

# Jakýsi úvod do diskrétní matematiky

Áďa Klepáčovic

11. dubna 2023



# Obsah

<b>1</b>	<b>Teorie grafů</b>	<b>1</b>
1.1	Pohyb v grafu . . . . .	7
1.2	Stromy . . . . .	11
1.2.1	Minimální kostra . . . . .	14
1.2.2	Kruskalův algoritmus . . . . .	17
1.3	Jordanovo centrum . . . . .	20
<b>A</b>	<b>Dodatky k teorii grafů</b>	<b>23</b>
A.1	Souvislost grafu a metrika . . . . .	23



## 1 | Teorie grafů

Velkou část moderní matematiky (zcela jistě topologii, geometrii i algebru) tvoří studium „struktur“. Toto obecně nedefinované slovo obvykle značí množinu s nějakou další informací o vztahu mezi jejími prvky – tím obvykle bývá operace nebo třeba, jako v případě grafů, relace.

Tato kapitola zároveň značí jakýsi milník ve vývoji matematického myšlení, především algebraickým směrem. Můžeme se totiž začít bavit o speciálních zobrazeních, které zachovávají strukturu na množinách, mezi kterými vedou, tzv. *homomorfismech*; pochopit, že je dobré mít více popisů stejné struktury ekvivalentních v tom smyslu, že poskytují stejné množství informací, přestože se o žádné bijekci nedá formálně hovořit; uvidět, že je užitečné dva různé grafy (či obecně dvě různé struktury) považovat za stejné, když se liší pouze zanedbatelně.

Jednou, avšak zdaleka ne *jedinou*, motivací pro teorii grafů je schopnost analyzovat struktury tvořené množinou „uzlů“, mezi některýmiž vedou „spojnice“. Takováto struktura úspěšně modeluje až neuvěřitelné množství přírodních i společenských úkazů. Mezi nimi jmenujmež

- návrhy elektrických obvodů, kde uzly jsou elektrická zařízení a spojnice jsou kabely mezi nimi vedoucí;
- lingvistické modely, kde uzly jsou slova a spojnice vede mezi těmi syntakticky souvisejícími;
- studium molekul, kde uzly jsou atomy a spojnice vazby mezi nimi;
- analýza šíření fámy v sociologii, kde uzly jsou lidské komunity a spojnice vede mezi komunitami s bezprostředním kontaktem.

Snad pro to, že uzly a spojnice grafu se obvykle kreslí jako body a úsečky v prostoru, ujal se pro ně názvy *vrcholy* a *hrany* (jako v mnohoúhelnících), respektive. Struktura zvaná *graf* tedy sestává ze dvou údajů:

- (1) množiny (obvykle konečné) vrcholů značené  $V$  a
- (2) množiny hran  $E$ , která je spjata s množinou vrcholů; toto „sepětí“ se však definuje různě, v závislosti na vkusu a aplikaci. My si ukážeme tři z jistě většího množství různých definic.

Asi prvním přirozeným kandidátem pro „strukturu“ je množina s relací. To je také první způsob, jak si budeme definovat pojem *graf*. Je to také ten nejobecnější v tom smyslu, že jeho pouze drobné modifikace nám umožní definovat i obdobné struktury, jež také zmateně slují *grafy*, byť s přípojným atributem.

Abychom úsečky mezi body mohli popsat jako relaci, čili pomocí uspořádaných dvojic bodů, zcela jistě budeme požadovat, aby nevedly úsečky z bodu do něho samého. Úsečku délky 0 lze totiž triviálně ztotožnit s bodem. Dále, úsečka z bodu  $A$  do bodu  $B$  je jistě tatáž, která úsečka z bodu  $B$  do bodu  $A$ . Tento fakt musí rovněž relace  $E$  odrážet.

První vlastností relace se, snad nepřekvapivě, říká *antireflexivita*. Čili, relace  $E$  na  $V$  je *antireflexivní*, když hrana  $(v, v) \notin E$  pro každý vrchol  $v \in V$ .

Druhou vlastnost už jsme potkali a nazvali ji symetrií. Požadujeme, aby s hranou  $(v, w) \in E$  obsahovala  $E$  též hranu  $(w, v) \in E$  pro každé dva vrcholy  $v, w \in V$ .

**Definice 1.0.1 (Graf poprvé).** Dvojici  $G := (V, E)$ , kde  $V$  je konečná množina a  $E$  je relace na  $V$ , nazveme *grafem*, pokud je  $E$  **antireflexivní** a **symetrická**.

**Poznámka.** Z hlediska ryze formálního neodpovídá tato definice dokonale naší geometrické představě. My jsme totiž pouze požadovali, aby  $E$  obsahovala jak úsečku z  $v$  do  $w$ , tak úsečku z  $w$  do  $v$ , ale nikoli, aby se jednalo o *tutéž* úsečku. Tedy, jedna úsečka mezi body je v množině  $E$  reprezentována dvěma dvojicemi.

Nápravou by bylo definovat navíc ještě relaci  $R$  na  $E$ , kde  $(v, v')$  je v relaci  $R$  s  $(w, w')$  právě tehdy, když  $(w, w') = (v', v)$  nebo  $(w, w') = (v, v')$ . Jinak řečeno, úsečka z bodu  $v$  do bodu  $v'$  je v relaci sama se sebou a s úsečkou z bodu  $v'$  do bodu  $v$ .

Uvážíme-li pak jako hrany grafu  $G$  nikoli množinu  $E$ , ale její třídy ekvivalence podle  $R$  (**Ověřte, že  $R$  je ekvivalence!**), dostaneme již přesnou množinovou paralelu bodů a úseček.

My však budeme v zájmu přehlednosti tento nedostatek ignorovat, protože není pro pochopení ani rozvoj teorie relevantní.

Cesta k druhé možné definici grafu není od první daleká. Stačí vlastně relaci  $E$  interpretovat trochu jinak. Přece, antireflexivní a symetrická relace je „totéž“ jako množina dvouprvkových podmnožin  $V$ .

Vskutku, vezměme nějakou  $E' \subseteq \binom{V}{2}$ . Relaci  $E$  z [definice 1.0.1](#) sestojíme tak, že z množiny  $\{v, w\} \in E'$  vyrobíme dvojici  $(v, w)$  a  $(w, v)$ . Protože prvky v množině nejsou uspořádané a nemohou se opakovat, dává tato konstrukce opravdu antireflexivní a symetrickou relaci. Vizualně odpovídá rozdělení úsečky mezi  $v$  a  $w$  na šipku z  $v$  do  $w$  a šipku z  $w$  do  $v$ .

Z druhé strany, mějme nějakou antireflexivní a symetrickou relaci  $E$  na  $V$ . Protože s dvojicí  $(v, w)$  obsahuje  $E$  i dvojici  $(w, v)$ , můžeme z těchto dvojic ztvárnit množinu  $\{v, w\}$ . Relace  $E$  je antireflexivní, čili se nemůže stát, že  $v = w$ , a množina  $\{v, w\}$  je pročež vždy dvouprvková. Posbíráme-li všechny množiny  $\{v, w\}$  do jedné velké množiny  $E'$ , bude platit  $E' \subseteq \binom{V}{2}$ . Vizualně odpovídá tato konstrukce slepení šipky z  $v$  do  $w$  a šipky z  $w$  do  $v$  do jedné úsečky mezi  $v$  a  $w$ .

**Výstraha.** Mezi množinami  $E$  a  $E'$  **nemůže existovat bijekce**, třeba jen pro to, že  $\#E = 2\#E'$ . Co konstrukce v předchozích dvou odstavcích ukazují, je pouze fakt, že pro naše účely definují  $E$  a  $E'$  stejnou strukturu na množině  $V$ .

Ovšem, uvážili-li bychom místo  $E$  pouze třídy ekvivalence jejích prvků podle relace  $R$  popsané v poznámce pod [definicí 1.0.1](#), pak bychom skutečně tímto způsobem našli bijekci s množinou  $E'$ .

**Definice 1.0.2 (Graf podruhé).** Dvojici  $G := (V, E')$ , kde  $V$  je konečná množina a  $E' \subseteq \binom{V}{2}$ , nazveme *grafem*.

Třetí pohled na hrany v grafu je více „kategoriální“. Zatímco množiny  $E$  a  $E'$  jsou závislé ve své definici na množině  $V$ , třetí množina hran  $E''$ , kterou si zde definujeme, bude libovolná konečná množina.

Tento popis grafové struktury bude odpovídat trochu jiné představě; konkrétně takové, kdy začínáme s množinou bodů  $V$  a s množinou šipek  $E$  (jež jsou od sebe naprosto odděleny) a oba konce každé šipky zapíchneme do dvou různých bodů z  $V$ . Toto „zapíchnutí“ realizují zobrazení  $s, t : E'' \rightarrow V$  (z angl. *source* a *target*), která zobrazují šipky z  $E''$  do bodů z  $V$ . Přičemž

budeme trvat na tom, aby  $s(e) \neq t(e)$  pro všechny šipky  $e \in E''$  a navíc, aby pro každou  $e \in E''$  existovala šipka  $e' \in E''$  taková, že  $s(e) = t(e')$ ,  $t(e) = s(e')$ . Lidsky řečeno, nesmíme zapíchnout konce šipky do téhož vrcholu a, když zapíchneme začátek šipky do bodu  $v$  a její konec do bodu  $w$ , pak musíme vzít další šipku, jejíž začátek zapíchneme do  $w$  a konec do  $v$ .

Je snadné si rozmyslet, že z množiny šipek  $E''$  zrekonstruujeme množinu  $E$  z [definice 1.0.1](#) tak, že z šipky  $e \in E''$  vytvoříme dvojici  $(s(e), t(e)) \in E$ . Podmínky kladené na zobrazení  $s$  a  $t$  zaručují, že vzniklá množina  $E$  je relace na  $V$ , která je antireflexivní a symetrická. V tomto případě dává uvedená konstrukce dokonce bijekci  $E \cong E''$ , čili jsme opět definovali tutéž strukturu na  $V$ .

Tato struktura bude zvlášť užitečná, až budeme probírat *toky v síti*.

**Definice 1.0.3 (Graf potřetí).** Čtveřici  $G := (V, E'', s, t)$ , kde  $V$  a  $E''$  jsou konečné množiny a  $s$  a  $t$  jsou zobrazení  $E'' \rightarrow V$  nazveme *grafem*, pokud

- (a)  $s(e) \neq t(e) \forall e \in E''$  a
- (b)  $\forall e \in E'' \exists e' \in E'' : s(e) = t(e') \wedge t(e) = s(e')$ .

**Poznámka.** Opět, aby [definice 1.0.3](#) odpovídala představě bodů a úseků (nebo oboustranných šipek), museli bychom definovat relaci  $R$  na  $E$  tak, aby  $e$  a  $e'$  byly v  $R$ , právě když  $s(e) = s(e')$  a  $t(e) = t(e')$  nebo  $s(e) = t(e')$  a  $t(e) = s(e')$ . V takovém případě bychom mohli sestavit bijekci mezi  $E''$  a  $E'$  z [definice 1.0.2](#).

**Příklad.** Ať  $V := \{1, 2, 3, 4, 5\}$  a

- (1)  $E := \{(1, 2), (2, 1), (1, 3), (3, 1), (1, 4), (4, 1), (1, 5), (5, 1), (2, 3), (3, 2), (4, 5), (5, 4)\}$ ;
- (2)  $E' := \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{2, 3\}, \{4, 5\}\}$ ;
- (3)  $E'' := \{e_1, e_2, e_3, e_4, e_5, e_6, e'_1, e'_2, e'_3, e'_4, e'_5, e'_6\}$ , kde
  - (a)  $s(e_1) = s(e_2) = s(e_3) = s(e_4) = 1, s(e_5) = 2, s(e_6) = 4,$

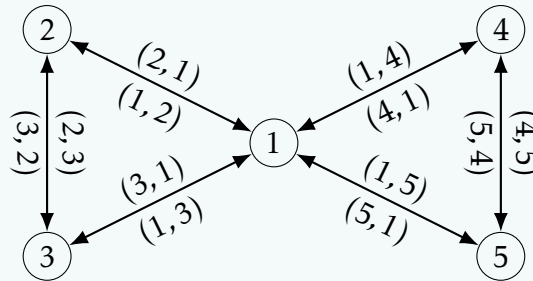


(b)  $t(e_1) = 2$ ,  $t(e_2) = t(e_5) = 3$ ,  $t(e_3) = 4$ ,  $t(e_4) = t(e_6) = 5$  a

(c)  $(s(e_i), t(e_i)) = (t(e'_i), s(e'_i))$  pro všechna  $i \leq 6$ .

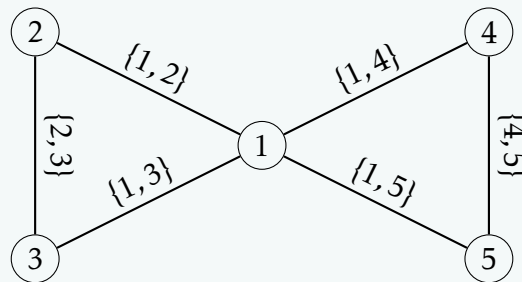
Není těžké nahlédnout, že  $E$ ,  $E'$  i  $(E'', s, t)$  definují tutéž strukturu na  $V$ . Nakreslíme si grafy  $G = (V, E)$ ,  $G' = (V, E')$  a  $G'' = (V, E'', s, t)$ . Přičemž hrany z  $E$  budeme kreslit jako oboustranné šipky, ty z  $E'$  jako prosté úsečky a ty z  $E''$  rozdělíme na dvě protichůdné šipky, abychom vyjádřili rozdíly v interpretaci těchto grafových struktur.

Graf  $G = (V, E)$  vypadá například [takto](#). Pomněte, že například oboustranná šipka mezi vrcholy 1 a 2 představuje ve skutečnosti **dvě** dvojice –  $(1, 2)$  a  $(2, 1)$  z relace  $E$ .



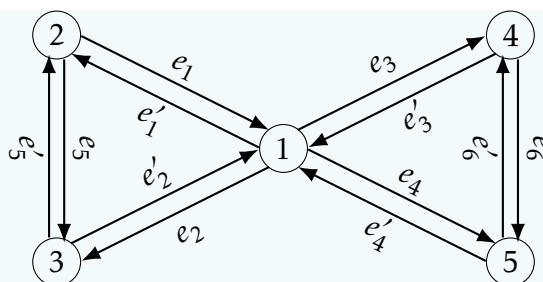
Obrázek 1: Graf jako množina  $V$  s relací  $E$ .

Zcela stejně vypadá i graf  $G' = (V, E')$ .

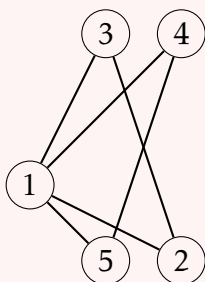


Obrázek 2: Graf jako množiny  $V$  a  $E' \subseteq \binom{V}{2}$ .

Konečně,  $G'' = (V, E'', s, t)$  můžeme načrtnout taktéž velmi podobně.

Obrázek 3: Graf jako množina  $V$  s trojicí  $(E'', s, t)$ .

**Výstraha.** Grafová struktura je obecně zcela nezávislá na jejím nakreslení. Například graf  $G = (V, E')$  z [předchozího příkladu](#) lze ekvivalentně vyobrazit třeba následovně.

Obrázek 4: Graf  $G = (V, E')$  nakreslený jinak.

V následujícím textu spojíme všechny tři interpretace dohromady a pro  $v, w \in V$  budeme hranu mezi  $v$  a  $w$  značit zjednodušeně jako  $vw$ . Pokud nehrozí nedorozumění, budeme pod tímto zápisem rozumět hranu, jejíž začátek je  $v$  a konec  $w$ , čili  $s(vw) = v$  a  $t(vw) = w$ . Avšak, kdykoli se nám to bude hodit, ztotožníme ji bez okolků s hranou  $wv$  s obrácenými konci.

Tento neformální přístup k popisu hran se může zdát jako nebezpečný, ale jak uvidíme, ve skutečnosti velmi zjednodušuje zápis a újma na rigorozitě je obecně minimální. Kompletněji řečeno, hranou mezi dvěma vrcholy  $v, w \in V$  myslíme buď dvojici  $(v, w) \in E$  nebo dvojici  $(w, v) \in E$  nebo množinu  $\{v, w\} \in E'$  nebo prvek  $e \in E''$  takový, že  $s(e) = v$  a  $t(e) = w$ , nebo prvek  $e' \in E''$  takový, že  $s(e') = w$  a  $t(e') = v$ , a je nám to u ...

Obecně, v teorii grafů se velmi často pracuje s konečnými posloupnostmi (či  $n$ -ticemi, chcete-li) vrcholů a hran. Zavedeme proto zjednodušené zna-

čení  $x_1x_2\cdots x_n$  pro uspořádanou  $n$ -tici  $(x_1, \dots, x_n)$ . Kdyby hrozil konflikt se zápisem součinu prvků  $x_1, \dots, x_n$ , samozřejmě tento úzus dočasně opustíme.

## 1.1 Pohyb v grafu

Jak jsme zmínili na začátku kapitoly, mnoho aplikací teorie grafů využívá interpretace této struktury jako množiny „uzlů“ se „spojnicemi“, po kterých se dá mezi uzly pohybovat. V této podsekci dáme pohybu po spojnicích formální tvář.

Nejzákladnějším typem pohybu v grafu je libovolná posloupnost vrcholů se zcela přirozenou podmínkou, že mezi vrcholy v posloupnosti bezprostředně za sebou musí vést hrana. Takové posloupnosti se říká *sled* (angl. *walk*).

**Definice 1.1.1 (Sled poprvé).** Ať  $G = (V, E)$  je graf. Posloupnost vrcholů  $v_1v_2\cdots v_n$  nazveme *sledem* v grafu  $G$ , pokud  $v_iv_{i+1} \in E$  pro každý index  $i \in \{1, \dots, n-1\}$ .

Je však užitečné si uvědomit, že každý sled lze ekvivalentně definovat jako posloupnost navazujících hran. Tento pohled (jak uvidíme později v kapitole) má své aplikace – často nás totiž zajímá, po kterých spojnicích chodíme, spíše než které uzly procházíme.

Máme-li sled  $v_1v_2\cdots v_n$ , tak každá dvojice  $v_iv_{i+1}$  musí být hranou v  $G$ . To ovšem znamená, že zcela identickou informaci poskytuje i posloupnost hran  $e_1e_2\cdots e_{n-1}$ , kde  $e_i = v_iv_{i+1}$ .

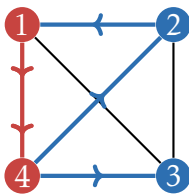
**Poznámka.** Tento princip, který prostupuje matematické struktury, je až překvapivě zásadního významu. Zobecníme-li trochu předchozí pozorování, uvědomíme si, že množina hran vlastně už v sobě obsahuje informaci o všech prvcích množiny  $V$ . Tedy bychom dokonce mohli definovat graf pouze jako množinu hran a množinu vrcholů bychom takto získali automaticky. Opak samozřejmě není pravdou.

Situace, kdy struktura na množině (či něčem složitějším) je dostatečně „hustá“, aby obsahovala kompletní informaci o této množině,

je obecně velmi žádaná, jelikož struktura je často konstruována systematicky (a tedy ji rozumíme lépe) ve srovnání s náhodnou volbou její báze množiny.

Pro nedostatek představivosti uvedeme příklad z homologické algebry, kde injektivní a projektivní rezolventy modulů obsahují již kompletní informaci o daném modulu. Tedy člověku pro pochopení teorie modulů z homologického hlediska „stačí“ studovat injektivní a projektivní moduly, které jsou z definice více omezené než moduly obecné, pročť snadněji popsatelné.

**Definice 1.1.2 (Sled podruhé).** Ať  $G = (V, E)$  je graf. Posloupnost hran  $e_1 e_2 \dots e_n$  nazveme *sledem* v grafu  $G$ , pokud  $t(e_i) = s(e_{i+1})$  pro všechna  $i \in \{1, 2, \dots, n-1\}$ .



Obrázek 5: Příklad sledu 142143 v grafu. Jednou navštívené hrany a vrcholy jsou značené **modře**, dvakrát navštívené **červeně**.

Méně obecným pohybem v grafu je sled, ve kterém se nesmějí opakovat hrany. Takové sledy lze najít často třeba v běžné situaci, kdy si jako rozvůzce jídla plánujete cestu městem. Jako uzly si označíte například místa, která musíte objet, a hrany budou nejkratší trasy mezi nimi. Projít přes některá místa vícekrát vám příliš vadit nemusí, ale ztrácet čas cestováním po stejné trase sem a tam byste neradi.

Takovému sledu se říká *tah* (angl. *trail*). Jeho definice je velmi přirozená.

**Definice 1.1.3 (Tah).** Ať  $G = (V, E)$ . *Tahem* v grafu  $G$  nazveme buď

- (1) sled (vrcholů)  $v_1 v_2 \dots v_n$  takový, že  $v_i v_{i+1} \neq v_j v_{j+1}$  pro všechna  $i \neq j$ , nebo
- (2) sled (hran)  $e_1 e_2 \dots e_{n-1}$  takový, že  $e_i \neq e_j$  pro všechna  $i \neq j$ .

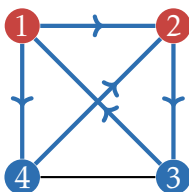
**Výstraha.** Obě uvedené definice tahu jsou v našem (částečně neformálním) pojetí hran skutečně ekvivalentní. Napíšeme-li totiž  $v_i v_{i+1} \neq v_j v_{j+1}$  myslíme tím vlastně dvě nerovnosti:

$$(v_i, v_{i+1}) \neq (v_j, v_{j+1}) \wedge (v_i, v_{i+1}) \neq (v_{j+1}, v_j);$$

nebo zápis můžeme též chápat jako

$$\{v_i, v_{i+1}\} \neq \{v_j, v_{j+1}\},$$

čili každou hranu chceme projít (kterýmkoli směrem) maximálně jednou.



Obrázek 6: Příklad tahu 142312 v grafu. Jednou navštívené hrany a vrcholy jsou značené modře, dvakrát navštívené červeně.

Posledním, a asi nejčastěji zkoumaným, typem pohybu grafem je *cesta* (angl. *path*), což je sled, ve kterém se nesmějí opakovat ani hrany ani vrcholy. Pro jednoduchost je užitečné si uvědomit, že z podmínky neopakování vrcholů automaticky plyne podmínka neopakování hran. Pokud bychom totiž chtěli po nějaké hraně přejít dvakrát, tak zároveň dvakrát projdeme její koncové vrcholy. Definovat cestu pomocí sledu vrcholů je tudíž přímočaré.

U sledu hran je to však horší. Protože každá hrana definuje dva vrcholy sledu, je třeba zařídit, aby se oba koncové body každé hrany lišily od obou koncových bodů jiné hrany, ale jenom tehdy, když hrany nejdou bezprostředně za sebou. To bohužel vede na trochu neintuitivní definici cesty přes hrany. Totiž, cesta obsahující  $n$  hran nutně obsahuje  $n + 1$  vrcholů, protože po sobě jdoucí hrany vždy sdílejí jeden vrchol. Rafinovaně se tedy cesta přes hrany dá vyjádřit jako tah, kde sjednocení přes všechny hrany (vnímané tentokrát jako množiny) má velikost právě  $n + 1$ .

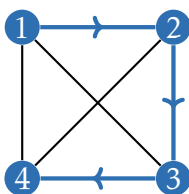
Studium cest má aplikace například právě v návrhu elektrických obvodů, kdy zcela jistě nechcete, aby do připojených zařízení šel proud z více, než

jednoho místa.

**Definice 1.1.4 (Cesta).** Ať  $G = (V, E)$  je graf. *Cestou* v grafu  $G$  nazveme buď

- (1) sled (vrcholů)  $v_1 v_2 \cdots v_n$  takový, že  $v_i \neq v_j$  pro všechna  $i \neq j$ , nebo
- (2) tah (hran)  $e_1 e_2 \cdots e_{n-1}$  takový, že

$$\# \bigcup_{i=1}^{n-1} e_i = n.$$



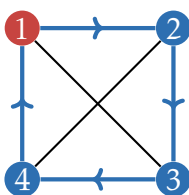
Obrázek 7: Příklad cesty 1234 v grafu. Navštívené hrany a vrcholy jsou značené modře.

Posledním důležitým konceptem v grafu je tzv. *cyklus* (též *kružnice*, angl. *cycle*). Jedná se vlastně o „téměř cestu“, která končí tam, kde začala. Konkrétně je to tedy cesta prodloužená o svůj první vrchol (samozřejmě automaticky předpokládáme, že existuje hrana z posledního vrcholu do prvního).

**Definice 1.1.5 (Cyklus).** Ať  $G = (V, E)$  je graf. *Cyklem* v grafu nazveme buď

- (1) sled (vrcholů)  $v_1 v_2 \cdots v_n v_1$ , pokud  $v_1 v_2 \cdots v_n$  je cesta v  $G$ , nebo
- (2) tah (hran)  $e_1 e_2 \cdots e_{n-1} e_n$ , kde  $t(e_n) = s(e_1)$  a  $e_1 e_2 \cdots e_{n-1}$  je cesta v  $G$ .

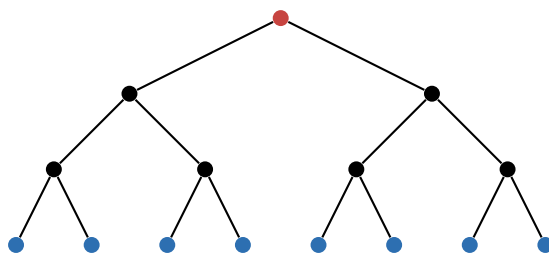
Příklad cyklu vidíte na [obrázku 8](#). Jejich význam zatím necháme zahalen tajemstvím, jež má být odkryto v nejvíce dramatickou chvíli.



Obrázek 8: Příklad cyklu 12341 v grafu. Jednou navštívené hrany a vrcholy jsou značené **modře**, dvakrát navštívené **červeně**.

## 1.2 Stromy

Chvilí se budeme bavit o stromech – ano, těch s listy a kořenem. Stromy jsou speciální typy grafů, které se takto nazývají ne nadarmo. Jsou to totiž grafy, u kterých si člověk může zvolit jakýsi „počáteční“ vrchol (zvaný *kořen*), z něž se po cestě (ve smyslu [definice 1.1.4](#)) vždy dostane do jednoho z „koncových“ vrcholů, tzv. *listů*. Příklad stromu je na [obrázku 9](#).



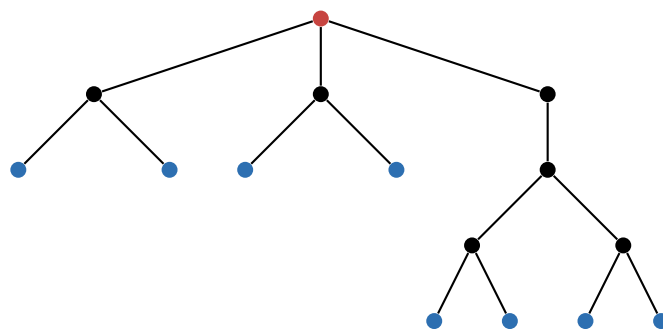
Obrázek 9: Příklad stromu. Kořen je značen **červeně** a listy **modře**.

**Výstraha.** Listy stromu jsou určeny jednoznačně jeho strukturou (jsou to ty jediné vrcholy, do nichž cesty od kořene mohou pouze vést a nikoli jimi procházet). Za kořen lze však volit libovolný vrchol, klidně i jeden z listů. Strom z [obrázku 9](#) může proto vypadat i jak ukazuje [obrázek 10](#).

Často není nutno o kořenu a listech stromu hovořit, pokud je v konkrétní situaci irrelevantní rozlišovat jednotlivé vrcholy. Součástí definice stromu (kterou si záhy odvodíme) kořen ani listy nejsou.

Nyní si rozmyslíme dvě ekvivalentní definice stromu.

Za první podmínku, abychom mohli graf nazvat stromem, budeme považovat fakt, že od kořene se dá dostat po hranách do každého z listů. Ekví-



Obrázek 10: Strom z obrázku 9 s jinou volbou kořene.

valentně, že z každého vrcholu se dá cestou dostat do každého vrcholu, protože za kořen lze, jak jsme nahlédli, volit kterýkoli vrchol, a cestu z kořene do vrcholu můžeme zkrátit tak, aby končila v nějakém vrcholu, jímž původně procházela. Grafy splňující tuto podmínku slují *souvislé*.

**Definice 1.2.1 (Souvislý graf).** Graf  $G = (V, E)$  nazveme *souvislým*, pokud pro každé dva vrcholy  $v, w \in V$  existuje cesta z  $v$  do  $w$ , tedy cesta  $v_1 v_2 \dots v_n$ , kde  $v_1 = v$  a  $v_n = w$ .

Samotný název „strom“ plyne z faktu, že se jako graf pouze „větví“, čímž míníme, že při cestě směrem od (libovolného) kořene se jeden může pouze přibližovat k listům, ale nikoli se dostat zpět blíže ke kořeni. To lze snadno zařídit tak, že zakážeme cykly. Totiž, neexistuje-li v grafu cyklus, pak se po libovolné cestě ze zvoleného vrcholu můžeme od tohoto vrcholu pouze vzdalovat. Takové grafy nazveme, přirozeně, *acyklické*.

**Definice 1.2.2 (Acyklický graf).** Graf  $G = (V, E)$  nazveme *acyklický*, pokud neobsahuje cyklus o aspoň dvou vrcholech (samotné vrcholy jsou totiž z [definice](#) vždy cykly).

**Definice 1.2.3 (Strom).** Graf  $G = (V, E)$  nazveme *stromem*, je-li souvislý a acyklický.

Na začátku sekce jsme slíbili ještě ekvivalentní definici stromu; ta činí začnou část důvodu užitečnosti stromů, především v informatice.

Ukazuje se totiž, že neexistence cyklů spolu se souvislostí způsobují, že



mezi dvěma vrcholy vede vždy **přesně jedna cesta**. Po chvíli zamyšlení snad toto nepřichází jako nijak divoké tvrzení. Přeci, pokud by mezi vrcholy vedly cesty dvě, pak vrchol, kde se rozpojují, a vrchol, kde se opět spojují, by byly součástí cyklu uvnitř stromu, který jsme výslovně zakázali. Třeba překvapivější je fakt, že platí i opačná implikace.

**Tvrzení 1.2.4 (Ekvivalentní definice stromu).** Graf  $G = (V, E)$  je stromem ve smyslu [definice 1.2.3](#) právě tehdy, když mezi každými dvěma vrcholy  $G$  vede přesně jedna cesta.

**Důkaz.** Dokazujeme dvě implikace. Obě budeme dokazovat v jejich *kontrapozitivní* formě, tedy jako obrácenou implikaci mezi negacemi výroků. Lidsky, dokážeme, že (1) když existují vrcholy, mezi kterými nevede žádná nebo vede více než jedna cesta, pak  $G$  není strom, a (2) když  $G$  není strom, tak existují vrcholy, mezi kterými nevede žádná cesta nebo vede více než jedna.

- (1) Pokud existují vrcholy, mezi kterými nevede cesta, pak  $G$  není souvislý, což odporuje [definici stromu](#). Budeme tedy předpokládat, že existují vrcholy  $v, w$ , mezi kterými vedou různé cesty  $v_1 \cdots v_n$  a  $v'_1 \cdots v'_m$ , kde  $v_1 = v'_1 = v$  a  $v_n = v'_m = w$ . Myšlenka důkazu je najít cyklus obsahující vrchol, kde se cesty rozdělují, a vrchol, kde se opět spojují. Vizte [obrázek 11a](#).

Ať  $r$  (od rozpojení) je **největší** index takový, že  $v_i = v'_i$  pro všechna  $i \leq r$  (čili  $v_r = v'_r$  je vrchol, ve kterém se cesty rozpojují). Ten určitě existuje, protože cesty se v nejhorším případě dělí už ve vrcholu  $v = v_1$ .

Podobně, ať  $s$  (od spojení) je **nejmenší** index takový, že existuje  $k \in \mathbb{Z}$  splňující  $v_j = v'_{j+k}$  pro všechna  $j \geq s$ . Čili, vrchol  $v_s = v'_{s+k}$  je vrchol, ve kterém se cesty opět spojily. Ovšem, mohlo se tak stát v okamžiku, kdy jsme po jedné cestě prošli více nebo méně vrcholů než po druhé – tento počet vyjadřuje ono číslo  $k$ . Takový vrchol jistě existuje, v nejhorším je to přímo koncový vrchol  $w = v_n$ .

Zřejmě platí  $r < s$ , jinak by cesty nebyly různé. Potom je ovšem například posloupnost vrcholů

$$(v_r, v_{r+1}, \dots, v_s = v'_{s+k}, v'_{s+k-1}, \dots, v'_r = v_r)$$

cyklem v  $G$ . Tedy ani v tomto případě  $G$  není strom.

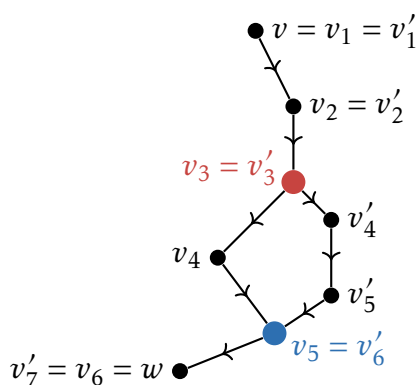
- (2) Když  $G$  není strom, tak není souvislý nebo obsahuje cyklus. Když  $G$  není souvislý, tak existují vrcholy, mezi nimiž nevede v  $G$  cesta, což protirečí podmínce, aby mezi každým párem vrcholů vedla přesně jedna.

Budeme tedy předpokládat, že  $G$  obsahuje cyklus  $v_1 v_2 \cdots v_n$  (tedy  $v_n = v_1$  a  $n \geq 2$ ). Pak ovšem pro libovolné indexy  $i < j \leq n$  jsou posloupnosti

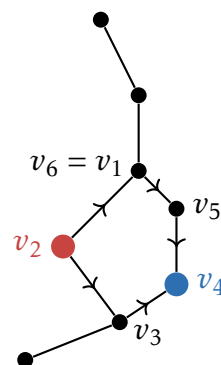
$$(v_i, v_{i+1}, \dots, v_j) \quad \text{a} \quad (v_i, v_{i-1}, \dots, v_1 = v_n, v_{n-1}, \dots, v_j)$$

dvě různé cesty mezi  $v_i$  a  $v_j$ . Vizte [obrázek 11b](#).

Tím je důkaz dokončen. □



(a) Část (1) důkazu [tvrzení 1.2.4](#). Zde  $r = 3, s = 5$  a  $k = 1$ . Sestrojený cyklus je  $v_3 v_4 v_5 v_6 v_7 v_3$ .



(b) Část (2) důkazu [tvrzení 1.2.4](#). Zde  $i = 2, j = 4$  a sestrojené cesty jsou  $v_2 v_3 v_4$  a  $v_2 v_1 v_5 v_4$ .

Obrázek 11: Ilustrace k důkazu [tvrzení 1.2.4](#).

### 1.2.1 Minimální kostra

Ne všechny hrany jsou si rovny. Kterési se rodí krátké, jiné dlouhé; kteréši štíhlé, jiné otlé; kteréši racionální, jiné iracionální.

Často nastávají situace, kdy jeden potřebuje hranám grafu přiřadit nějakou hodnotu, obvykle číselnou, která charakterizuje klíčovou vlastnost této hrany. Při reprezentaci dopravní sítě grafem to může být délka silnice

či její vytížení, při reprezentaci elektrických obvodů pak například odpor. V teorii grafů takové přiřazení hodnoty hranám grafu sluje *ohodnocení*.

**Definice 1.2.5 (Ohodnocený graf).** Ať  $G = (V, E)$  je graf. Libovolné zobrazení  $w : E \rightarrow \mathbb{R}^+ = (0, \infty)$  nazveme *ohodnocením* grafu  $G$ . Trojici  $(V, E, w)$ , kde  $w$  je ohodnocení  $G$ , nazveme *ohodnoceným grafem*.

**Poznámka.** Každý graf  $G = (V, E)$  lze triviálně ztotožnit s ohodnoceným grafem  $(V, E, w)$ , kde  $w : E \rightarrow \mathbb{R}^+$  je konstantní zobrazení. Obvykle se volí konkrétně  $w \equiv 1$ , tedy zobrazení  $w$  takové, že  $w(e) = 1$  pro každou  $e \in E$ .

Porozumění struktuře ohodnocených grafů může odpovědět na spoustu zajímavých (jak prakticky tak teoreticky) otázek. Můžeme se kupříkladu ptát, jak se nejlépe (vzhledem k danému ohodnocení) dostaneme cestou z jednoho vrcholu do druhého. Slovo „nejlépe“ zde chápeme pouze intuitivně. V závislosti na zpytovaném problému můžeme požadovat, aby cesta třeba minimalizovala či maximalizovala součet hodnot všech svých hran přes všechny možné cesty mezi danými vrcholy. Jsou však i případy, kdy člověk hledá cestu, která je nejbližší „průměru“.

Abychom pořád neříkali „součet přes všechny hrany cesty“, zavedeme si pro toto často zkoumané množství název *váha cesty*. Čili, je-li  $\mathcal{P} := e_1 \cdots e_n$  cesta v nějakém ohodnoceném grafu  $G$ , pak její vahou rozumíme výraz

$$w(\mathcal{P}) := \sum_{i=1}^n w(e_i).$$

Zápis  $w(\mathcal{P})$  můžeme vnímat buď jako zneužití zavedeného značení, nebo jako fakt, že jsme zobrazení  $w$  rozšířili z množiny všech hran na množinu všech cest v grafu  $G$  (kde samotné hrany jsou z [definice](#) též cesty).

**Poznámka.** Záměrně jsme užili slovního spojení *váha cesty* místo snad přirozenějšího *délka cesty*. V teorii grafů se totiž délkou cesty myslí obvykle počet hran (nebo vrcholů), které obsahuje. Délka cesty  $\mathcal{P} = e_1 \cdots e_n$  je tudíž  $n$  (resp.  $n + 1$ ), bo obsahuje  $n$  hran (resp.  $n + 1$  vrcholů).

Tento úzus svědčí účelu [předchozí poznámky](#). Pokud totiž každý graf bez ohodnocení vnímáme vlastně jako ohodnocený graf, kde každá

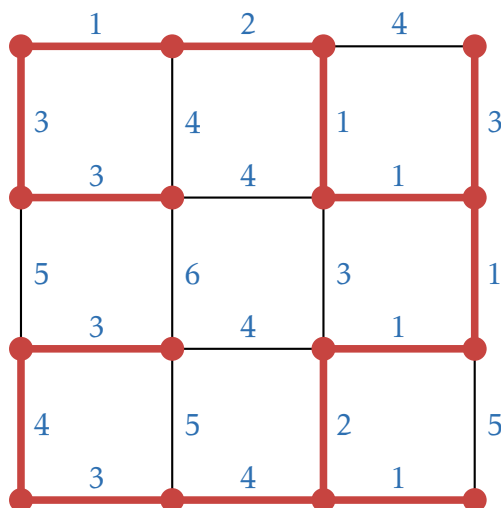
hrana má váhu přesně 1, pak váha každé cesty je rovna její délce.

První (a nejjednodušší) problém, kterým se budeme zabývat, je nalezení *minimální kostry* (angl. *spanning tree*).

**Definice 1.2.6 (Minimální kostra).** Ať  $G = (V, E, w)$  je **souvislý** ohodnocený graf. Ohodnocený graf  $K = (V', E', w)$  nazveme *minimální koustou* grafu  $G$ , pokud je souvislý,  $V' = V$  (tedy  $K$  obsahuje všechny vrcholy  $G$ ),  $E' \subseteq E$  a

$$\sum_{e \in E'} w(e)$$

je minimální vzhledem ke všem možným volbám podmnožiny  $E' \subseteq E$ . Lidsky řečeno, graf  $K$  spojuje všechny vrcholy  $G$  tím „nejlevnějším“ způsobem vzhledem k ohodnocení  $w$ .



Obrázek 12: **Minimální kostra** grafu s ohodnocením  $w$ .

**Pozorování.** Minimální kostra ohodnoceného grafu je strom.

**Důkaz.** Kdyby minimální kostra nebyla strom, pak buď není souvislá, což jsme výslovně zakázali, nebo obsahuje cyklus. Tudíž se mezi nějakými dvěma vrcholy dá jít po více než jedné cestě, a proto můžeme přinejmenším jednu hranu z kostry odebrat. Protože každá hrana má kladné ohodnocení, snížili jsme tím součet hodnot všech hran. To je

spor. □

**Důsledek 1.2.7.** Minimální kostra ohodnoceného stromu je s ním tožná.

Minimální kostra je zvláště užitečná právě při návrhů elektrických obvodů, kdy je potřeba zařídit, aby všechna připojená zařízení čerpala co nejmenší množství energie. Protože elektřina proudí rychlostí světla, délka kabelu (pokud není zrovna mezigalaktický) nás příliš netrápí, ale právě odpor či kvalita/vodivost konkrétních spojů by mohly.

Další praktickou grafovou úlohou vedoucí na problém nalezení minimální kostry je potřeba spojit vzdálené servery. Korporace mají obvykle mnoho různých serverů rozmístěných po světě, jež spolu ale musejí sdílet data. Problém je v tom, že propojení mezi servery by nejen mělo vést k nejmenší možné prodlevě při přenosu dat (od toho **minimální**), ale nesmí ani obsahovat cykly (od toho **kostra**). Kdyby totiž cykly obsahovalo, pak by se při přenosu dat stalo, že by aspoň jeden server v tomto cyklu dostal aspoň dvakrát stejnou informaci z dvou různých zdrojů, ale v odlišný čas. Taková situace vede nezbytně dříve nebo později ke korupci dat; řekněme, když daný server už s obdrženou informací začal po přijetí provádět výpočet. Pro více detailů k tomuto využití minimálních koster vizte [Spanning Tree Protocol](#).

### 1.2.2 Kruskalův algoritmus

Na problém nalezení minimální kostry souvislého ohodnoceného grafu existuje skoro až zázračně přímočarý algoritmus, pojmenovaný po americkém matematiku, Josephu B. Kruskalovi. Jeho základní myšlenkou je prostě začít s grafem  $K$  obsahujícím všechny vrcholy z  $V$  a přidávat hrany od těch s nejnižším ohodnocením po ty s nejvyšším tak dlouho, dokud nevznikne souvislý graf. Je potřeba pouze dávat pozor na cykly. K tomu stačí si pamatovat stromy (jako množiny vrcholů), které přidáváním hran vytváříme, a povolit přidání hrany jedině v případě, že spojuje dva různé stromy.

Pro zápis v pseudokódu vizte [algoritmus 1](#). Manim s průběhem algoritmu s náhodně vygenerovaným hodnocením je k dispozici [zde](#).

**Algoritmus 1:** Kruskalův algoritmus.

---

**input** : souvislý ohodnocený graf  $G = (V, E, w)$ , kde  $V = \{v_1, \dots, v_n\}$   
**output**: množina hran  $E'$  minimální kostry grafu  $G$

---

```

1 Inicializace;
2  $E' \leftarrow \emptyset$ ;
3 Množina hran, které nelze přidat (jinak by vznikl cyklus);
4  $X \leftarrow \emptyset$ ;
5 for  $i \leftarrow 1$  to  $n$  do
6   Každý strom nejprve obsahuje pouze jediný vrchol;
7    $T_i \leftarrow \{v_i\}$ ;
8 Množina indexů pro pamatování sloučených stromů;
9  $I \leftarrow \{1, \dots, n\}$ ;
10 Přidávám hrany, dokud mám pořád víc než jeden strom;
11 while  $\#I > 1$  do
12    $e \leftarrow$  libovolná hrana s minimální  $w(e)$ , která není v  $E'$  ani v  $X$ ;
13    $i \leftarrow$  index v  $I$  takový, že  $s(e) \in T_i$ ;
14    $j \leftarrow$  index v  $I$  takový, že  $t(e) \in T_j$ ;
15   if  $i = j$  then
16     Hrana spojuje vrcholy ve stejném stromě, jejím přidáním by
17     vznikl cyklus;
18      $X \leftarrow X \cup \{e\}$ ;
19   else
20     Hrana spojuje různé stromy. Přidávám ji do kostry;
21      $E' \leftarrow E' \cup \{e\}$ ;
22      $T_i \leftarrow T_i \cup T_j$ ;
23      $I \leftarrow I \setminus \{j\}$ ;
23 return  $E'$ ;

```

---

**Tvrzení 1.2.8.** Kruskalův algoritmus je korektní.

**Důkaz.** Potřebujeme ověřit, že

- (1) algoritmus provede pouze konečný počet kroků;
- (2) algoritmus vrátí správnou odpověď.

Případ (1) je zřejmý, protože  $\#E < \infty$ , čili algoritmus přidá do  $E'$  pouze

konečně mnoho hran. Navíc, graf  $G$  je z předpokladu souvislý, a tedy vždy existuje hrana  $e$  spojující dva různé stromy  $T_i$  a  $T_j$ .

V případě (2) uvažme, že  $K = (V, E', w)$  není minimální kostra  $G$ . V takovém případě se mohlo stát, že

- (a)  $K$  není strom – tedy buď není souvislý nebo obsahuje cyklus;
- (b) existuje podmnožina  $E'' \subseteq E$  taková, že  $K' = (V, E'', w)$  je strom a

$$\sum_{e \in E''} w(e) < \sum_{e \in E'} w(e).$$

Případ (a) lze vyloučit snadno, neboť souvislost  $K$  plyne ihned ze souvislosti  $G$ , tedy, jak již bylo řečeno v bodě (1), vždy lze nalézt hranu spojující do té doby dva různé stromy. Pokud  $K$  obsahuje cyklus, tak algoritmus musel v jednom kroku spojit hranou dva vrcholy ze stejného stromu, což je spor.

Pokud nastal případ (b), pak musejí existovat hrany  $e'' \in E'' \setminus E'$  a  $e' \in E' \setminus E''$  takové, že  $w(e'') < w(e')$ . Protože algoritmus zkouší přidávat hrany vždy počínaje těmi s nejmenší vahou, musel v nějakém kroku narazit na hranu  $e''$  a zavrhnout ji. To ovšem znamená, že hrana  $e''$  spojila dva různé vrcholy téhož stromu a graf  $K' = (V, E'', w)$  obsahuje cyklus. To je spor s předpokladem, že  $K'$  je strom. Tedy, taková podmnožina  $E'' \subseteq E$  nemůže existovat a  $K$  je vskutku minimální kostra  $G$ .  $\square$

**Výstraha.** Minimální kostra grafu **není jednoznačně určena!** Všimněte si, že na řádku 12 **Kruskalova algoritmu** volím **libovolnou** hranu, která má ze všech zatím nepřidaných nejnižší váhu a jejímž přidáním nevznikne cyklus.

**Poznámka.** V **definici minimální kostry** požadujeme, aby  $G$  byl souvislý graf. Pokud tomu však tak není, je pořád možné zkonstruovat minimální kostru pro každou část  $G$ , která souvislá je (pro každou jeho tzv. *komponentu souvislosti*), zkrátka tím, že **Kruskalův algoritmus** spouštíme opakovaně.

### 1.3 Jordanovo centrum

V návaznosti na [sekcí o minimální kostře](#) se rozhovoříme o jednom dalším optimalizačním problému – konkrétně hledání „centra“ ohodnoceného grafu.

Motivační úlohou je tzv. *facility location problem*, v přibližném překladu *úloha umístění střediska*. Jde o úlohu, kdy máte danu dopravní síť sídlišť (obecně obydlených zón) a význačných uzlů, přes které se chtít nechtít musí jezdit (například velké křižovatky, Nuselák apod.). Hrany vedou mezi sídlišti či uzly, když od jednoho k druhému vede bezprostřední cesta (tedy cesta neprocházející žádným jiným sídlištěm nebo uzlem).

Nejprve si však rozmyslíme, že onen *facility location problem*, odteď FLP, má v závislosti na typu střediska dvě různá znění. V branži se jim občas říká *emergency facility location problem* (EFLP), který zahrnuje určení vhodné polohy například právě nemocnic a poliklinik, a *service facility location problem* (SFLP), sestávající z úloh nalezení nejdostupnějšího místa pro školy či obchodní centra, kde není nezbytně nutné, aby se z **každé** pokryté oblasti dalo ke středisku dostat efektivně (bo nejde o život), ale aby naopak průměr délek cest od všech obydlených oblastí ke středisku byl co nejmenší.

K formulaci obou úloh ale potřebujeme zavést základní pojem *vzdálenosti* mezi vrcholy v ohodnoceném grafu. Pro podrobnější výklad o cestách, souvislosti grafu, vzdálenosti a jejich relevanci k pojmu *metriky* na množině vizte [dodatek A.1](#).

V zájmu strohosti vyjádření označíme pro libovolné dva vrcholy  $u, v \in V$  grafu  $(V, E, w)$  symbolem  $\mathcal{P}(u, v)$  množinu všech cest mezi  $u$  a  $v$ . Speciálně,  $\mathcal{P}(v, v) = \{v\}$ , čili cesta z vrcholu do něj samého obsahuje pouze tento jeden vrchol, a  $\mathcal{P}(u, v) = \emptyset$ , pokud mezi  $u$  a  $v$  nevede v  $G$  cesta.

**Definice 1.3.1 (Vzdálenost v grafu).** Ať  $G = (V, E, w)$  je ohodnocený graf a  $u, v \in V$ . *Vzdáleností* mezi  $u$  a  $v$  v grafu  $G$ , značenou  $d_G(u, v)$  (z angl. **d**istance) definujeme jako

$$d_G(u, v) := \begin{cases} \min_{P \in \mathcal{P}(u, v)} w(P), & \text{pokud } \mathcal{P}(u, v) \neq \emptyset; \\ \infty, & \text{pokud } \mathcal{P}(u, v) = \emptyset. \end{cases}$$



Lidsky řečeno, vzdáleností mezi vrcholy je váha nejkratší cesty mezi nimi vedoucí, pokud taková existuje.

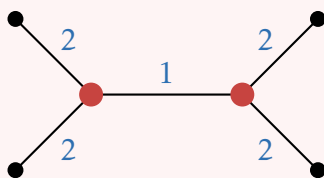
Abychom našli pro daný graf  $G = (V, E, w)$  řešení EFLP, musíme najít takový vrchol, který minimalizuje největší možnou vzdálenost od něj ke všem ostatním vrcholům  $G$ . Takový vrchol (nebo vrcholy?) nazveme *Jordanovým centrem* grafu  $G$ , po žabožroutím počtáři, Marie E. C. Jordanovi.

Formálně, *excentricita* vrcholu  $v \in V$  je kvantita  $e(v) := \max_{u \in V} d_G(v, u)$ , tedy maximum přes všechny vzdálenosti od něj k ostatním vrcholům. Toto číslo vyjadřuje, jak moc je vrchol vzdálen od „ideálního centra“ grafu, tedy od bodu, od kterého by každý vrchol byl stejně daleko. Samozřejmě, toto ideální centrum málokdy existuje, takže hledáme pouze vrchol s nejmenší excentricitou, s nejmenší *odchylkou* od centra.

**Definice 1.3.2 (Emergency Facility Location Problem).** Ať  $G = (V, E, w)$  je **souvislý** ohodnocený graf. Úlohu nalézt vrchol s minimální excentricitou nazveme EFLP. Jejím *řešením* je vrchol s touto vlastností, tedy vrchol  $c \in V$  splňující

$$e(c) = \min_{v \in V} e(v).$$

**Výstraha.** Řešení EFLP **není jednoznačně určeno!** Vizte např. graf na obrázku 13.



Obrázek 13: **Vrcholy s minimální excentricitou** v grafu  $G = (V, E, w)$ .

**Definice 1.3.3 (Jordanovo centrum).** Množinu všech řešení EFLP pro graf  $G = (V, E, w)$  nazýváme *Jordanovým centrem* grafu  $G$ .

**Definice 1.3.4 (Poloměr grafu).** Je-li  $c$  vrchol v Jordanově centru grafu  $G = (V, E, w)$ , pak hodnotu  $e(c)$  nazýváme *poloměrem* grafu  $G$  a zna-

číme ji  $\rho(G)$ .

**Poznámka.** V [definici EFLP](#) jsme požadovali, aby byl graf souvislý. To z ryze technického hlediska není nutné, protože excentricita vrcholu je definována i pro nesouvislý graf. Uvědomme si ale, že pro nesouvislý graf je excentricita každého vrcholu rovna  $\infty$ , tedy Jordanovým centrem je celý graf a úloha poněkud pozbývá smyslu.

Obdobným způsobem si formalizujeme i SFLP a později si ukážeme souvislost obou úloh s minimální kostrou.

## **A | Dodatky k teorii grafů**

### **A.1 Souvislost grafu a metrika**