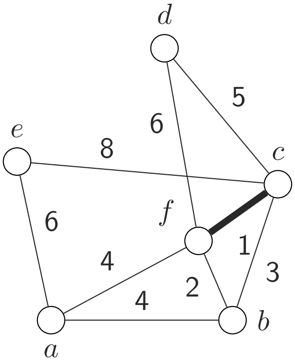
$$e_1, e_2, \ldots, e_9 = \{c, f\}, \{b, f\}, \{b, c\}, \{a, b\}, \{a, f\}, \{c, d\}, \{a, e\}, \{d, f\}, \{c, e\}.$$

(Správně, odpovídající posloupnost hodnot $w(\{u, v\})$ je $1, 2, 3, 4, 4, 5, 6, 6, 8.$)

Krok 2: Algoritmus nastaví $E_0 = \emptyset$ a $i = 0$.

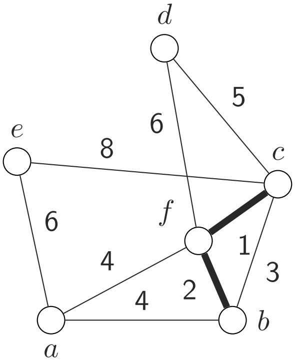
Krok 3:

$$\boldsymbol{\{c, f\}}, \{b, f\}, \{b, c\}, \{a, b\}, \{a, f\}, \{c, d\}, \{a, e\}, \{d, f\}, \{c, e\}$$



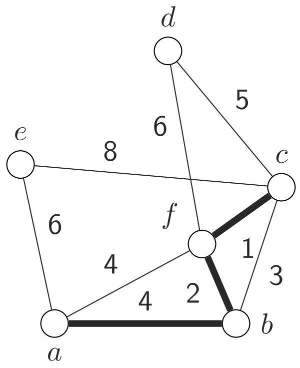
Krok 3:

$$\boldsymbol{\{c,f\}}, \boldsymbol{\{b,f\}}, \{b,c\}, \{a,b\}, \{a,f\}, \{c,d\}, \{a,e\}, \{d,f\}, \{c,e\}$$



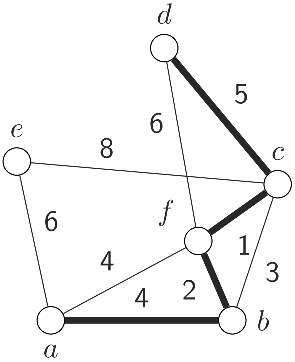
Krok 3:

$$\boldsymbol{\{c,f\}}, \boldsymbol{\{b,f\}}, {\color{red}\{b,c\},}\boldsymbol{\{a,b\}}, \{a,f\}, \{c,d\}, \{a,e\}, \{d,f\}, \{c,e\}$$



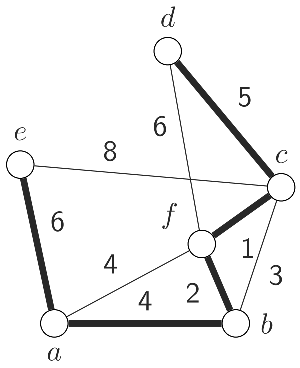
Krok 3:

$$\boldsymbol{\{c,f\}}, \boldsymbol{\{b,f\}}, \{b,c\}, \boldsymbol{\{a,b\}}, \{a,f\}, \boldsymbol{\{c,d\}}, \{a,e\}, \{d,f\}, \{c,e\}$$



Krok 3:

$$\boldsymbol{\{c, f\}}, \boldsymbol{\{b, f\}}, \{b, c\}, \boldsymbol{\{a, b\}}, \{a, f\}, \boldsymbol{\{c, d\}}, \boldsymbol{\{a, e\}}, \{d, f\}, \{c, e\}$$



Krok 3: vybráno $5 = n - 1$ hran.

Krok 4: $E' = \{\{c, f\}, \{b, f\}, \{a, b\}, \{c, d\}, \{a, e\}\}$, hodnota kostry $= 18$

Jak v kroku 4 zjistit, zda $\langle V, E_{i-1} \cup \{e_i\}\rangle$ obsahuje kružnici? Takto:

Graf $\langle V, E_{i-1}\rangle$ se rozpadá na komponenty.

Množiny vrcholů $V_1, \ldots, V_k$ těchto podgrafů tvoří rozklad množiny V.

Protože $\langle V, E_{i-1}\rangle$ kružnice neobsahuje, neobsahuje kružnice ani žádná z jeho komponent. Každá jeho komponenta je tedy stromem.

Dle uvedené věty přidáním hrany e_i vznikne kružnice, právě když oba vrcholy hrany e_i leží v jedné z množin V_l.

Takový test je zřejmě možné provést poměrně rychle pomoc datových struktur pro uchování systému disjunktních množin.

Časová složitost Kruskalova algoritmu v nejhorším případě

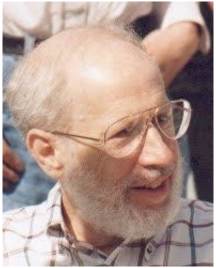
Lze ukázat, že pro $n = |V|$ a $m = |E|$ pro tuto časovou složitost $T(n,m)$ je

$$T(n,m) = O(m \log n).$$

Později.
(V tomto případě lze také ukázat, že $T(n,m) = O(m \log m)$.)

Důkaz správnosti Kruskalova algoritmu

Přednáška; podrobně viz [DS1]



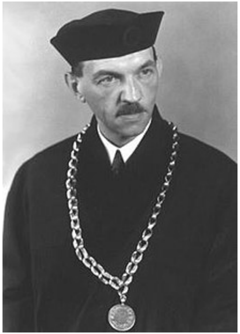
Joseph Kruskal (1928–2010)

- americký matematik a informatik,
- J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem. Proc. Amer. Math. Soc. 7 (1)(1956): 48–50.

Dřívější (jiné) algoritmy:

- O. Borůvka, O jistém problému minimálním. 1926
  O. Borůvka, Příspěvek k řešení otázky ekonomické stavby elektrovodních sítí. 1926
- V. Jarník, O jistém problému minimálním. 1930
- Jarníkův algoritmus znovuobjeven:
  R. C. Prim, Shortest connection networks and some generalizations. 1957
  E. W. Dijkstra, A note on two problems in connexion with graphs. 1959

Otakar Borůvka (1899–1995)



Vojtěch Jarník (1897–1970)

Kořenové stromy

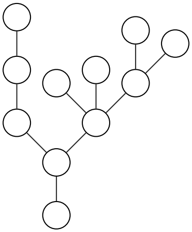
= stromy, ve kterých je jeden vrchol označen za kořen (určuje „orientaci stromu“)

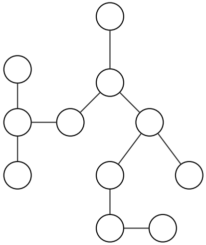
Definice Kořenový strom je dvojice $\langle G, r\rangle$, kde $G = \langle V, E\rangle$ je strom a $r \in V$ je vrchol, tzv. kořen.

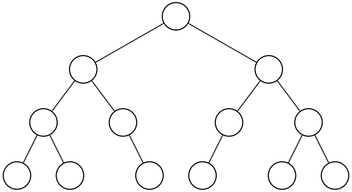
Nechť $\langle G, r\rangle$ je kořenový strom.

- Vrchol v se nazývá potomek vrcholu u (u se nazývá předek vrcholu v), právě když cesta z kořene r do v má tvar $r, \dots, u, \dots, v$.
- Vrchol v se nazývá následník (někdy přímý potomek) vrcholu u (u se nazývá předchůdce (někdy rodič) vrcholu v), právě když cesta z kořene r do v má tvar $r, \dots, u, e, v$.
- Hloubka (někdy úroveň) vrcholu v je délka cesty od kořene r do v.
- Výška vrcholu v je délka nejdelší cesty neobsahující předky v, která vede z v do některého z listů.
- Výška (někdy hloubka) stromu je výška jeho kořene (ekvivalentně: je největší z hloubek jeho listů).

Příklad U následujících stromů zvolte kořen, určete úroveň vrcholů, hloubku stromu.
Přednáška; podrobnosti [DS1].







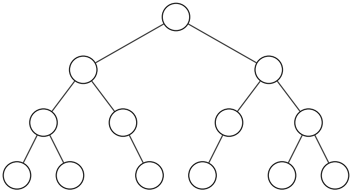
- Vrchol může mít více následovníků, ale nejvýše jednoho rodiče.
- Výška stromu je délka nejdelší cesty, která začíná v kořeni.
- Kořenové stromy se kreslí „od kořene dolů“ a po úrovních.
- Podstrom indukovaný vrcholem v je podgraf indukovaný množinou vrcholů u, ke kterým vede z kořene r cesta obsahující v (tedy množinou sestávající z potomků vrcholu v).

Definice Kořenový strom se nazývá

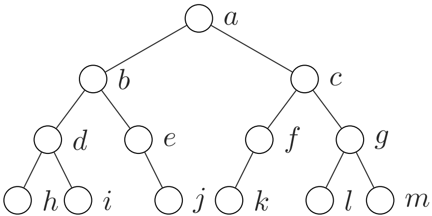
- m-ární, právě když každý jeho vrchol má nejvýše m potomků;

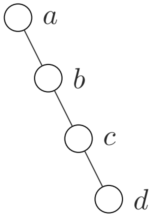
  2-ární strom se nazývá binární, 3-ární pak ternární.
- zaplněný pokud splňuje následující podmínky:
  - každý vrchol s výjimkou vrcholů s hloubkou $h-1$ má 0 nebo m následníků;
  - každý list má hloubku h nebo $h-1$.

Zvolíme-li nejvýše nakreslený vrchol kořenem, je následující strom binárním kořenovým stromem.



Následující stromy jsou binární kořenové stromy (kořeny jsou nejvýše nakreslené vrcholy). Strom vlevo je zaplněný, strom vpravo ne.





Vztahy mezi výškou, počtem vrcholů a počtem listů stromu

Otázky o kořenových stromech (binárních, m-árních):

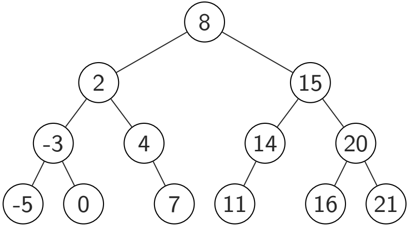
- Strom má výšku h. Kolik může mít listů? Kolik může mít vrcholů?
- Strom má l listů. Jakou má výšku?
- Strom má n vrcholů. Jakou má výšku?

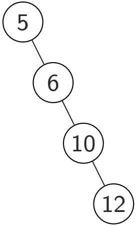
Proč jsou otázky významné?

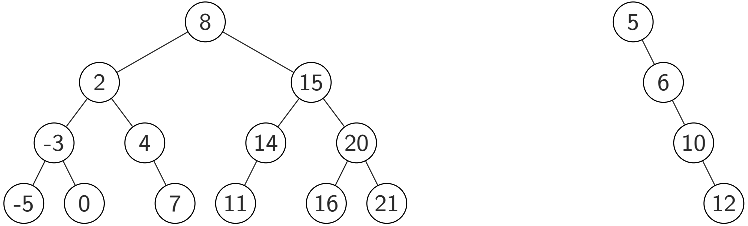
- Vyhledávací stromy a další stromové struktury pro ukládání dat.
- Analýza složitosti různých algoritmů (např. rekurzívních).

Příklad (binární vyhledávací stromy)

- binární kořenové stromy, ve kterých
  - následovníci každého vrcholu mají dáno pořadí: levý (první), pravý (druhý);
  - v každém vrcholu v je „uložena“ číselná hodnota $w(v) \in \mathbb{R}$
  - je-li u levým potomkem vrcholu v, pak $h(u) < h(v)$,
    je-li u pravým potomkem vrcholu v, pak $h(u) > h(v)$.
- příklad: přednáška a [DS1]







- jak zjistit, zda se hodnota w ve stromu nachází?: přednáška a [DS1]
- kolik kroků je třeba ke zjištění ve stromu s n vrcholy provést?: v nejhorším případě „zhruba h"
- není-li strom nijak vyvážený, může být $h = n - 1$ (degenerovaný strom = seznam uložených hodnot);
- ve vyváženém stromě (např. dle výše uvedené definice) je ale $h =$ „zhruba" $\log_2 n$;
- tedy k vyhledání čísla ve stromu, který obsahuje 1000 čísel a který je vyvážený, stačí zhruba $\log_2 1024 = 10$ kroků.

Věta V zaplněném binárním kořenovém stromu s n vrcholy a l listy, který má výšku h, platí

(a) $2^h \leq n \leq 2^{h+1} - 1$;

(b) $h = \lfloor \log_2 n \rfloor$;

(c) $2^{h-1} \leq l \leq 2^h$;

(d) $\lceil \log_2 l \rceil \leq h \leq \lfloor \log_2 l \rfloor + 1$.

Důkaz Přednáška; podrobně viz [DS1]

Věta V libovolném binárním kořenovém stromu s n vrcholy a l listy, který má výšku h, platí

(a) $h + 1 \leq n \leq 2^{h+1} - 1$;

(b) $\lfloor \log_2 n \rfloor \leq h \leq n - 1$;

(c) $1 \leq l \leq 2^h$;

(d) $\lceil \log_2 l \rceil \leq h$.

Důkaz Viz [DS1]