

Matematika III – 7. přednáška

Teorie grafů – základní pojmy

Michal Bulant

Masarykova univerzita
Fakulta informatiky

11. 11. 2009

Obsah přednášky

- 1 Základní pojmy teorie grafů
 - Dva příklady
 - Základní definice
 - Galerie jednoduchých grafů
- 2 (Iso)Morfismy grafů a podgrafy
 - Morfismy
 - Podgrafy
 - Stupně uzlů a skóre grafu
- 3 Algoritmy a reprezentace grafů
 - Grafové algoritmy
- 4 Prohledávání v grafech
 - Prohledávání do šířky a do hloubky

Doporučené zdroje

- Martin Panák, Jan Slovák, Drsná matematika, e-text.
- *Předmětové záložky v IS MU*

Doporučené zdroje

- Martin Panák, Jan Slovák, Drsná matematika, e-text.
- *Předmětové záložky v IS MU*
- Jiří Matoušek, Jaroslav Nešetřil, Kapitoly z diskretní matematiky, Univerzita Karlova v Praze, Karolinum, Praha, 2000, 377 s.
- Petr Hliněný, Teorie grafů, studijní materiály,
<http://www.fi.muni.cz/~hlineny/Vyuka/TG.html>

Doporučené zdroje

- Martin Panák, Jan Slovák, Drsná matematika, e-text.
- *Předmětové záložky v IS MU*
- Jiří Matoušek, Jaroslav Nešetřil, Kapitoly z diskrétní matematiky, Univerzita Karlova v Praze, Karolinum, Praha, 2000, 377 s.
- Petr Hliněný, Teorie grafů, studijní materiály,
<http://www.fi.muni.cz/~hlineny/Vyuka/TG.html>
- Donald E. Knuth, The Stanford GraphBase, ACM, New York, 1993
(<http://www-cs-faculty.stanford.edu/~knuth/sgb.html>).

Doporučené zdroje

- Martin Panák, Jan Slovák, Drsná matematika, e-text.
- *Předmětové záložky v IS MU*
- Jiří Matoušek, Jaroslav Nešetřil, Kapitoly z diskrétní matematiky, Univerzita Karlova v Praze, Karolinum, Praha, 2000, 377 s.
- Petr Hliněný, Teorie grafů, studijní materiály,
<http://www.fi.muni.cz/~hlineny/Vyuka/TG.html>
- Donald E. Knuth, The Stanford GraphBase, ACM, New York, 1993
(<http://www-cs-faculty.stanford.edu/~knuth/sgb.html>).

Plán přednášky

- 1 Základní pojmy teorie grafů
 - Dva příklady
 - Základní definice
 - Galerie jednoduchých grafů
- 2 (Iso)Morfismy grafů a podgrafy
 - Morfismy
 - Podgrafy
 - Stupně uzlů a skóre grafu
- 3 Algoritmy a reprezentace grafů
 - Grafové algoritmy
- 4 Prohledávání v grafech
 - Prohledávání do šířky a do hloubky

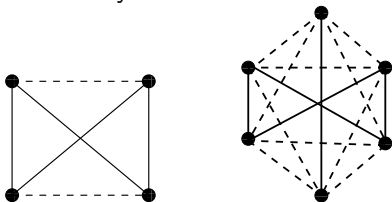
Příklad

Na večírku se někteří návštěvníci po dvojicích znají a jiné dvojice se naopak neznají. Kolik lidí musíme pozvat, abychom zaručili, že se alespoň tři hosté budou buď navzájem znát nebo neznát?

Příklad

Na večírku se někteří návštěvníci po dvojicích znají a jiné dvojice se naopak neznají. Kolik lidí musíme pozvat, abychom zaručili, že se alespoň tři hosté budou buď navzájem znát nebo neznát?

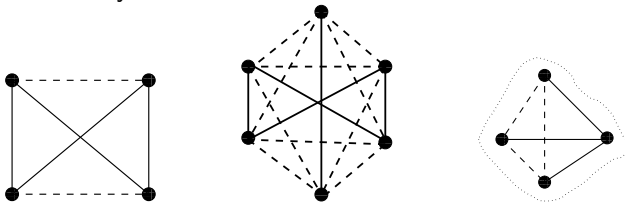
Představíme situaci pomocí obrázku. Puntíky nám představí jednotlivé hosty, plnou čarou spojíme ty dvojice, které se znají, čárkovanou ty ostatní. Naše tvrzení pak zní: při jakém počtu puntíků vždy najdeme trojúhelník, jehož strany jsou buď všechny plné nebo všechny čárkované?



Příklad

Na večírku se někteří návštěvníci po dvojicích znají a jiné dvojice se naopak neznají. Kolik lidí musíme pozvat, abychom zaručili, že se alespoň tři hosté budou buď navzájem znát nebo neznát?

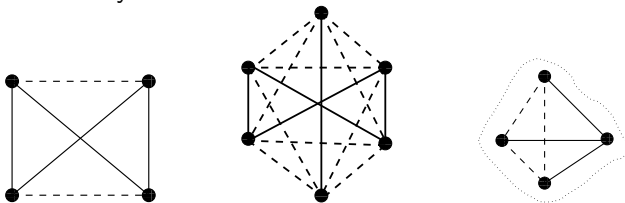
Představíme situaci pomocí obrázku. Puntíky nám představí jednotlivé hosty, plnou čarou spojíme ty dvojice, které se znají, čárkovanou ty ostatní. Naše tvrzení pak zní: při jakém počtu puntíků vždy najdeme trojúhelník, jehož strany jsou buď všechny plné nebo všechny čárkované?



Příklad

Na večírku se někteří návštěvníci po dvojicích znají a jiné dvojice se naopak neznají. Kolik lidí musíme pozvat, abychom zaručili, že se alespoň tři hosté budou buď navzájem znát nebo neznát?

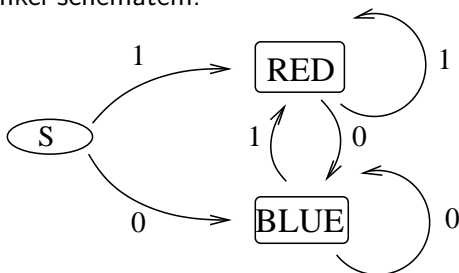
Představíme situaci pomocí obrázku. Puntíky nám představí jednotlivé hosty, plnou čarou spojíme ty dvojice, které se znají, čárkovanou ty ostatní. Naše tvrzení pak zní: při jakém počtu puntíků vždy najdeme trojúhelník, jehož strany jsou buď všechny plné nebo všechny čárkované?



Na levém obrázku takový trojúhelník není, uprostřed je. Pravý naznačuje důkaz, že existuje vždy, když počet hostů bude alespoň šest.

Představme si krabičku, která požívá jeden bit za druhým podle toho, jestli dveřmi zrovna prošel muž nebo žena – jednička nechť označuje třeba ženu. Přitom svítí buď modře nebo červeně podle toho, zda byl poslední bit nula nebo jednička (a podle barvy světla tedy poznáme, zda je za dveřmi muž nebo žena).

Znázorníme funkci schématem:



Třetí uzel, ze kterého pouze vychází dvě šipky naznačuje start před prvním zaslaným bitem.

Takovým schématům se říká *konečný automat*.

Zachytíme společné rysy předchozích příkladů.

Definice

Grafem $G = (V, E)$ rozumíme množinu V jeho **vrcholů** (někdy též uzlů) spolu s podmnožinou E množiny $\binom{V}{2}$ všech dvouprvkových podmnožin ve V . Prvkům E říkáme **hrany grafu**. Vrcholům ve hraně $e = \{v, w\}$, $v \neq w$, říkáme **hraniční vrcholy** hrany e . O hranách, které mají daný vrchol v za hraniční říkáme, že z vrcholu v **vycházejí**.

Zachytíme společné rysy předchozích příkladů.

Definice

Grafem $G = (V, E)$ rozumíme množinu V jeho **vrcholů** (někdy též uzlů) spolu s podmnožinou E množiny $\binom{V}{2}$ všech dvouprvkových podmnožin ve V . Prvkům E říkáme **hrany grafu**. Vrcholům ve hraně $e = \{v, w\}$, $v \neq w$, říkáme **hraniční vrcholy** hrany e . O hranách, které mají daný vrchol v za hraniční říkáme, že z vrcholu v **vycházejí**.

Orientovaným grafem $G = (V, E)$ rozumíme množinu V jeho vrcholů spolu s podmnožinou $E \subseteq V \times V$. Prvnímu z vrcholů definujících hranu $e = (v, w)$ říkáme **počáteční vrchol hrany**, druhému pak **koncový vrchol**.

Definice (. . . pokračování)

Hrana e **vychází** ze svého počátečního vrcholu a **vchází** do koncového. U orientovaných hran mohou být koncový a počáteční vrchol totožný, hovoříme pak o **smyčce**.

Sousední hrany grafu jsou ty, které sdílí hraniční vrchol, u **sousedních hran orientovaného grafu** musí být vrchol pro jednu koncový a pro druhou počáteční.

Naopak, **sousední vrcholy** jsou ty, které jsou hraničními pro tutéž hranu.

Definice (. . . pokračování)

Hrana e **vychází** ze svého počátečního vrcholu a **vchází** do koncového. U orientovaných hran mohou být koncový a počáteční vrchol totožný, hovoříme pak o **smyčce**.

Sousední hrany grafu jsou ty, které sdílí hraniční vrchol, u **sousedních hran orientovaného grafu** musí být vrchol pro jednu koncový a pro druhou počáteční.

Naopak, **sousední vrcholy** jsou ty, které jsou hraničními pro tutéž hranu.

Jistě umíme grafy popisovat pomocí relací, viz první kapitola prvního semestru. Jednoduchým věcem se dá říkat i složitě:

Definice (. . . pokračování)

Hrana e **vychází** ze svého počátečního vrcholu a **vchází** do koncového. U orientovaných hran mohou být koncový a počáteční vrchol totožný, hovoříme pak o **smyčce**.

Sousední hrany grafu jsou ty, které sdílí hraniční vrchol, u **sousedních hran orientovaného grafu** musí být vrchol pro jednu koncový a pro druhou počáteční.

Naopak, **sousední vrcholy** jsou ty, které jsou hraničními pro tutéž hranu.

Jistě umíme grafy popisovat pomocí relací, viz první kapitola prvního semestru. Jednoduchým věcem se dá říkat i složitě: Např. v prvním příkladu pracujeme na množině hostů se dvěma komplementárními symetrickými a antireflexními relacemi.

Grafy – příklad kompromisu mezi přirozeným sklonem
k *přemýšlením v obrázcích* a přesným matematickým vyjadřováním.

Grafy – příklad kompromisu mezi přirozeným sklonem k *přemýšlení v obrázcích* a přesným matematickým vyjadřováním. Časté použití – dodatečné informace o vrcholech nebo hranách (**obarvení** vrcholů, příp. map, **ohodnocení** hran apod.)

Grafy – příklad kompromisu mezi přirozeným sklonem k *přemýšlením v obrázcích* a přesným matematickým vyjadřováním. Časté použití – dodatečné informace o vrcholech nebo hranách (**obarvení** vrcholů, příp. map, **ohodnocení** hran apod.)

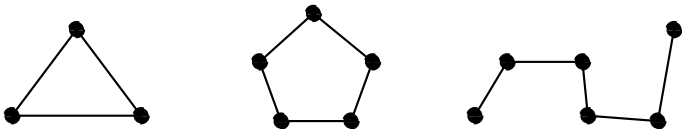
Náš první příklad můžeme tedy chápat jako graf s obarvenými hranami. Dokázané tvrzení v této řeči zní:

V grafu $K_n = (V, \binom{V}{2})$ s n vrcholy a se všemi možnými hranami obarvenými dvěma barvami, je vždy alespoň jeden trojúhelník z hran o stejné barvě, pokud (\iff) je počet vrcholů alespoň šest.

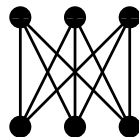
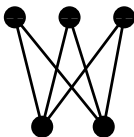
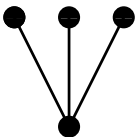
Grafu se všemi možnými hranami (tj. $E = \binom{V}{2}$) říkáme **úplný graf**. Značíme symbolem K_n , kde n je počet vrcholů grafu. Graf K_4 a K_5 jsme již viděli, K_3 je trojúhelník, K_2 je úsečka.

Grafu se všemi možnými hranami (tj. $E = \binom{V}{2}$) říkáme **úplný graf**. Značíme symbolem K_n , kde n je počet vrcholů grafu. Graf K_4 a K_5 jsme již viděli, K_3 je trojúhelník, K_2 je úsečka.

Cesta je graf, v němž existuje uspořádání vrcholů (v_0, \dots, v_n) takové, že $E = \{e_1, \dots, e_n\}$, kde $e_i = \{v_{i-1}, v_i\}$, pro všechny $i = 1, \dots, n$. Hovoříme o **cestě délky n** a značíme ji P_n . Pokud cestu upravíme tak, že poslední a první vrchol splývají, dostaneme **kružnici délky n** a značíme C_n . Na obrázku jsou $K_3 = C_3$, C_5 a P_5

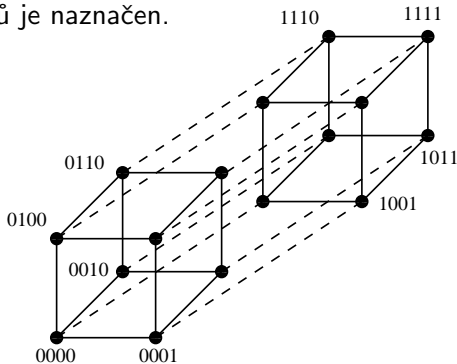


Úplný **bipartitní graf** vznikne tak, že vrcholy rozdělíme do dvou skupin (tj. obarvíme dvěma barvami) a pak přidáme všechny hrany, které spojí vrcholy z různých skupin. Značíme jej $K_{m,n}$, kde m a n jsou počty vrcholů v jednotlivých skupinách. Na obrázku je vidět $K_{1,3}$, $K_{2,3}$ a $K_{3,3}$.

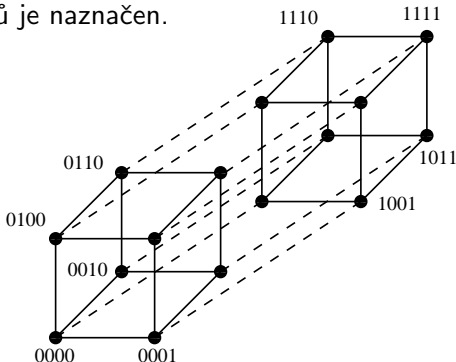


Pro $V_1 = \{u_1, \dots, u_m\}$, $V_2 = \{v_1, \dots, v_n\}$ je $K_{m,n} = (V, E)$, kde $V = V_1 \cup V_2$ a $E = V_1 \times V_2$.

Hyperkostka H_n v dimenzi n vznikne tak, že vrcholy jsou všechna čísla $0, \dots, 2^n - 1$. Hrany spojí právě ta čísla, která se v zápisu v dvojkové soustavě liší v právě jednom bitu. Na obrázku níže je H_4 a popis vrcholů je naznačen.

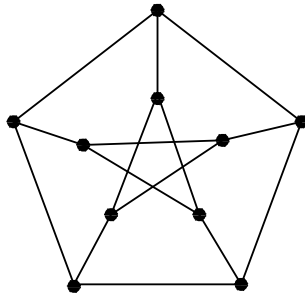
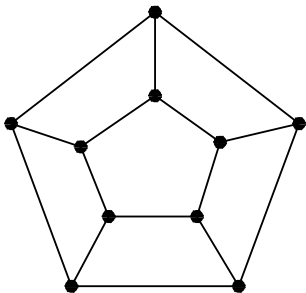


Hyperkostka H_n v dimenzi n vznikne tak, že vrcholy jsou všechna čísla $0, \dots, 2^n - 1$. Hrany spojí právě ta čísla, která se v zápisu v dvojkové soustavě liší v právě jednom bitu. Na obrázku níže je H_4 a popis vrcholů je naznačen.



Přímo z definice vyplývá, že hyperkostku v dané dimenzi vždy dostaneme tak, že vhodně spojíme hranami dvě hyperkostky o jednu dimenzi menší. Na obrázku je naznačeno spojení dvou H_3 čárkovanými hranami. Samozřejmě ale můžeme tímto způsobem rozložit H_4 mnoha různými způsoby.

Cyklický žebřík CL_n s $2n$ vrcholy je složen propojením dvou kopií kružnice C_n tak, že hrany spojí odpovídající vrcholy dle pořadí. Tzv. **Petersenův graf** je sice docela podobný CL_5 , ale ve skutečnosti je to nejjednodušší „vyvraceč nesprávných úvah“ – graf, na němž se vyplatí testovat tvrzení, než je začneme dokazovat.



Plán přednášky

- 1 Základní pojmy teorie grafů
 - Dva příklady
 - Základní definice
 - Galerie jednoduchých grafů
- 2 (Iso)Morfismy grafů a podgrafy
 - Morfismy
 - Podgrafy
 - Stupně uzlů a skóre grafu
- 3 Algoritmy a reprezentace grafů
 - Grafové algoritmy
- 4 Prohledávání v grafech
 - Prohledávání do šířky a do hloubky

Definice

Pro grafy $G = (V, E)$ a $G' = (V', E')$ budeme za (homo)morfismus $f : G \rightarrow G'$ považovat zobrazení $f_V : V \rightarrow V'$ mezi množinami vrcholů takové, že je-li $e = \{v, w\}$ hrana v E , pak $e' = \{f(v), f(w)\}$ musí být hranou v E' . V dalším textu nebudeme ve značení odlišovat morfismus f a zobrazení f_V . Zároveň pak takové zobrazení f_V určuje i zobrazení $f_E : E \rightarrow E'$, $f(e) = e'$, kde e a e' jsou jako výše.

Pro orientované grafy je definice shodná, jen pracujeme s uspořádanými dvojicemi $e = (v, w)$ v roli hran.

Definice

Pro grafy $G = (V, E)$ a $G' = (V', E')$ budeme za (homo)morfismus $f : G \rightarrow G'$ považovat zobrazení $f_V : V \rightarrow V'$ mezi množinami vrcholů takové, že je-li $e = \{v, w\}$ hrana v E , pak $e' = \{f(v), f(w)\}$ musí být hranou v E' . V dalším textu nebudeme ve značení odlišovat morfismus f a zobrazení f_V . Zároveň pak takové zobrazení f_V určuje i zobrazení $f_E : E \rightarrow E'$, $f(e) = e'$, kde e a e' jsou jako výše.

Pro orientované grafy je definice shodná, jen pracujeme s uspořádanými dvojicemi $e = (v, w)$ v roli hran.

Všimněme si, že tato definice znamená, že pokud $f(v) = f(w)$ pro dva různé vrcholy ve V , pak mezi nimi nesměla být hrana.

U orientovaných grafů je taková hrana přípustná, pokud je na společném obrazu smyčka.

Speciálním případem je morfismus libovolného grafu G do úplného grafu K_m . Takový morfismus je ekvivalentní vybranému obarvení vrcholů grafu V pomocí m různých jmen vrcholů K_m tak, že stejně obarvené vrcholy nejsou spojeny hranou. Hovoříme v tomto případě o **barvení grafu** pomocí m barev.

Příklad

Určete počet morfismů grafu P_2 do K_5 .

Definice

V případě, že je morfismus $f : G \rightarrow G'$ bijekcí na vrcholech takovou, že $i \circ f^{-1}$ je morfismem, hovoříme o **izomorfismu** grafů.

Izomorfní grafy se liší pouze různým pojmenováním vrcholů.

Definice

V případě, že je morfismus $f : G \rightarrow G'$ bijekcí na vrcholech takovou, že f^{-1} je morfismem, hovoříme o **izomorfismu** grafů.

Izomorfní grafy se liší pouze různým pojmenováním vrcholů. Snadno umíme načrtnout až na izomorfismus všechny grafy na málo vrcholech (třeba třech nebo čtyřech). Obecně jde ale o nesmírně složitý kombinatorický problém a i rozhodnutí o konkrétních dvou daných grafech, zda jsou izomorfní, je obecně mimořádně obtížné.

Jednoduchými a mimořádně užitečnými příklady morfismů grafů jsou pojmy **cesta**, **sled** a **kružnice** v grafu:

Definice

Cestou délky n v grafu G rozumíme morfismus $p : P_n \rightarrow G$ takový, že p je injektivní zobrazení (tj. všechny obrazy vrcholů v_0, \dots, v_n z P_n jsou různé). **Sled délky** n v grafu G je jakýkoliv morfismus $s : P_n \rightarrow G$ (tj. v obrazu se mohou opakovat vrcholy i hrany). **Tah** je speciální případ sledu, v němž se mohou opakovat vrcholy, ale nikoliv hrany.

Jednoduchými a mimořádně užitečnými příklady morfismů grafů jsou pojmy **cesta**, **sled** a **kružnice** v grafu:

Definice

Cestou délky n v grafu G rozumíme morfismus $p : P_n \rightarrow G$ takový, že p je injektivní zobrazení (tj. všechny obrazy vrcholů v_0, \dots, v_n z P_n jsou různé). **Sled délky** n v grafu G je jakýkoliv morfismus $s : P_n \rightarrow G$ (tj. v obrazu se mohou opakovat vrcholy i hrany). **Tah** je speciální případ sledu, v němž se mohou opakovat vrcholy, ale nikoliv hrany.

Sled si můžeme představit jako dráhu *příčinlivého, ale tápajícího* poutníka z uzlu $f(v_0)$ do uzlu $f(v_n)$. Poutník totiž zdárně dojde, ale klidně se po cestě grafem vrací do uzlů nebo i dokonce po hranách, kterými dříve šel. *Táhnoucí* poutník je již o něco moudřejší. Cesta je naopak průchod grafem z počátečního uzlu $f(v_0)$ do koncového $f(v_n)$ bez takových zbytečných oklik.

Obrazy cest i sledů jsou příkladem tzv. **podgrafů**:

Obrazy cest i sledů jsou příkladem tzv. **podgrafů**:

Definice

Graf $G' = (V', E')$ je podgrafem v grafu $G = (V, E)$, jestliže $V' \subseteq V, E' \subseteq E$.

Obrazy cest i sledů jsou příkladem tzv. **podgrafů**:

Definice

Graf $G' = (V', E')$ je podgrafem v grafu $G = (V, E)$, jestliže $V' \subseteq V, E' \subseteq E$.

Speciální příklady: Uvažujme graf $G = (V, E)$ a nějakou podmnožinu $V' \subseteq V$. **Indukovaný podgraf** je graf $G' = (V', E')$, kde $e \in E$ patří i do E' právě, když oba krajní vrcholy hrany e patří do V' ($E' = E \cap \binom{V'}{2}$). **Faktor** $G' = (V, E')$ je takový graf, který má stejnou množinu vrcholů jako G , ale jeho množina hran E' je libovolnou podmnožinou. Obecný případ je kombinací těchto dvou.

Obrazy cest i sledů jsou příkladem tzv. **podgrafů**:

Definice

Graf $G' = (V', E')$ je podgrafem v grafu $G = (V, E)$, jestliže $V' \subseteq V$, $E' \subseteq E$.

Speciální příklady: Uvažujme graf $G = (V, E)$ a nějakou podmnožinu $V' \subseteq V$. **Indukovaný podgraf** je graf $G' = (V', E')$, kde $e \in E$ patří i do E' právě, když oba krajní vrcholy hrany e patří do V' ($E' = E \cap \binom{V'}{2}$). **Faktor** $G' = (V, E')$ je takový graf, který má stejnou množinu vrcholů jako G , ale jeho množina hran E' je libovolnou podmnožinou. Obecný případ je kombinací těchto dvou.

Věta

Každý obraz homomorfismu (tj. obraz jak vrcholů, tak hran) tvoří podgraf.

Neizomorfních grafů nemůže být méně než

$$k(n) = \frac{2^{\binom{n}{2}}}{n!}.$$

Neizomorfních grafů nemůže být méně než

$$k(n) = \frac{2^{\binom{n}{2}}}{n!}.$$

Pro $n \rightarrow \infty$ lze přímo odvodit

$$\log_2 k(n) = \frac{1}{2}n^2 - O(n \log_2 n).$$

Neizomorfních grafů nemůže být méně než

$$k(n) = \frac{2^{\binom{n}{2}}}{n!}.$$

Pro $n \rightarrow \infty$ lze přímo odvodit

$$\log_2 k(n) = \frac{1}{2}n^2 - O(n \log_2 n).$$

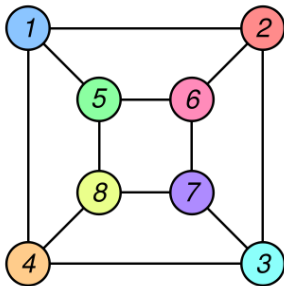
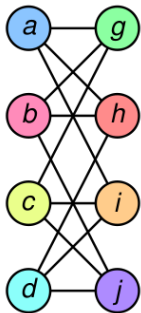
Můžeme to nepřesně formulovat tak, že velká většina všech možných grafů bude po dvou neizomorfní.

Poznámka

Vlastní konstrukce „mnoha“ neizomorfních grafů na n vrcholech ale není triviální. Zkuste jich sestavit alespoň více než n^2 !

Poznámka (Graph isomorphism problem)

GI problém je poměrně zvláštním příslušníkem třídy **NP**-problémů – není o něm známo ani, je-li **NP**-úplný, ani je-li polynomiální složitosti. Naproti tomu, o **Subgraph isomorphism problem** je známo, že je **NP**-úplný.



Stupně uzlů a skóre grafu

Izomorfní grafy se od sebe liší pouze přejmenováním vrcholů. Proto musí mít stejné všechny číselné charakteristiky, které se přečíslováním vrcholů nemění. Jednoduché údaje tohoto typu můžeme dostat sledováním počtů hran vycházejících z jednotlivých vrcholů.

Stupně uzlů a skóre grafu

Izomorfní grafy se od sebe liší pouze přejmenováním vrcholů. Proto musí mít stejné všechny číselné charakteristiky, které se přečíslováním vrcholů nemění. Jednoduché údaje tohoto typu můžeme dostat sledováním počtů hran vycházejících z jednotlivých vrcholů.

Definice

Pro vrchol $v \in V$ v grafu $G = (V, E)$ říkáme, že jeho **stupeň** je k , jestliže v E existuje k hran, jejichž hraničním vrcholem je v .

Píšeme v takovém případě $\deg v = k$.

U orientovaných grafů rozlišujeme **vstupní stupeň** $\deg_+ v$ vrcholu v a **výstupní stupeň** $\deg_- v$.

Stupně uzlů a skóre grafu

Izomorfní grafy se od sebe liší pouze přejmenováním vrcholů. Proto musí mít stejné všechny číselné charakteristiky, které se přečíslováním vrcholů nemění. Jednoduché údaje tohoto typu můžeme dostat sledováním počtů hran vycházejících z jednotlivých vrcholů.

Definice

Pro vrchol $v \in V$ v grafu $G = (V, E)$ říkáme, že jeho **stupeň** je k , jestliže v E existuje k hran, jejichž hraničním vrcholem je v .

Píšeme v takovém případě $\deg v = k$.

U orientovaných grafů rozlišujeme **vstupní stupeň** $\deg_+ v$ vrcholu v a **výstupní stupeň** $\deg_- v$.

Skóre grafu G s vrcholy $V = (v_1, \dots, v_n)$ je posloupnost

$$(\deg v_1, \deg v_2, \dots, \deg v_n)$$

Pro izomorfní grafy se jejich skóre může lišit pouze permutací hodnot. Pokud tedy porovnáme skóre grafů seříděné podle velikosti hodnot, pak různá skóre zaručují neizomorfnost grafu.

Pro izomorfní grafy se jejich skóre může lišit pouze permutací hodnot. Pokud tedy porovnáme skóre grafů setříděné podle velikosti hodnot, pak různá skóre zaručují neizomorfnost grafu. Naopak ale snadno najdeme příklad grafů se stejným skóre, které izomorfní být nemohou, např. $G = C_3 \cup C_3$ má skóre $(2, 2, 2, 2, 2, 2)$, stejně jako C_6 . Zjevně ale izomorfní nejsou, protože v C_6 existuje cesta délky 5, která v druhém grafu být nemůže.

Pro izomorfní grafy se jejich skóre může lišit pouze permutací hodnot. Pokud tedy porovnáme skóre grafů setříděné podle velikosti hodnot, pak různá skóre zaručují neizomorfnost grafu. Naopak ale snadno najdeme příklad grafů se stejným skóre, které izomorfní být nemohou, např. $G = C_3 \cup C_3$ má skóre $(2, 2, 2, 2, 2, 2)$, stejně jako C_6 . Zjevně ale izomorfní nejsou, protože v C_6 existuje cesta délky 5, která v druhém grafu být nemůže.

Jaká skóre mohou grafy mít?

Pro izomorfní grafy se jejich skóre může lišit pouze permutací hodnot. Pokud tedy porovnáme skóre grafů setříděné podle velikosti hodnot, pak různá skóre zaručují neizomorfnost grafu. Naopak ale snadno najdeme příklad grafů se stejným skóre, které izomorfní být nemohou, např. $G = C_3 \cup C_3$ má skóre $(2, 2, 2, 2, 2, 2)$, stejně jako C_6 . Zjevně ale izomorfní nejsou, protože v C_6 existuje cesta délky 5, která v druhém grafu být nemůže.

Jaká skóre mohou grafy mít?

Protože každá hrana vychází ze dvou vrcholů, musí být v celkovém součtu skóre započtena každá hrana dvakrát. Proto platí

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

Zejména tedy musí být součet všech hodnot skóre sudý.

Následující věta je vlastně o návod, jak pro dané skóre buď zjistit, že graf s takovým neexistuje nebo takový graf sestrojít.

Věta (Havel, Hakimi – Algoritmus na sestrojení grafu s daným skóre)

Pro libovolná přirozená čísla $0 \leq d_1 \leq \dots \leq d_n$ existuje graf G na n vrcholech s těmito hodnotami skóre tehdy a jen tehdy, když existuje graf se skóre

$$(d_1, d_2, \dots, d_{n-d_n} - 1, d_{n-d_n+1} - 1, \dots, d_{n-1} - 1)$$

na $n - 1$ vrcholech.

Příklad

Existuje graf, jehož skóre je $(2, 3, 3, 3, 3, 3, 4, 5)$? Pokud ano, sestrojte jej.

Důkaz.

„ \Leftarrow “ zřejmé.

„ \Rightarrow “ idea: ukáže se, že při pevně zadaném (vzestupném) skóre (d_1, \dots, d_n) existuje graf s tímto skóre, jehož vrchol v_n je spojen hranou právě s posledními d_n vrcholy $v_{n-d_n}, \dots, v_{n-1}$. □

Plán přednášky

- 1 Základní pojmy teorie grafů
 - Dva příklady
 - Základní definice
 - Galerie jednoduchých grafů
- 2 (Iso)Morfismy grafů a podgrafy
 - Morfismy
 - Podgrafy
 - Stupně uzlů a skóre grafu
- 3 Algoritmy a reprezentace grafů
 - Grafové algoritmy
- 4 Prohledávání v grafech
 - Prohledávání do šířky a do hloubky

Grafy jsou jazykem, ve kterém často formulujeme algoritmy.

Grafy jsou jazykem, ve kterém často formulujeme algoritmy. Rozumíme tím postup, kdy v nějakém orientovaném grafu přecházíme z uzlu do uzlu podél hran a přitom zpracováváme informace, které jsou určeny a ovlivněny:

- výsledkem předchozích operací,

Grafy jsou jazykem, ve kterém často formulujeme algoritmy. Rozumíme tím postup, kdy v nějakém orientovaném grafu přecházíme z uzlu do uzlu podél hran a přitom zpracováváme informace, které jsou určeny a ovlivněny:

- výsledkem předchozích operací,
- uzlem, ve kterém se zrovna nacházíme

Grafy jsou jazykem, ve kterém často formulujeme algoritmy. Rozumíme tím postup, kdy v nějakém orientovaném grafu přecházíme z uzlu do uzlu podél hran a přitom zpracováváme informace, které jsou určeny a ovlivněny:

- výsledkem předchozích operací,
- uzlem, ve kterém se zrovna nacházíme
- a hranou, kterou jsme do uzlu vstoupili.

Grafy jsou jazykem, ve kterém často formulujeme algoritmy. Rozumíme tím postup, kdy v nějakém orientovaném grafu přecházíme z uzlu do uzlu podél hran a přitom zpracováváme informace, které jsou určeny a ovlivněny:

- výsledkem předchozích operací,
- uzlem, ve kterém se zrovna nacházíme
- a hranou, kterou jsme do uzlu vstoupili.

Grafy jsou jazykem, ve kterém často formulujeme algoritmy. Rozumíme tím postup, kdy v nějakém orientovaném grafu přecházíme z uzlu do uzlu podél hran a přitom zpracováváme informace, které jsou určeny a ovlivněny:

- výsledkem předchozích operací,
- uzlem, ve kterém se zrovna nacházíme
- a hranou, kterou jsme do uzlu vstoupili.

Při zpracování informace se zároveň rozhodujeme, kterými výstupními hranami budeme pokračovat a v jakém pořadí.

Grafy jsou jazykem, ve kterém často formulujeme algoritmy. Rozumíme tím postup, kdy v nějakém orientovaném grafu přecházíme z uzlu do uzlu podél hran a přitom zpracováváme informace, které jsou určeny a ovlivněny:

- výsledkem předchozích operací,
- uzlem, ve kterém se zrovna nacházíme
- a hranou, kterou jsme do uzlu vstoupili.

Při zpracování informace se zároveň rozhodujeme, kterými výstupními hranami budeme pokračovat a v jakém pořadí. Pokud je graf neorientovaný, můžeme všechny hrany považovat za dvojice hran orientované opačnými směry.

Abychom mohli algoritmy realizovat pomocí počítače, je třeba umět uvažovaný graf efektivně zadat. Uvedeme dva příklady:

Abychom mohli algoritmy realizovat pomocí počítače, je třeba umět uvažovaný graf efektivně zadat. Uvedeme dva příklady:

Definice (**Hranový seznam**/Edge List)

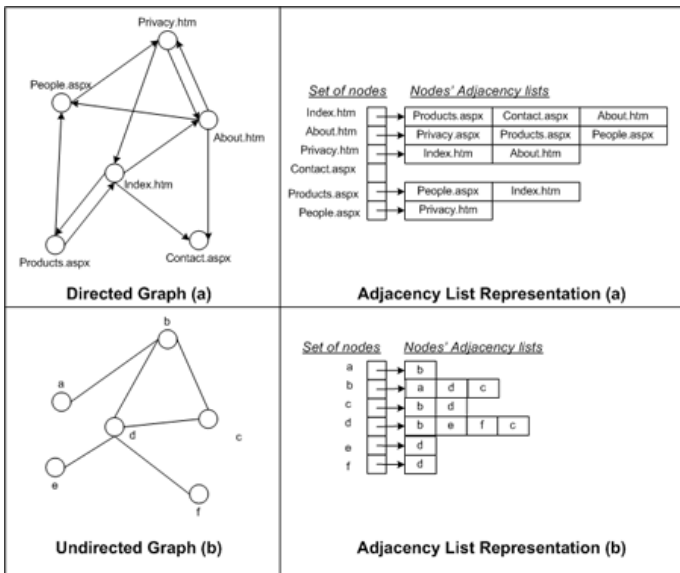
Graf $G = (V, E)$ si v něm reprezentujeme jako dva seznamy V a E propojené ukazateli tak, že každý vrchol ukazuje na všechny z něj vycházející hrany (případně také na všechny do něj vcházející hrany u orientovaných grafů) a každá hrana ukazuje na svůj počáteční a koncový vrchol.

Abychom mohli algoritmy realizovat pomocí počítače, je třeba umět uvažovaný graf efektivně zadat. Uvedeme dva příklady:

Definice (**Hranový seznam**/Edge List)

Graf $G = (V, E)$ si v něm reprezentujeme jako dva seznamy V a E propojené ukazateli tak, že každý vrchol ukazuje na všechny z něj vycházející hrany (případně také na všechny do něj vcházející hrany u orientovaných grafů) a každá hrana ukazuje na svůj počáteční a koncový vrchol.

Je vidět, že paměť potřebná na uchování grafu je v tomto případě $O(|V| + |E|)$, protože na každou hranu ukazujeme právě dvakrát a na každý vrchol ukazujeme tolikrát, kolik je jeho stupeň a součet stupňů je také roven dvojnásobku počtu hran. Až na konstantní násobek jde tedy stále o optimální způsob uchovávání grafu v paměti.



Definice (Matice sousednosti grafu)

Uvažme (neorientovaný) graf $G = (V, E)$, zvolme uspořádání jeho vrcholů $V = (v_1, \dots, v_n)$ a definujme matici $A_G = (a_{ij})$ nad \mathbb{Z}_2 (tj. zaplněnou jen nulami a jedničkami) takto:

$$a_{ij} = \begin{cases} 1 & \text{jestliže je hrana } e_{ij} = \{v_i, v_j\} \in E \\ 0 & \text{jestliže není hrana } e_{ij} = \{v_i, v_j\} \in E \end{cases}$$

Matici A_G nazýváme matice sousednosti grafu G .

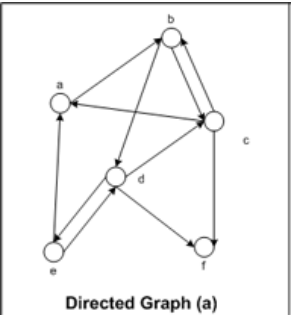
Definice (Matice sousednosti grafu)

Uvažme (neorientovaný) graf $G = (V, E)$, zvolme uspořádání jeho vrcholů $V = (v_1, \dots, v_n)$ a definujme matici $A_G = (a_{ij})$ nad \mathbb{Z}_2 (tj. zaplněnou jen nulami a jedničkami) takto:

$$a_{ij} = \begin{cases} 1 & \text{jestliže je hrana } e_{ij} = \{v_i, v_j\} \in E \\ 0 & \text{jestliže není hrana } e_{ij} = \{v_i, v_j\} \in E \end{cases}$$

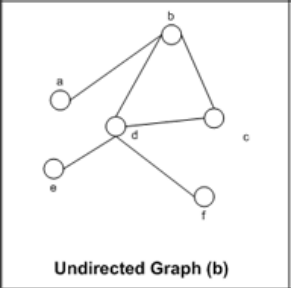
Matici A_G nazýváme matice sousednosti grafu G .

Zadání grafu pomocí matice sousednosti potřebuje vždy $O(n^2)$ místa v paměti. Pokud je ale v grafu málo hran, dostáváme tzv. řídkou matici se skoro všemi prvky nulovými. Takové lze naopak reprezentovat pomocí *hranových seznamů* odpovídajících grafů a to i včetně obecných číselných hodnot pro jednotlivé hrany. Příklad pro procvičení – násobení matic pomocí reprezentace hranovým seznamem.



Adjacency Matrix Representation (a)

	a	b	c	d	e	f
a		✓				
b			✓	✓		
c		✓				✓
d			✓		✓	✓
e	✓			✓		
f						



Adjacency Matrix Representation (b)

	a	b	c	d	e	f
a		✓				
b	✓		✓	✓		
c		✓		✓		
d		✓	✓		✓	✓
e				✓		
f				✓		

Definice (Základní operace nad grafem)

- *odebrání hrany*
- *přidání hrany*
- *přidání vrcholu*
- *odebrání vrcholu*
- *dělení hrany nově přidaným vrcholem*

Jak se projeví tyto operace v našich reprezentacích?

Jednoduchou aplikací maticového počtu je tvrzení:

Věta

Nechť $G = (V, E)$ je graf s uspořádanými vrcholy $V = (v_1, \dots, v_n)$ a maticí sousednosti A_G . Označme $A_G^k = (a_{ij}^{(k)})$ prvky k -té mocniny matice A_G . Pak $a_{ij}^{(k)}$ je počet sledů délky k mezi vrcholy v_i a v_j .

Důkaz.

Indukcí. □

Jednoduchou aplikací maticového počtu je tvrzení:

Věta

Nechť $G = (V, E)$ je graf s uspořádanými vrcholy $V = (v_1, \dots, v_n)$ a maticí sousednosti A_G . Označme $A_G^k = (a_{ij}^{(k)})$ prvky k -té mocniny matice A_G . Pak $a_{ij}^{(k)}$ je počet sledů délky k mezi vrcholy v_i a v_j .

Důkaz.

Indukcí. □

Důsledek

Jsou-li $G = (V, E)$ a A_G jako v předchozí větě, pak lze všechny dvojice vrcholů G spojit cestou právě tehdy, když má matice $(A + \mathbb{I}_n)^{n-1}$ samé nenulové členy (zde \mathbb{I}_n označuje jednotkovou matici s n řádky a sloupci).

Plán přednášky

- 1 Základní pojmy teorie grafů
 - Dva příklady
 - Základní definice
 - Galerie jednoduchých grafů
- 2 (Iso)Morfismy grafů a podgrafy
 - Morfismy
 - Podgrafy
 - Stupně uzlů a skóre grafu
- 3 Algoritmy a reprezentace grafů
 - Grafové algoritmy
- 4 Prohledávání v grafech
 - Prohledávání do šířky a do hloubky

Prohledávání v grafech

Algoritmy bývají založeny na postupném prohledávání všech vrcholů v grafu. Zpravidla máme zadaný počáteční vrchol nebo si jej na začátku procesu zvolíme. V průběhu procesu pak v každém okamžiku jsou vrcholy

- *již zpracované*, tj. ty, kterými jsme již při běhu algoritmu prošli a definitivně je zpracovali;
- *aktivní*, tj. ty vrcholy, které jsou detekovány a připraveny pro zpracování;
- *spící*, tj. ty vrcholy, na které teprve dojde.

Prohledávání v grafech

Algoritmy bývají založeny na postupném prohledávání všech vrcholů v grafu. Zpravidla máme zadaný počáteční vrchol nebo si jej na začátku procesu zvolíme. V průběhu procesu pak v každém okamžiku jsou vrcholy

- *již zpracované*, tj. ty, kterými jsme již při běhu algoritmu prošli a definitivně je zpracovali;
- *aktivní*, tj. ty vrcholy, které jsou detekovány a připraveny pro zpracování;
- *spící*, tj. ty vrcholy, na které teprve dojde.

Zároveň si udržujeme přehled o již zpracovaných hranách. V každém okamžiku musí být množiny vrcholů a /nebo hran v těchto skupinách disjunktním rozdělením množin vrcholů V a množin E hran grafu G a některý z aktivních vrcholů je aktuálně zpracováván.

Základní postup:

- Na počátku máme jeden aktivní vrchol a všechny ostatní vrcholy jsou spící.

Základní postup:

- Na počátku máme jeden aktivní vrchol a všechny ostatní vrcholy jsou spící.
- V prvním kroku projdeme všechny hrany vycházející z aktivního vrcholu a jejich příslušným koncovým vrcholům, které jsou spící, změním stav na aktivní.

Základní postup:

- Na počátku máme jeden aktivní vrchol a všechny ostatní vrcholy jsou spící.
- V prvním kroku projdeme všechny hrany vycházející z aktivního vrcholu a jejich příslušným koncovým vrcholům, které jsou spící, změním stav na aktivní.
- V dalších krocích vždy u zpracovávaného vrcholu probíráme ty z něho vycházející hrany, které dosud nebyly probrány a jejich koncové vrcholy přidáváme mezi aktivní. Tento postup aplikujeme stejně u orientovaných i neorientovaných grafů, jen se drobně mění význam adjektiv koncový a počáteční u vrcholů.

Základní postup:

- Na počátku máme jeden aktivní vrchol a všechny ostatní vrcholy jsou spící.
- V prvním kroku projdeme všechny hrany vycházející z aktivního vrcholu a jejich příslušným koncovým vrcholům, které jsou spící, změním stav na aktivní.
- V dalších krocích vždy u zpracovávaného vrcholu probíráme ty z něho vycházející hrany, které dosud nebyly probrány a jejich koncové vrcholy přidáváme mezi aktivní. Tento postup aplikujeme stejně u orientovaných i neorientovaných grafů, jen se drobně mění význam adjektiv koncový a počáteční u vrcholů.

V konkrétních úlohách se můžeme omezovat na některé z hran, které vychází z aktuálního vrcholu. Na principu to ale nic podstatného nemění.

Pro realizaci algoritmů je nutné se rozhodnout, v jakém pořadí zpracováváme aktivní vrcholy a v jakém pořadí zpracováváme hrany z nich vycházející. V zásadě přichází v úvahu dvě možnosti zpracovávání vrcholů:

- 1 vrcholy vybíráme pro další zpracování ve stejném pořadí, jak se stávaly aktivními (fronta – FIFO)
- 2 dalším vrcholem vybraným pro zpracování je poslední zaktivněný vrchol (zásobník – LIFO).

V prvním případě hovoříme o **prohledávání do šířky**, ve druhém o **prohledávání do hloubky**.

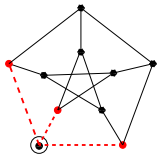
Na první pohled je zřejmá role volby vhodných datových struktur pro uchování údajů o grafu. Hranový seznam umožňuje projít všechny hrany vycházející z právě zpracovávaného vrcholu v čase lineárně úměrném jejich počtu. Každou hranu přitom diskutujeme nejvýše dvakrát, protože má právě dva konce. Zjevně tedy platí:

Na první pohled je zřejmá role volby vhodných datových struktur pro uchování údajů o grafu. Hranový seznam umožňuje projít všechny hrany vycházející z právě zpracovávaného vrcholu v čase lineárně úměrném jejich počtu. Každou hranu přitom diskutujeme nejvýše dvakrát, protože má právě dva konce. Zjevně tedy platí:

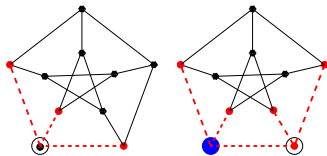
Věta

*Celkový čas realizace vyhledávání do šířky i do hloubky je $O((n + m) * K)$, kde n je počet vrcholů v grafu, m je počet hran v grafu a K je čas potřebný na zpracování jedné hrany, resp. jednoho vrcholu.*

Ilustrace prohledávání do šířky:

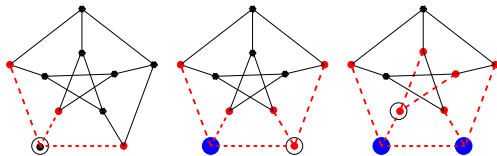


Ilustrace prohledávání do šířky:



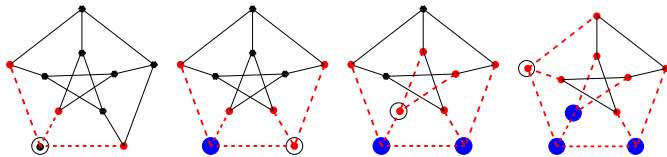
Zakroužkovaný vrchol je ten právě zpracováváný, modré velké puntíky jsou již zpracované uzly, čárkované červené hrany jsou již zpracované a červené drobné uzly jsou ty aktivní (poznají se také podle toho, že do nich již vede některá zpracovaná hrana). Hrany zpracováváme v pořadí orientace proti hodinovým ručkám, přičemž za *první* bereme směr *kolmo dolů*.

Ilustrace prohledávání do šířky:



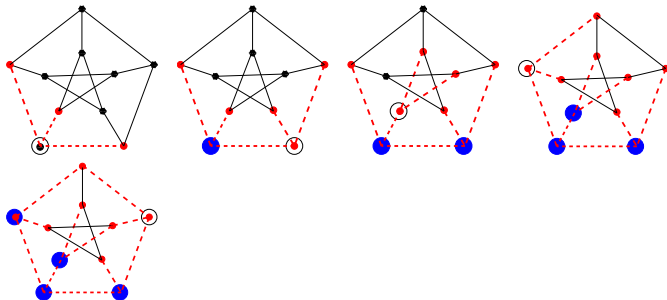
Zakroužkovaný vrchol je ten právě zpracováváný, modré velké puntíky jsou již zpracované uzly, čárkované červené hrany jsou již zpracované a červené drobné uzly jsou ty aktivní (poznají se také podle toho, že do nich již vede některá zpracovaná hrana). Hrany zpracováváme v pořadí orientace proti hodinovým ručkám, přičemž za *první* bereme směr *kolmo dolů*.

Ilustrace prohledávání do šířky:



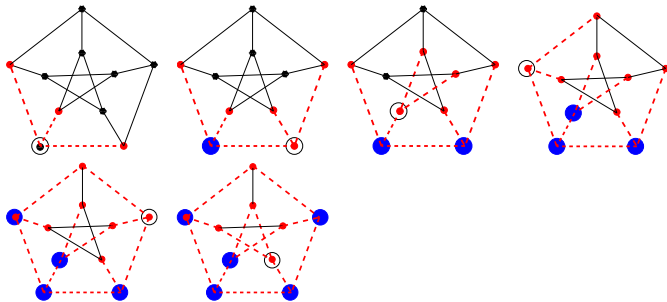
Zakroužkovaný vrchol je ten právě zpracováváný, modré velké puntíky jsou již zpracované uzly, čárkované červené hrany jsou již zpracované a červené drobné uzly jsou ty aktivní (poznají se také podle toho, že do nich již vede některá zpracovaná hrana). Hrany zpracováváme v pořadí orientace proti hodinovým ručkám, přičemž za *první* bereme směr *kolmo dolů*.

Ilustrace prohledávání do šířky:



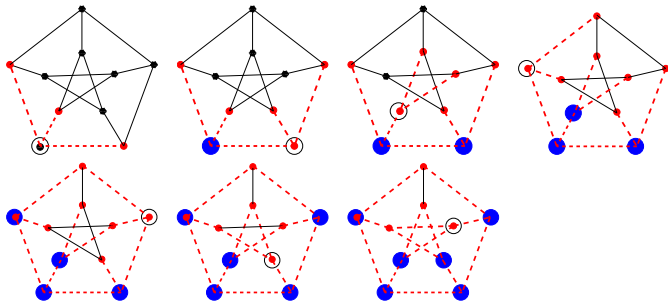
Zakroužkovaný vrchol je ten právě zpracovávaný, modré velké puntíky jsou již zpracované uzly, čárkované červené hrany jsou již zpracované a červené drobné uzly jsou ty aktivní (poznají se také podle toho, že do nich již vede některá zpracovaná hrana). Hrany zpracováváme v pořadí orientace proti hodinovým ručkám, přičemž za *první* bereme směr *kolmo dolů*.

Ilustrace prohledávání do šířky:



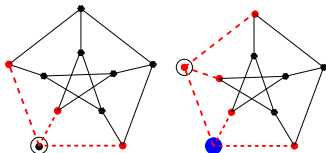
Zakroužkovaný vrchol je ten právě zpracováváný, modré velké puntíky jsou již zpracované uzly, čárkované červené hrany jsou již zpracované a červené drobné uzly jsou ty aktivní (poznají se také podle toho, že do nich již vede některá zpracovaná hrana). Hrany zpracováváme v pořadí orientace proti hodinovým ručkám, přičemž za *první* bereme směr *kolmo dolů*.

Ilustrace prohledávání do šířky:

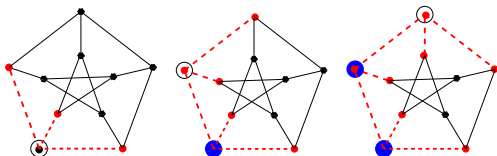


Zakroužkovaný vrchol je ten právě zpracováváný, modré velké puntíky jsou již zpracované uzly, čárkované červené hrany jsou již zpracované a červené drobné uzly jsou ty aktivní (poznají se také podle toho, že do nich již vede některá zpracovaná hrana). Hrany zpracováváme v pořadí orientace proti hodinovým ručkám, přičemž za *první* bereme směr *kolmo dolů*.

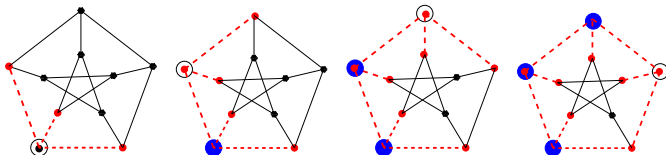
Totéž postupem *do hloubky*. Všimněte si, že první krok je stejný jako v předchozím případě.



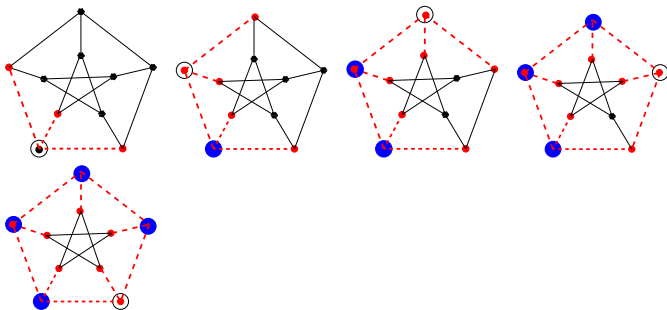
Totéž postupem *do hloubky*. Všimněte si, že první krok je stejný jako v předchozím případě.



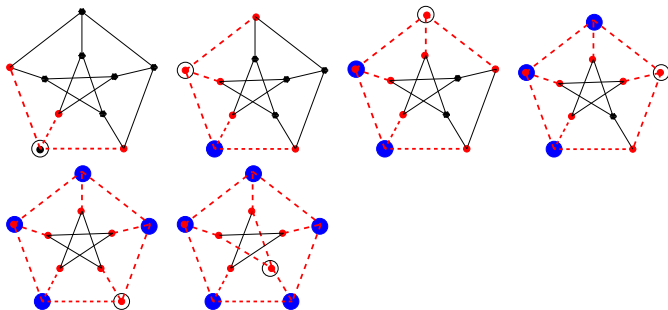
Totéž postupem *do hloubky*. Všimněte si, že první krok je stejný jako v předchozím případě.



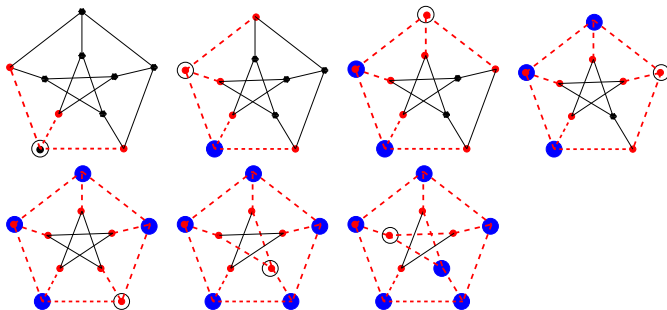
Totéž postupem *do hloubky*. Všimněte si, že první krok je stejný jako v předchozím případě.



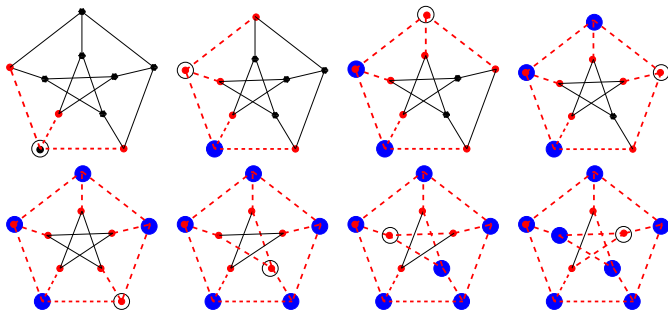
Totéž postupem *do hloubky*. Všimněte si, že první krok je stejný jako v předchozím případě.



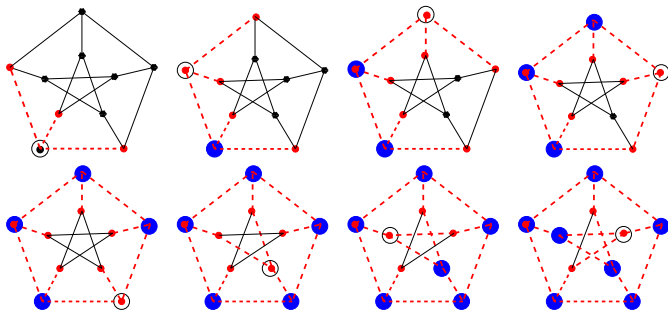
Totéž postupem *do hloubky*. Všimněte si, že první krok je stejný jako v předchozím případě.



Totéž postupem *do hloubky*. Všimněte si, že první krok je stejný jako v předchozím případě.



Totéž postupem *do hloubky*. Všimněte si, že první krok je stejný jako v předchozím případě.



Prohledávání „do hloubky“ odpovídá interpretaci rekurze v běžných imperativních jazycích (např. i v Maplu) – v těchto případech je ale *stavový graf* potenciálně nekonečný a prohledávání do hloubky tak samozřejmě nemusí nalézt všechny vrcholy dostupné z daného vrcholu po cestách konečné délky (narozdíl od prohledávání do šířky).

Prohledávání bludiště

Průchod bludištěm (s křídou) prostřednictvím prohledávání do hloubky v neorientovaném grafu – úkolem je najít východ nebo dokázat, že neexistuje:

- každou chodbu při prvním průchodu označíme čarou, při návratu přidáme druhou čáru

Prohledávání bludiště

Průchod bludištěm (s křídou) prostřednictvím prohledávání do hloubky v neorientovaném grafu – úkolem je najít východ nebo dokázat, že neexistuje:

- každou chodbu při prvním průchodu označíme čarou, při návratu přidáme druhou čáru
- při východu z místnosti preferujeme neoznačenou chodbu

Prohledávání bludiště

Průchod bludištěm (s křídou) prostřednictvím prohledávání do hloubky v neorientovaném grafu – úkolem je najít východ nebo dokázat, že neexistuje:

- každou chodbu při prvním průchodu označíme čarou, při návratu přidáme druhou čáru
- při východu z místnosti preferujeme neoznačenou chodbu
- navštívenou místnost označíme, pokud je již označená, tak se ihned vrátíme stejnou chodbou zpět

Prohledávání bludiště

Průchod bludištěm (s křídou) prostřednictvím prohledávání do hloubky v neorientovaném grafu – úkolem je najít východ nebo dokázat, že neexistuje:

- každou chodbu při prvním průchodu označíme čarou, při návratu přidáme druhou čáru
- při východu z místnosti preferujeme neoznačenou chodbu
- navštívenou místnost označíme, pokud je již označená, tak se ihned vrátíme stejnou chodbou zpět
- z dosud nenavštívené místnosti postupujeme dále

Prohledávání bludiště

Průchod bludištěm (s křídou) prostřednictvím prohledávání do hloubky v neorientovaném grafu – úkolem je najít východ nebo dokázat, že neexistuje:

- každou chodbu při prvním průchodu označíme čarou, při návratu přidáme druhou čáru
- při východu z místnosti preferujeme neoznačenou chodbu
- navštívenou místnost označíme, pokud je již označená, tak se ihned vrátíme stejnou chodbou zpět
- z dosud nenavštívené místnosti postupujeme dále
- jsou-li již všechny chodby vedoucí z místnosti použity, vracíme se zpět chodbou, která je označena jen jednou čarou

Prohledávání bludiště

Průchod bludištěm (s křídou) prostřednictvím prohledávání do hloubky v neorientovaném grafu – úkolem je najít východ nebo dokázat, že neexistuje:

- každou chodbu při prvním průchodu označíme čarou, při návratu přidáme druhou čáru
- při východu z místnosti preferujeme neoznačenou chodbu
- navštívenou místnost označíme, pokud je již označená, tak se ihned vrátíme stejnou chodbou zpět
- z dosud nenavštívené místnosti postupujeme dále
- jsou-li již všechny chodby vedoucí z místnosti použity, vracíme se zpět chodbou, která je označena jen jednou čarou
- pokud z dané místnosti vedou pouze chodby, které jsou označeny 2 čarami, hledání ukončíme (nutně jde o výchozí místnost, východ neexistuje a můžeme s klidem umřít).