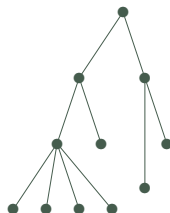
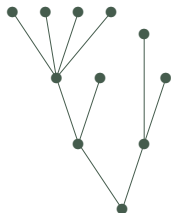


10 Stromy a kostry grafů

Na rozdíl od předchozí lekce, která se zabývala grafy z obecného a také trochu povrchního pohledu, se nyní soustředíme na jednu konkrétní nepříliš obtížnou oblast, které se budeme věnovat do větší hloubky. Jde o problematiku stromů, neboli acyklických souvislých grafů, představujících nejjednodušší podobu grafů.

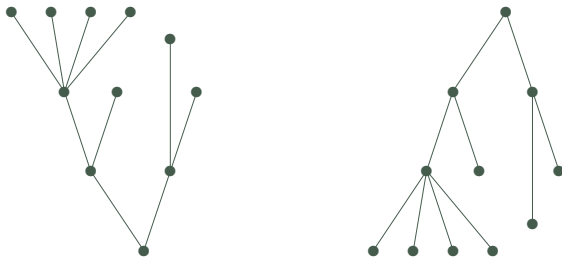


Stručný přehled lekce

- * Stromy a jejich vlastnosti s důkazy.
- * Minimální kostra grafu a její základní algoritmy.
- * Dodatek: systémy různých reprezentantů.

10.1 Stromy – grafy bez kružnic

Začněme ilustračními obrázky stromů. Povšimněte si přitom jedné zvláštnosti, totiž že v informatice stromy typicky rostou „shora dolů“ . . .



Charakteristickými znaky stromů je absence kružnic a souvislost. □

Definice 10.1. **Strom** je (jednoduchý) souvislý graf T bez kružnic.

□

Obecněji **les** je pak graf bez kružnic (jednoduchý, nemusí být souvislý). Komponenty souvislosti lesa jsou stromy. Jeden vrchol bez hran a prázdný graf jsou také stromy. Grafy bez kružnic také obecně nazýváme **acyklické**.

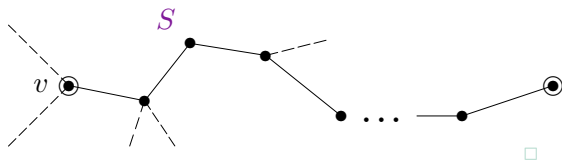
Vlastnosti stromů

Tvrzení 10.2. *Strom s více než jedním vrcholem obsahuje vrchol **stupně 1**.* □

Důkaz: Vezmeme libovolný strom T a v něm libovolný vrchol v . Jelikož souvislý graf s více než jedním vrcholem nemá vrchol stupně 0, vrch. v má incidentní hranu. □ Zvolme (sestrojme) nyní nejdelší možnou cestu S v T začínající ve v :

- * S začne libovolnou hranou vycházející z v ; □
- * v každém dalším vrcholu u cesty S , který má stupeň větší než 1, pokračuje S další hranou. □

Pokud by tomu tak nebylo, není S nejdelší možná nebo další hrana vede do některého předchozího vrcholu cesty S . Tím bychom ale získali **kružnici**.



Proto cesta S nutně skončí v nějakém vrcholu stupně 1 v T . □

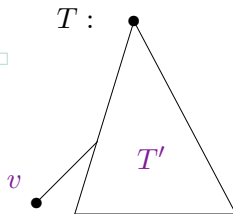
Definice: Každý jeho vrchol stromu stupně 1 nazveme **listem** stromu.

Počet hran stromu

Věta 10.3. *Strom na n vrcholech má přesně $n - 1$ hran pro $n \geq 1$.* \square

Důkaz: Toto tvrzení dokážeme indukcí podle n .

- * Strom s jedním vrcholem má přesně $0 = n - 1$ hran. \square



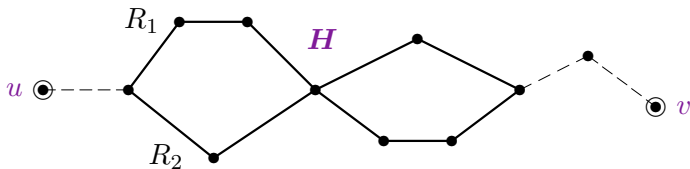
- * Předpokládejme platnost tvrzení pro libovolné přirozené $n := i \geq 1$. Necht' T je nyní libovolný strom na $n := i + 1$ vrcholech. \square

Podle Tvrzení 10.2 obsahuje T vrchol v stupně 1. Označme $T' = T - v$ graf vzniklý z T odebráním vrcholu v a jedné jeho hrany. \square

Pak T' je také souvislý graf bez kružnic, a tudíž strom na $n - 1 = i$ vrcholech. Dle indukčního předpokladu T' má $i - 1 = n - 2$ hran, a proto T má $n - 2 + 1 = n - 1$ hran. \square

Cesty ve stromech

Věta 10.5. Mezi každými dvěma vrcholy stromu vede právě *jediná cesta*. \square



Důkaz: Jelikož strom T je souvislý dle definice, mezi libovolnými dvěma vrcholy u, v vede nějaká cesta. \square

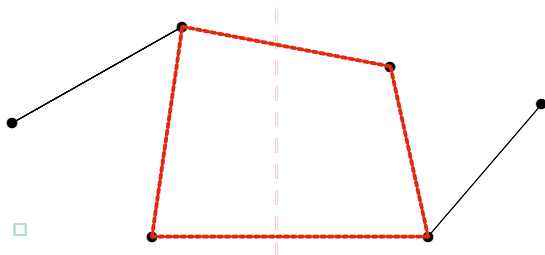
Pokud by existovaly dvě různé cesty R_1, R_2 mezi u a v , tak bychom vzali jejich symetrický rozdíl, podgraf $H = R_1 \Delta R_2$ s neprázdnou množinou hran, kde H zřejmě má všechny stupně sudé. \square Na druhou stranu se však podgraf stromu musí opět skládat z komponent stromů, a tudíž obsahovat vrchol stupně 1 podle Tvzení 10.2, což je spor. \square

Důsledek 10.6. Přidáním jedné hrany do stromu vznikne právě *jedna kružnice*.

Alternativní charakterizace stromů

Na dané množině vrcholů je (vzhledem k inkluzi množin hran) *strom*

- **minimální souvislý** graf (plyne z Věty 10.5)
- a zároveň **maximální acyklický** graf (plyne z Důsledku 10.6).

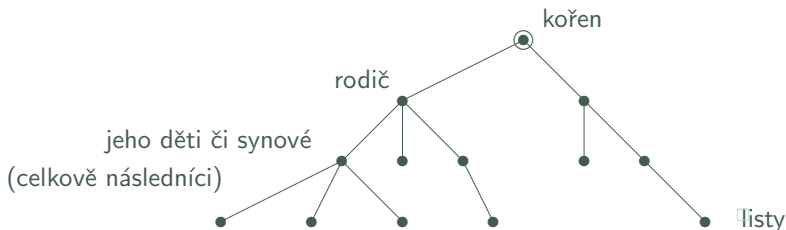


Kořenový strom

Definice: Strom T s vyznačeným vrcholem $r \in V(T)$, zkráceně dvojice (T, r) , nazýváme *kořenovým stromem* s *kořenem* r . □ Vrchol p nazveme *následníkem* vrcholu q , pokud cesta z kořene r do p obsahuje (neboli vede přes) q . Dále p nazveme *potomkem* q , pokud p je následníkem q a zároveň $\{p, q\}$ je hranou T . □

V kořenovém stromu nazveme *listem* každý vrchol, který nemá potomky. □

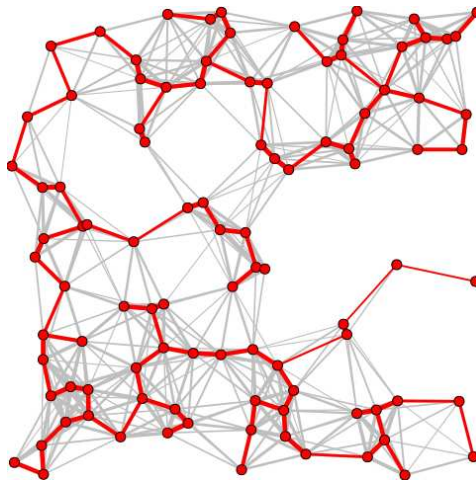
Každý vrchol kořenového stromu je *následníkem* kořene. V přirozeně přeneseném významu se u kořenových stromů používají pojmy rodič, prarodič, děti/synové, sourozenci, předchůdci, atd.



Definice: *Výška kořenového stromu* je rovna největší vzdálenosti z jeho kořene do některého listu.

10.2 Problém minimální kostry

V návaznosti na učivo o stromech se podíváme na tradiční a široce studovaný problém nalezení minimálního souvislého podgrafu (stromu) – této úloze se říká **minimální kostra** neboli MST (z anglického minimum spanning tree).



Minimální kostra a vážené grafy

Definice: Podgraf $T \subseteq G$ souvislého grafu G se nazývá *kostrou*, pokud

- * T je stromem a
- * $V(T) = V(G)$, neboli T propojuje všechny vrcholy G . \square

Avšak každý strom na daných n vrcholech má stejně hran, přesně $n - 1$. Jaký smysl tedy má dávat slovo „minimální“ u hledané kostry?

Definice: Vážený graf je graf G spolu s ohodnocením w hran reálnými čísly

$$w : E(G) \rightarrow \mathbb{R}. \square$$

Vahou (délkou) kostry $T \subseteq G$ váženého souvislého grafu (G, w) rozumíme

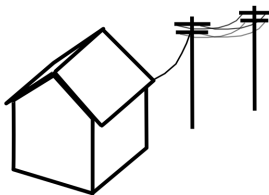
$$d_G^w(T) := \sum_{e \in E(T)} w(e). \square$$

Definice 10.7. Problém minimální kostry (MST) ve váž. grafu (G, w) hledá kostru $T \subseteq G$ s nejmenší možnou vahou (přes všechny kostry grafu G).

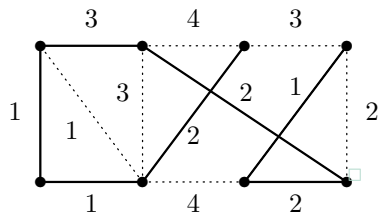
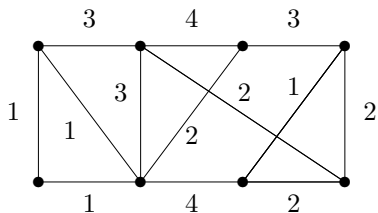
Historické ohlédnutí

Problém minimální kostry je ve skutečnosti historicky úzce svázán s jižní Moravou a Brnem, konkrétně s elektrifikací jihomoravských vesnic ve dvacátých letech! Právě na základě tohoto praktického optimalizačního problému brněnský matematik Otakar Borůvka jako první v matematické literatuře zformuloval a podal řešení problému minimální kostry v roce 1926.

Ve výzkumu minimálních koster pokračoval i velmi dobře známý český matematik Vojtěch Jarník, s publikací v roce 1930 (viz Algoritmus 10.10). První ne-českou publikací na toto téma je pak až Kruskalův hladový algoritmus z roku 1956 (viz Algoritmus 10.8).



Řešení minimální kostry



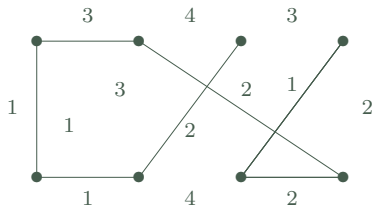
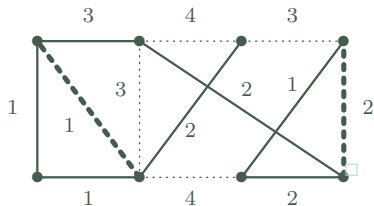
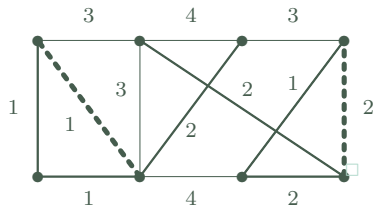
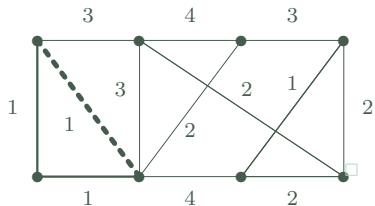
Algoritmus 10.8. Hladový postup pro minimální kostru grafu (G, w) .

Mějme dán *souvislý* vážený graf G s ohodnocením hran w .

1. Seřadíme všechny hrany G jako $E(G) = (e_1, e_2, \dots, e_m)$ tak, že $w(e_1) \leq w(e_2) \leq \dots \leq w(e_m)$. \square
2. Inicializujeme prázdnou kostru $T = (V(G), \emptyset)$. \square
3. Po řadě pro $i = 1, 2, \dots, m$ provedeme následující:
 - Pokud $T + e_i$ *nevytváří kružnici*, tak $E(T) \leftarrow E(T) \cup \{e_i\}$.
(Neboli pokud e_i spojuje různé komponenty souvislosti dosav. T .) \square
4. Na konci T obsahuje minimální kostru grafu G .

Ukážeme si postup hladového algoritmu pro vyhledání kostry výše zakresleného grafu. Hrany si nejprve seřadíme podle jejich vah 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4.

V obrázku průběhu algoritmu používáme tlusté čáry pro vybrané hrany kostry a tečkované čáry pro „zahozené“ hrany. Hrany teď postupně přidáváme do kostry či zahazujeme...



Získáme tak minimální kostru velikosti $1 + 2 + 2 + 3 + 1 + 1 + 2 = 12$, která je v tomto případě (náhodou) cestou, na posledním obrázku vpravo.

Jarníkův (Primův) algoritmus

Algoritmus 10.10. Hledání minimální kostry ve váženém grafu (G, w) .

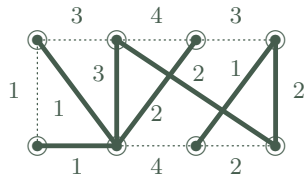
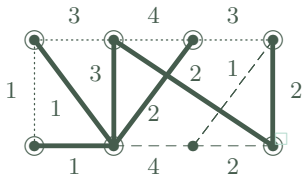
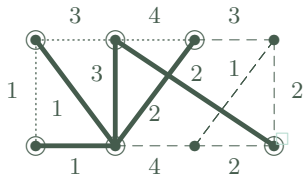
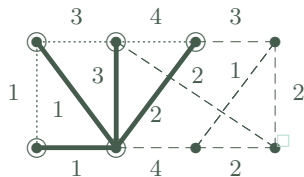
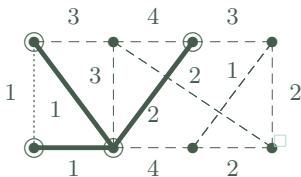
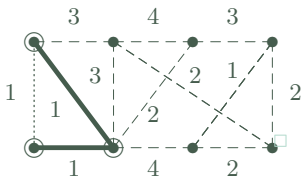
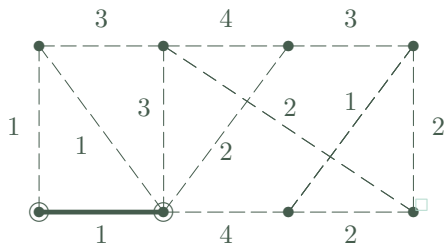
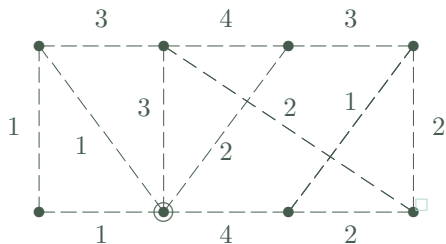
Opět mějme dán *souvislý* vážený graf G s ohodnocením hran w . \square

1. Na začátku zvolíme lib. vrchol $u \in V(G)$ a podstrom $T := (\{u\}, \emptyset)$. \square
2. Dokud $V(T) \neq V(G)$, provádíme:
Nalezneme hranu $f = uv \in E(G)$ nejmenší váhy s jedním koncem u ve $V(T)$ a druhým v ve $V(G) \setminus V(T)$ a položíme $T := T + v + f$. \square

Formálně přesněji tento krok popíšeme následovně.

- * Necht' $X = \{uv \in E(G) : u \in V(T), v \in V(G) \setminus V(T)\}$ a zvolme $f = uv \in X$ minimalizující váhu $w(f)$.
 - * Položme $V(T) := V(T) \cup \{v\}$ a $E(T) := E(T) \cup \{f\}$. \square
3. Nyní je T minimální kosterou grafu G .

Následuje stručná ukázka průběhu Jarníkova algoritmu.



10.3 Dodatek: Výběr různých reprezentantů

S grafy je úzce svázán i tento klasický kombinatorický problém, vybírající různé reprezentanty z daného systému množin.

Definice: Necht' M_1, M_2, \dots, M_k jsou neprázdné množiny. *Systémem různých reprezentantů* množin M_1, M_2, \dots, M_k nazýváme posloupnost *různých* prvků (x_1, x_2, \dots, x_k) takových, že $x_i \in M_i$ pro $i = 1, 2, \dots, k$. \square

Věta 10.12. (Hall) *Necht' M_1, M_2, \dots, M_k jsou neprázdné množiny. Pro tyto množiny existuje systém různých reprezentantů, právě když platí*

$$\forall J \subseteq \{1, 2, \dots, k\} : \left| \bigcup_{j \in J} M_j \right| \geq |J|, \quad (1)$$

neboli pokud sjednocení libovolné skupiny z těchto množin má alespoň tolik prvků, kolik množin je sjednoceno. \square

Co nám tedy Věta 10.12 prakticky říká?

- * Pokud systém různých reprezentantů existuje, stačí jej najít či uhodnout.
- * A pokud neexistuje, stačí nalézt vhodnou množinu J porušující podmínku (1).

Párování grafu a reprezentanti

Přiřazení reprezentantů množinám lze chápat jako hrany grafu, ve kterém na jedné straně jsou naše množiny a na druhé straně jsou jejich všechny prvky.

Takto vypadajícím grafům říkáme *bipartitní* (formálně je bipartitní graf libovolným podgrafem vhodného úplného bipartitního grafu $K_{m,n}$). □

Definice: *Párování* v (nyní bipartitním) grafu G je podmnožina hran $M \subseteq E(G)$ taková, že žádné dvě hrany z M nesdílejí koncový vrchol. □

Uvažujme systém množin M_1, \dots, M_k a označme p_1, \dots, p_m všechny prvky ve sjednocení $M_1 \cup \dots \cup M_k$. Definujme si bipartitní graf G na množině vrcholů $\{1, 2, \dots, k\} \cup \{p_1, \dots, p_m\}$, který je tvořen hranami $\{i, p_j\}$ pro všechny dvojice i, j , pro které $p_j \in M_i$. □ Pak platí:

Tvrzení 10.13. *Množiny M_1, \dots, M_k mají systém různých reprezentantů právě tehdy, když v grafu G existuje párování pokrývající vrcholy $\{1, 2, \dots, k\}$.*

Poznámka: Pro nalezení největšího párování v bipartitním grafu existují rychlé postupy (algoritmy), související s tzv. toky v sítích. Tyto postupy se také dají využít (podle uvedeného tvrzení) k hledání systémů různých reprezentantů.