

Matematika III – 9. přednáška

Nejkratší cesty, eulerovské grafy, stromy, rovinné grafy

Michal Bulant

Masarykova univerzita
Fakulta informatiky

25. 11. 2009

Obsah přednášky

- 1 Hledání nejkratších cest
 - Dijkstrův algoritmus pro hledání nejkratších cest
 - Floyd-Warshallův algoritmus
- 2 Eulerovské grafy a hamiltonovské kružnice
- 3 Stromy
- 4 Rovinné grafy
 - Platónská tělesa
 - Barvení map

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 diskrétní matematiky, Univerzita Karlova v Praze, Karolinum, Praha, 2000, 377 s.
- Petr Hliněný, Teorie grafů, studijní materiály, <http://www.fi.muni.cz/~hlineny/Vyuka/GT/>

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/GT/>
- 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/GT/>
- Donald E. Knuth, The Stanford GraphBase, ACM, New York, 1993
(<http://www-cs-faculty.stanford.edu/~knuth/sgb.html>).

Plán přednášky

- 1 Hledání nejkratších cest
 - Dijkstrův algoritmus pro hledání nejkratších cest
 - Floyd-Warshallův algoritmus
- 2 Eulerovské grafy a hamiltonovské kružnice
- 3 Stromy
- 4 Rovinné grafy
 - Platónská tělesa
 - Barvení map

Dijkstrův algoritmus

Nejkratší cestu v grafu, která vychází z daného uzlu v a končí v jiném uzlu w můžeme hledat pomocí prohledávání grafu do šířky. Při tomto typu prohledávání totiž postupně diskutujeme vrcholy, do kterých se umíme dostat z výchozího vrcholu po jediné hraně, poté projdeme všechny, které mají vzdálenost nejvýše 2 atd. Na této jednoduché úvaze je založen jeden z nejpoužívanějších grafových algoritmů – tzv. **Dijkstrův algoritmus**.

Dijkstrův algoritmus

Nejkratší cestu v grafu, která vychází z daného uzlu v a končí v jiném uzlu w můžeme hledat pomocí prohledávání grafu do šířky. Při tomto typu prohledávání totiž postupně diskutujeme vrcholy, do kterých se umíme dostat z výchozího vrcholu po jediné hraně, poté projdeme všechny, které mají vzdálenost nejvýše 2 atd. Na této jednoduché úvaze je založen jeden z nejpoužívanějších grafových algoritmů – tzv. **Dijkstrův algoritmus**.

Tento algoritmus hledá nejkratší cesty za (reálného) předpokladu, kdy jednotlivé hrany e jsou ohodnoceny *vzdálenostmi*, tj. kladnými reálnými čísly $w(e)$. Kromě aplikace na hledání vzdáleností v silničních nebo jiných sítích to mohou být také výnosy, toky v sítích atd.

- Vstupem algoritmu je graf $G = (V, E)$ s ohodnocením hran a počáteční vrchol v_0 .

- Vstupem algoritmu je graf $G = (V, E)$ s ohodnocením hran a počáteční vrchol v_0 .
- Výstupem je ohodnocení vrcholů čísla $d_w(v)$, která udávají nejmenší možný součet ohodnocení hran podél cest z vrcholu v_0 do vrcholu v .

Všimněte si, že místo původní úlohy (nalézt nejkratší cestu mezi **dvěma** vrcholy) řešíme i něco navíc – nalezneme nejkratší cestu z v do **všech** vrcholů.

Postup dobře funguje v orientovaných i neorientovaných grafech.

- Vstupem algoritmu je graf $G = (V, E)$ s ohodnocením hran a počáteční vrchol v_0 .
- Výstupem je ohodnocení vrcholů čísla $d_w(v)$, která udávají nejmenší možný součet ohodnocení hran podél cest z vrcholu v_0 do vrcholu v .

Všimněte si, že místo původní úlohy (nalézt nejkratší cestu mezi **dvěma** vrcholy) řešíme i něco navíc – nalezneme nejkratší cestu z v do **všech** vrcholů.

Postup dobře funguje v orientovaných i neorientovaných grafech. Je skutečně podstatné, že všechna naše ohodnocení jsou kladná. Zkusme si rozmyslet třeba cestu P_3 se záporně ohodnocenou prostřední hranou. Při procházení sledu mezi krajními vrcholy bychom *vzdálenost* zmenšovali každým prodloužením sledu o průchod prostřední hranou tam a zpět.

Dijkstrův algoritmus vyžaduje jen drobnou modifikaci obecného prohledávání do šířky:

- U každého vrcholu v budeme po celý chod algoritmu udržovat číselnou hodnotu $d(v)$, která bude horním odhadem skutečné vzdálenosti vrcholu v od vrcholu v_0 .

Dijkstrův algoritmus vyžaduje jen drobnou modifikaci obecného prohledávání do šířky:

- U každého vrcholu v budeme po celý chod algoritmu udržovat číselnou hodnotu $d(v)$, která bude horním odhadem skutečné vzdálenosti vrcholu v od vrcholu v_0 .
- Množina již zpracovaných vrcholů bude v každém okamžiku obsahovat ty vrcholy, u kterých již nejkratší cestu známe, tj. $d(v) = d_w(v)$.

Dijkstrův algoritmus vyžaduje jen drobnou modifikaci obecného prohledávání do šířky:

- U každého vrcholu v budeme po celý chod algoritmu udržovat číselnou hodnotu $d(v)$, která bude horním odhadem skutečné vzdálenosti vrcholu v od vrcholu v_0 .
- Množina již zpracovaných vrcholů bude v každém okamžiku obsahovat ty vrcholy, u kterých již nejkratší cestu známe, tj. $d(v) = d_w(v)$.
- Do množiny aktivních (právě zpracovávaných) vrcholů W zařadíme vždy právě ty vrcholy y z množiny spících vrcholů Z , pro které je $d(y) = \min\{d(z); z \in Z\}$.

Dijkstrův algoritmus

Předpokládáme, že graf G má alespoň dva vrcholy.

- *Inicializační krok*: Nastavíme hodnoty u všech $v \in V$,

$$d(v) = \begin{cases} 0 & \text{pro } v = v_0 \\ \infty & \text{pro } v \neq v_0, \end{cases}$$

nastavíme $Z = V$, $W = \emptyset$.

Dijkstrův algoritmus

Předpokládáme, že graf G má alespoň dva vrcholy.

- *Inicializační krok*: Nastavíme hodnoty u všech $v \in V$,

$$d(v) = \begin{cases} 0 & \text{pro } v = v_0 \\ \infty & \text{pro } v \neq v_0, \end{cases}$$

nastavíme $Z = V$, $W = \emptyset$.

- *Test cyklu*: Pokud není ohodnocení všech vrcholů $y \in Z$ rovno ∞ , pokračujeme dalším krokem, v opačném případě algoritmus končí.

- *Aktualizace stavu vrcholů:*

- Najdeme množinu N všech vrcholů $v \in Z$, pro které $d(v)$ nabývá nejmenší možné hodnoty

$$\delta = \min\{d(y); y \in Z\};$$

- posledně zpracované aktivní vrcholy W přesuneme do množiny zpracovaných a za nové aktivní vrcholy zvolíme $W = N$ a odebereme je ze spících, tj. množina spících bude nadále $Z \setminus N$.

- *Aktualizace stavu vrcholů:*

- Najdeme množinu N všech vrcholů $v \in Z$, pro které $d(v)$ nabývá nejmenší možné hodnoty

$$\delta = \min\{d(y); y \in Z\};$$

- posledně zpracované aktivní vrcholy W přesuneme do množiny zpracovaných a za nové aktivní vrcholy zvolíme $W = N$ a odebereme je ze spících, tj. množina spících bude nadále $Z \setminus N$.
- *Tělo hlavního cyklu:* Pro všechny hrany v množině E_{WZ} všech hran vycházejících z některého aktivního vrcholu v a končících ve spícím vrcholu y opakujeme:
 - Vybereme dosud nezpracovanou hranu $e\{x, y\} \in E_{WZ}$;
 - Pokud je $d(x) + w(e) < d(y)$, nahradíme $d(y)$ touto menší hodnotou.

Věta

Pro všechny vrcholy v ležící v souvislé komponentě vrcholu s najde Dijkstraův algoritmus vzdálenosti $d_w(v)$. Vrcholy ostatních souvislých komponent zůstanou ohodnoceny $d(v) = \infty$. Algoritmus lze implementovat tak, že ukončí svoji práci v čase $O(n \log n + m)$, kde n je počet vrcholů a m je počet hran v grafu G .

Důkaz.

- během celého algoritmu platí $d(v) \geq d_w(v)$ pro všechna $v \in V$.

Věta

Pro všechny vrcholy v ležící v souvislé komponentě vrcholu s najde Dijkstrův algoritmus vzdálenosti $d_w(v)$. Vrcholy ostatních souvislých komponent zůstanou ohodnoceny $d(v) = \infty$. Algoritmus lze implementovat tak, že ukončí svoji práci v čase $O(n \log n + m)$, kde n je počet vrcholů a m je počet hran v grafu G .

Důkaz.

- během celého algoritmu platí $d(v) \geq d_w(v)$ pro všechna $v \in V$.
- po skončení algoritmu platí $d(v) \leq d_w(v)$ pro všechna $v \in V$ (Prostřednictvím silnějšího tvrzení: jsou-li $0 = d_1 < d_2 < \dots < d_k < \infty$ všechny různé konečné vzdálenosti vrcholů od s a označíme-li $M_i = \{v \in V; d_w(v) = d_i\}$, pak se krok *Aktualizace stavu vrcholů* provede přesně k -krát a po jeho i -tém vykonání platí $N = M_i, \delta = d_i$ a $V \setminus Z = \cup_{k=1}^i M_i$.)

Modifikace algoritmu

- Pokud nás skutečně zajímá jen nejkratší cesta do konkrétního vrcholu, pak samozřejmě výpočet ukončíme poté, co tento vrchol přejde do množiny již zpracovaných.

Modifikace algoritmu

- Pokud nás skutečně zajímá jen nejkratší cesta do konkrétního vrcholu, pak samozřejmě výpočet ukončíme poté, co tento vrchol přejde do množiny již zpracovaných.
- Můžeme využít dodatečné informace (při hledání nejkratší cesty z Brna do Prahy v silniční síti asi nepojedeme přes Ostravu) – *heuristika* $h : V \rightarrow \mathbb{R}$ splňující $w(\{x, y\}) \geq h(x) - h(y)$ pro každou hranu $\{x, y\}$.

Modifikace algoritmu

- Pokud nás skutečně zajímá jen nejkratší cesta do konkrétního vrcholu, pak samozřejmě výpočet ukončíme poté, co tento vrchol přejde do množiny již zpracovaných.
- Můžeme využít dodatečné informace (při hledání nejkratší cesty z Brna do Prahy v silniční síti asi nepojedeme přes Ostravu) – *heuristika* $h : V \rightarrow \mathbb{R}$ splňující $w(\{x, y\}) \geq h(x) - h(y)$ pro každou hranu $\{x, y\}$.

Modifikace algoritmu

- Pokud nás skutečně zajímá jen nejkratší cesta do konkrétního vrcholu, pak samozřejmě výpočet ukončíme poté, co tento vrchol přejde do množiny již zpracovaných.
- Můžeme využít dodatečné informace (při hledání nejkratší cesty z Brna do Prahy v silniční síti asi nepojedeme přes Ostravu) – *heuristika* $h : V \rightarrow \mathbb{R}$ splňující $w(\{x, y\}) \geq h(x) - h(y)$ pro každou hranu $\{x, y\}$.
Doporučení: $h(v)$ je dolní odhad vzdálenosti v do cílového vrcholu (např. vzdálenost "vzdušnou čarou"). Místo minimalizace $d(y)$ pro $y \in Z$ tak v algoritmu minimalizujeme $d(y) + h(y)$.

Další algoritmy a aplikace

Principy **Dijkstrova algoritmu** se využívají v OSPF (Open Shortest Paths First) routovacím protokolu.

Další algoritmy a aplikace

Principy **Dijkstrova algoritmu** se využívají v OSPF (Open Shortest Paths First) routovacím protokolu.

Bellman-Fordův algoritmus

- pracuje na stejném principu jako Dijkstrův; místo postupu po uzlech je zpracovává "naráz" – cyklus *relaxace* probíhá $(|V| - 1)$ krát přes všechny hrany
- připouští záporné hrany a detekuje záporné cykly
- je obvykle časově náročnější než Dijkstrův
- distribuovaná verze je (či spíše byla) používána v *distance-vector routing protocol*

Floyd-Warshallův algoritmus (*all pairs shortest paths*)

- v čase $O(n^3)$ vypočte vzdálenosti mezi všemi vrcholy
- vychází z matice A délek hran a postupně počítá matice $U_0, U_1, \dots, U_{|V|}$, kde $u_k(i, j)$ je délka nejkratší cesty z i do j , kde cesta prochází pouze vrcholy z $\{1, 2, \dots, k\}$.
- výpočet vychází ze vztahu

$$u_k(i, j) = \min\{u_{k-1}(i, j), u_{k-1}(i, k) + u_{k-1}(k, j)\}.$$

- je efektivnější než "upravená mocnina" A_G (násobení \rightarrow sčítání, sčítání \rightarrow minimum) – ta je totiž $O(n^4)$, resp. (optimalizovaná) $O(n^3 \log n)$.

Plán přednášky

- 1 Hledání nejkratších cest
 - Dijkstrův algoritmus pro hledání nejkratších cest
 - Floyd-Warshallův algoritmus
- 2 Eulerovské grafy a hamiltonovské kružnice
- 3 Stromy
- 4 Rovinné grafy
 - Platónská tělesa
 - Barvení map

Grafy jedním tahem

Jistě se každý setkal s dětskou hříčkou

Nakresli obrázek jedním tahem.

V řeči grafů to znamená *najděte sled, který projde všechny hrany právě jednou a každý vrchol alespoň jednou.*

Grafy jedním tahem

Jistě se každý setkal s dětskou hříčkou

Nakresli obrázek jedním tahem.

V řeči grafů to znamená *najděte sled, který projde všechny hrany právě jednou a každý vrchol alespoň jednou.*

Definice

Sled, který prochází právě jednou všemi hranami a začíná a končí v jednom vrcholu, se nazývá **uzavřený eulerovský tah**.

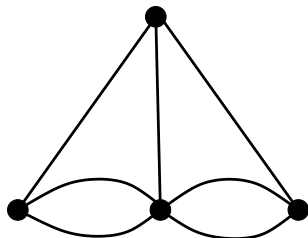
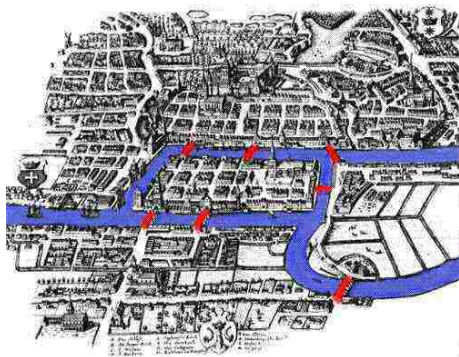
Grafům, které takový sled připouští říkáme **eulerovské**.

Hovoříme rovněž o (neuzavřeném) eulerovském tahu, kde vypouštíme požadavek na stejný výchozí a cílový vrchol.

Terminologie odkazuje na klasický příběh o sedmi mostech ve městě Královec (Königsberg, tj. Kaliningrad), které se měly projít na procházce každý právě jednou a důkaz nemožnosti takové procházky od Leonharda Eulera z roku 1736.

Terminologie odkazuje na klasický příběh o sedmi mostech ve městě Královec (Königsberg, tj. Kaliningrad), které se měly projít na procházce každý právě jednou a důkaz nemožnosti takové procházky od Leonharda Eulera z roku 1736.

Situace je znázorněna na obrázku. Nalevo dobová mapa, napravo odpovídající (multi)graf. Vrcholy tohoto grafu odpovídají *souvislé pevnině*, hrany mostům.



Kupodivu je obecné řešení takového problému dosti snadné, jak ukazuje následující věta. Samozřejmě také ukazuje, že se Euler zamýšleným způsobem procházet nemohl.

Kupodivu je obecné řešení takového problému dosti snadné, jak ukazuje následující věta. Samozřejmě také ukazuje, že se Euler zamýšleným způsobem procházet nemohl.

Věta

Graf G je eulerovský tehdy a jen tehdy, když je souvislý a všechny vrcholy v G mají sudý stupeň.

Důkaz.

Podmínka je zřejmě nutná.

Kupodivu je obecné řešení takového problému dosti snadné, jak ukazuje následující věta. Samozřejmě také ukazuje, že se Euler zamýšleným způsobem procházet nemohl.

Věta

Graf G je eulerovský tehdy a jen tehdy, když je souvislý a všechny vrcholy v G mají sudý stupeň.

Důkaz.

Podmínka je zřejmě nutná.

Dostatečnost podmínky se ukáže sporem uvážením tahu v G maximální možné délky. □

Kupodivu je obecné řešení takového problému dosti snadné, jak ukazuje následující věta. Samozřejmě také ukazuje, že se Euler zamýšleným způsobem procházet nemohl.

Věta

Graf G je eulerovský tehdy a jen tehdy, když je souvislý a všechny vrcholy v G mají sudý stupeň.

Důkaz.

Podmínka je zřejmě nutná.

Dostatečnost podmínky se ukáže sporem uvážením tahu v G maximální možné délky. □

Důsledek

Graf lze nakreslit jedním tahem právě tehdy, když má všechny stupně vrcholů sudé nebo když existují právě dva vrcholy se stupněm lichým.

Příklad

- 1 Určete nejmenší počet mostů, které je třeba v Královci přistavět, aby byl graf eulerovský.
- 2 Jaká je situace v Kaliningradu nyní (od dob Eulerových doznalo zejména působením válek město mnoho změn)? Byl by dnes schopen Euler svoji procházku realizovat?

Eulerovské orientované grafy

Definice

Orientovaný graf (V, E) nazveme *eulerovský*, jestliže v něm existuje uzavřený orientovaný tah, který obsahuje každou hranu právě jednou a každý vrchol aspoň jednou.

Orientované eulerovské grafy lze rovněž velmi dobře charakterizovat. K tomu ovšem potřebujeme některé nové pojmy.

Definice

Orientovaný graf nazveme *vyvážený*, jestliže pro každý jeho vrchol v platí $\deg_+(v) = \deg_-(v)$.

Symetrizací orientovaného grafu (V, E) nazýváme neorientovaný graf (V, \bar{E}) , kde

$$\bar{E} = \{\{x, y\}; (x, y) \in E \text{ nebo } (y, x) \in E\}.$$

Věta

Orientovaný graf G je eulerovský právě když je vyvážený a jeho symetrizace je souvislý graf (tj. graf G je slabě souvislý).

Důkaz.

Analogický jako v neorientovaném případě. □

Problém čínského poštáka (route inspection problem)

Route inspection problem je zobecněním problému nalezení eulerovského tahu. Úkolem je nalézt nejkratší sled v grafu s ohodnocenými hranami, který obsahuje každou hranu v grafu. Tento problém má v současnosti mnoho praktického využití (analýza DNA, směrování robotů, svoz odpadu, ...).

Problém čínského poštáka (route inspection problem)

Route inspection problem je zobecněním problému nalezení eulerovského tahu. Úkolem je nalézt nejkratší sled v grafu s ohodnocenými hranami, který obsahuje každou hranu v grafu. Tento problém má v současnosti mnoho praktického využití (analýza DNA, směrování robotů, svoz odpadu, ...). Zřejmě je v případě, že graf G je eulerovský, nejkratším takovým sledem příslušný eulerovský tah.

Problém čínského poštáka (route inspection problem)

Route inspection problem je zobecněním problému nalezení eulerovského tahu. Úkolem je nalézt nejkratší sled v grafu s ohodnocenými hranami, který obsahuje každou hranu v grafu.

Tento problém má v současnosti mnoho praktického využití (analýza DNA, směrování robotů, svoz odpadu, ...).

Zřejmě je v případě, že graf G je eulerovský, nejkratším takovým sledem příslušný eulerovský tah.

V opačném případě nutně graf obsahuje sudý počet vrcholů lichého stupně. Tento graf je třeba přidáváním hran doplnit na eulerovský (multi)graf (později ukážeme, že v případě stromů to znamená nutnost zdvojení všech hran). Snadno lze ukázat, že to lze udělat v polynomiálním čase jak v orientovaném, tak neorientovaném případě, v případě multigrafů je to však problém **NP-úplný**.

Hamiltonovské grafy

Obdobný požadavek na průchod grafem, ovšem tak, abychom prošli právě jednou každým vrcholem (tj. zároveň nejvýše jednou každou hranou), vede na obtížné problémy. Takový průchod grafem je realizován kružnicí, která obsahuje všechny vrcholy grafu G , hovoříme o **hamiltonovských kružnicích** v grafu G . Graf se nazývá *hamiltonovský*, jestliže má hamiltonovskou kružnici.

Hamiltonovské grafy

Obdobný požadavek na průchod grafem, ovšem tak, abychom prošli právě jednou každým vrcholem (tj. zároveň nejvýše jednou každou hranou), vede na obtížné problémy. Takový průchod grafem je realizován kružnicí, která obsahuje všechny vrcholy grafu G , hovoříme o **hamiltonovských kružnicích** v grafu G . Graf se nazývá *hamiltonovský*, jestliže má hamiltonovskou kružnici. Zatímco (zdánlivě podobně složitý) problém nalezení eulerovského tahu je triviální, zjistit, zda je daný graf hamiltonovský, je **NP-úplný problém**.

Hamiltonovské grafy

Obdobný požadavek na průchod grafem, ovšem tak, abychom prošli právě jednou každým vrcholem (tj. zároveň nejvýše jednou každou hranou), vede na obtížné problémy. Takový průchod grafem je realizován kružnicí, která obsahuje všechny vrcholy grafu G , hovoříme o **hamiltonovských kružnicích** v grafu G . Graf se nazývá *hamiltonovský*, jestliže má hamiltonovskou kružnici. Zatímco (zdánlivě podobně složitý) problém nalezení eulerovského tahu je triviální, zjistit, zda je daný graf hamiltonovský, je **NP-úplný problém**.

V praxi je ovšem problém nalezení hamiltonovské kružnice (či jeho modifikace – např. problém obchodního cestujícího) podstatou mnoha problémů v logistice, je proto často žádoucí nalezení i suboptimálního řešení (v případě problému obchodního cestujícího).

Příklad (Icosian Game – William Rowan Hamilton)

Nalezněte hamiltonovskou kružnici v grafu tvořeném vrcholy a hranami pravidelného dodekaedru (dvanáctistěnu) – viz <http://www.puzzlemuseum.com/month/picm02/200201hamilton.jpg>

Příklad

Existuje hamiltonovská kružnice v Petersenově grafu?

Příklad (Icosian Game – William Rowan Hamilton)

Nalezněte hamiltonovskou kružnici v grafu tvořeném vrcholy a hranami pravidelného dodekaedru (dvanáctistěnu) – viz <http://www.puzzlemuseum.com/month/picm02/200201hamilton.jpg>

Příklad

Existuje hamiltonovská kružnice v Petersenově grafu?

Věta (Dirac (1952))

Má-li v grafu G s $n \geq 3$ vrcholy každý vrchol stupeň alespoň $n/2$, je G hamiltonovský.

Věta (Ore (1960))

Má-li v grafu G s $n \geq 4$ vrcholy každá dvojice nesousedních vrcholů součet stupňů alespoň n , je G hamiltonovský.

Uzávěrem grafu G v této souvislosti rozumíme graf $cl(G)$, který dostaneme z G přidáním všech hran u, v takových, že u, v nejsou sousední a $\deg(u) + \deg(v) \geq n$.

Uzávěrem grafu G v této souvislosti rozumíme graf $cl(G)$, který dostaneme z G přidáním všech hran u, v takových, že u, v nejsou sousední a $\deg(u) + \deg(v) \geq n$.

Věta (Bondy, Chvátal (1972))

Graf G je hamiltonovský, právě když je $cl(G)$ hamiltonovský.

Uzávěrem grafu G v této souvislosti rozumíme graf $cl(G)$, který dostaneme z G přidáním všech hran u, v takových, že u, v nejsou sousední a $\deg(u) + \deg(v) \geq n$.

Věta (Bondy, Chvátal (1972))

Graf G je hamiltonovský, právě když je $cl(G)$ hamiltonovský.

Je vidět, že Oreho (a tedy i Diracova) věta je triviálním důsledkem této věty.

Plán přednášky

- 1 Hledání nejkratších cest
 - Dijkstrův algoritmus pro hledání nejkratších cest
 - Floyd-Warshallův algoritmus
- 2 Eulerovské grafy a hamiltonovské kružnice
- 3 **Stromy**
- 4 Rovinné grafy
 - Platónská tělesa
 - Barvení map

Definice

*Souvislý graf neobsahující kružnici, se nazývá **strom**. Obecně v grafech nazýváme vrcholy stupně jedna **listy** (případně také **koncové vrcholy**).*

Definice

*Souvislý graf neobsahující kružnici, se nazývá **strom**. Obecně v grafech nazýváme vrcholy stupně jedna **listy** (případně také **koncové vrcholy**). Obecněji, graf neobsahující kružnice nazýváme **les**.*

Tato definice nicméně není úplně nejvhodnější pro praktickou kontrolu – uvedeme proto za chvíli hned 5 ekvivalentních definic. Následující lemma ukazuje, že každý strom lze vybudovat postupně z jediného vrcholu přidáváním listů:

Definice

*Souvislý graf neobsahující kružnici, se nazývá **strom**. Obecně v grafech nazýváme vrcholy stupně jedna **listy** (případně také **koncové vrcholy**). Obecněji, graf neobsahující kružnice nazýváme **les**.*

Tato definice nicméně není úplně nejvhodnější pro praktickou kontrolu – uvedeme proto za chvíli hned 5 ekvivalentních definic. Následující lemma ukazuje, že každý strom lze vybudovat postupně z jediného vrcholu přidáváním listů:

Lemma

Každý strom s alespoň dvěma vrcholy obsahuje alespoň dva listy. Pro libovolný graf G s listem v jsou následující tvrzení ekvivalentní:

- G je strom;
- $G \setminus v$ je strom.

Charakterizace stromů

Věta

Pro každý graf $G = (V, E)$ jsou následující podmínky ekvivalentní

- 1 *G je strom;*
- 2 *pro každé dva vrcholy v, w grafu G existuje právě jedna cesta z v do w ;*
- 3 *graf G je souvislý, ale vyjmutím libovolné hrany vznikne nesouvislý graf*
- 4 *graf G neobsahuje kružnici, každým přidáním hrany do grafu G však již kružnice vznikne*
- 5 *G je souvislý graf a mezi velikostí množin jeho vrcholů a hran platí vztah (Eulerův vzorec) $|V| = |E| + 1$.*

Důkaz jednotlivých implikací obvykle vedeme indukcí podle počtu vrcholů s využitím lemmatu o výstavbě stromů.

Charakterizace stromů

Věta

Pro každý graf $G = (V, E)$ jsou následující podmínky ekvivalentní

- 1 *G je strom;*
- 2 *pro každé dva vrcholy v, w grafu G existuje právě jedna cesta z v do w ;*
- 3 *graf G je souvislý, ale vyjmutím libovolné hrany vznikne nesouvislý graf*
- 4 *graf G neobsahuje kružnici, každým přidáním hrany do grafu G však již kružnice vznikne*
- 5 *G je souvislý graf a mezi velikostí množin jeho vrcholů a hran platí vztah (Eulerův vzorec) $|V| = |E| + 1$.*

Důkaz jednotlivých implikací obvykle vedeme indukcí podle počtu vrcholů s využitím lemmatu o výstavbě stromů. Ke stromům se vrátíme později v souvislosti s praktickými aplikacemi.

Plán přednášky

- 1 Hledání nejkratších cest
 - Dijkstrův algoritmus pro hledání nejkratších cest
 - Floyd-Warshallův algoritmus
- 2 Eulerovské grafy a hamiltonovské kružnice
- 3 Stromy
- 4 Rovinné grafy
 - Platónská tělesa
 - Barvení map

Rovinné grafy

Velice často se setkáváme s grafy, které jsou nakresleny v rovině. To znamená, že každý vrchol grafu je ztotožněn s nějakým bodem v rovině a hrany mezi vrcholy v a w odpovídají spojitým křivkám $c : [0, 1] \rightarrow \mathbb{R}^2$ spojujícím vrcholy $c(0) = v$ a $c(1) = w$. Pokud navíc platí, že se jednotlivé dvojice hran protínají nejvýše v koncových vrcholech, pak hovoříme o **rovinném grafu** G .

Rovinné grafy

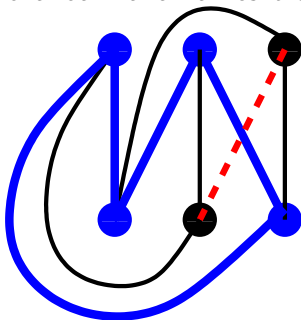
Velice často se setkáváme s grafy, které jsou nakresleny v rovině. To znamená, že každý vrchol grafu je ztotožněn s nějakým bodem v rovině a hrany mezi vrcholy v a w odpovídají spojitým křivkám $c : [0, 1] \rightarrow \mathbb{R}^2$ spojujícím vrcholy $c(0) = v$ a $c(1) = w$. Pokud navíc platí, že se jednotlivé dvojice hran protínají nejvýše v koncových vrcholech, pak hovoříme o **rovinném grafu** G . Otázka, jestli daný graf připouští realizaci (nakreslení) jako rovinný graf, vyvstává velice často v aplikacích.

Jednoduchý příklad je následující:

Tři dodavatelé vody, elektřiny a plynu mají každý své jedno přípojné místo v blízkosti tří rodinných domků. Chtějí je všichni napojit tak, aby se jejich sítě nekřížily (třeba se jim nechce kopat příliš hluboko. . .). Je to možné zvládnout? Odpověď zní *není*.

Jednoduchý příklad je následující:

Tři dodavatelé vody, elektřiny a plynu mají každý své jedno přípojné místo v blízkosti tří rodinných domků. Chtějí je všichni napojit tak, aby se jejich sítě nekřížily (třeba se jim nechce kopat příliš hluboko. . .). Je to možné zvládnout? Odpověď zní *není*. Jde o bipartitní úplný graf $K_{3,3}$, kde tři vrcholy představují přípojná místa, další tři pak domky. Hrany jsou linie sítí. Všechny hrany umíme zvládnout, jedna poslední ale už nejde, viz obrázek na kterém neumíme čárkovanou hranu nakreslit bez křížení:



Obecně se dá ukázat tzv. Kuratowského věta:

Věta

Graf G je rovinný právě tehdy když žádný jeho podgraf není izomorfní dělení grafu $K_{3,3}$ nebo grafu K_5 .

Obecně se dá ukázat tzv. Kuratowského věta:

Věta

Graf G je rovinný právě tehdy když žádný jeho podgraf není izomorfní dělení grafu $K_{3,3}$ nebo grafu K_5 .

Jedna implikace je zřejmá – dělením rovinného grafu vzniká vždy opět rovinný graf a jestliže podgraf nelze v rovině nakreslit bez křížení, totéž musí platit i pro celý graf G . Opačný směr důkazu je naopak velice složitý a nebudeme se jím zde zabývat.

Obecně se dá ukázat tzv. Kuratowského věta:

Věta

Graf G je rovinný právě tehdy když žádný jeho podgraf není izomorfní dělení grafu $K_{3,3}$ nebo grafu K_5 .

Jedna implikace je zřejmá – dělením rovinného grafu vzniká vždy opět rovinný graf a jestliže podgraf nelze v rovině nakreslit bez křížení, totéž musí platit i pro celý graf G . Opačný směr důkazu je naopak velice složitý a nebudeme se jím zde zabývat.

Problematicke rovinných grafů je věnováno ve výzkumu a aplikacích hodně pozornosti, my se zde omezíme pouze na vybrané ilustrace. Zmiňme alespoň naokraj, že existují algoritmy, které testují rovinnost grafu na n vrcholech v čase $O(n)$, což určitě nejde přímou aplikací Kuratowského věty.

Uvažme (konečný) rovinný graf G , včetně jeho nakreslení v \mathbb{R}^2 a necht' S je množina všech bodů $x \in \mathbb{R}^2$, které nepatří žádné hraně, ani nejsou vrcholem. Množina $\mathbb{R}^2 \setminus G$ se rozpadne na disjunktní souvislé podmnožiny S_i , kterým říkáme **stěny rovinného grafu** G . Jedna stěna je výjimečná – ta jejíž doplněk obsahuje všechny vrcholy grafu. Budeme jí říkat neohrazená stěna S_0 . Množinu všech stěn budeme označovat $S = \{S_0, S_1, \dots, S_k\}$ a rovinný graf $G = (V, E, S)$.

Jako příklad si můžeme rozebrat stromy. Každý strom je zjevně rovinný graf, jak je vidět například z možnosti realizovat jej postupným přidáváním listů k jedinému vrcholu. Samozřejmě také můžeme použít Kuratowského větu – když není v G žádná kružnice, nemůže obsahovat jakékoliv dělení grafů $K_{3,3}$ nebo K_5 . Protože strom G neobsahuje žádnou kružnici, dostáváme pouze jedinou stěnu S_0 a to tu neohrazenou. Protože víme, jaký je rozdíl mezi počty vrcholů a hran pro všechny stromy, dostáváme vztah

$$|V| - |E| + |S| = 2.$$

Vztah mezi počty hran, stěn a vrcholů lze odvodit pro všechny rovinné grafy. Jde o tzv. Eulerův vztah. Všimněme si, že z něho zejména vyplývá, že počet stěn v rovinném grafu nezávisí na způsobu, jaké jeho rovinné nakreslení vybereme.

Věta

Nechť $G = (V, E, S)$ je souvislý rovinný graf. Pak platí

$$|V| - |E| + |S| = 2.$$

Vztah mezi počty hran, stěn a vrcholů lze odvodit pro všechny rovinné grafy. Jde o tzv. Eulerův vztah. Všimněme si, že z něho zejména vyplývá, že počet stěn v rovinném grafu nezávisí na způsobu, jaké jeho rovinné nakreslení vybereme.

Věta

Nechť $G = (V, E, S)$ je souvislý rovinný graf. Pak platí

$$|V| - |E| + |S| = 2.$$

Důkaz.

Indukcí podle počtu hran. □

Rovinné grafy si můžeme dobře představit jako namalované na povrchu koule místo v rovině. Sféra vznikne z roviny tak, že přidáme jeden bod v *nekonečnu*. Opět můžeme stejným způsobem hovořit o stěnách a pro takovýto graf pak jsou všechny jeho stěny rovnocenné (i stěna S_0 je ohraničená).

Rovinné grafy si můžeme dobře představit jako namalované na povrchu koule místo v rovině. Sféra vznikne z roviny tak, že přidáme jeden bod v *nekonečnu*. Opět můžeme stejným způsobem hvořit o stěnách a pro takovýto graf pak jsou všechny jeho stěny rovnocenné (i stěna S_0 je ohraničená).

Naopak, každý konvexní mnohostěn $P \subset \mathbb{R}^3$ si můžeme představit jako graf nakreslený na povrchu koule. Vypuštěním jednoho bodu uvnitř jedné ze stěn (ta stane neohraničenou stěnou S_0) pak obdržíme rovinný graf jako výše.

Rovinné grafy, které vzniknou z konvexních mnohostránků, jsou zjevně 2-souvislé, protože každé dva vrcholy v konvexním mnohostránku leží na společné kružnici. Navíc v nich platí, že každá stěna kromě S_0 je vnitřkem nějaké kružnice a S_0 je vnějškem nějaké kružnice. Názorné se zdá i to, že ve skutečnosti budou grafy vznikající z konvexních mnohostránků 3-souvislé.

Rovinné grafy, které vzniknou z konvexních mnohostránků, jsou zjevně 2-souvislé, protože každé dva vrcholy v konvexním mnohostránku leží na společné kružnici. Navíc v nich platí, že každá stěna kromě S_0 je vnitřkem nějaké kružnice a S_0 je vnějškem nějaké kružnice. Názorné se zdá i to, že ve skutečnosti budou grafy vznikající z konvexních mnohostránků 3-souvislé.

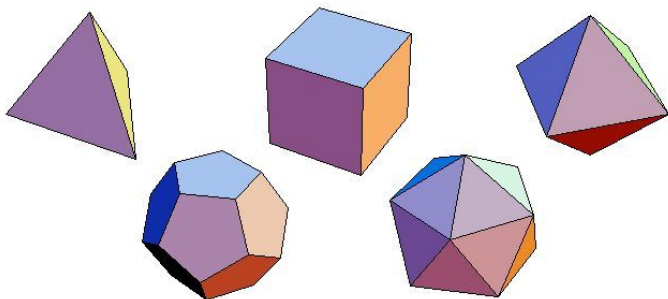
Ve skutečnosti platí dosti náročná Steinitzova věta:

Věta

Libovolný vrcholově 3-souvislý rovinný graf G vzniká z konvexního mnohostránku v \mathbb{R}^3 .

Jako ilustraci kombinatorické práce s grafy odvodíme klasifikaci tzv. pravidelných mnohostěnů, tj. mnohostěnů poskládaných ze stejných pravidelných mnohoúhelníků tak, že se jich v každém vrcholu dotýká stejný počet. Již v dobách antického myslitele Platóna se vědělo, že jich je pouze pět:

Jako ilustraci kombinatorické práce s grafy odvodíme klasifikaci tzv. pravidelních mnohostěnů, tj. mnohostěnů poskládaných ze stejných pravidelných mnohoúhelníků tak, že se jich v každém vrcholu dotýká stejný počet. Již v dobách antického myslitele Platóna se vědělo, že jich je pouze pět:



Přeložíme si požadavek pravidelnosti do vlastností příslušného grafu: chceme aby každý vrchol měl stejný stupeň $d \geq 3$ a zároveň aby na hranici každé stěny byl stejný počet $k \geq 3$ vrcholů. Označme n počet vrcholů, e počet hran a s počet stěn. Máme k dispozici jednak vztah provazující stupně vrcholů s počtem hran:

$$dn = 2e$$

Přeložíme si požadavek pravidelnosti do vlastností příslušného grafu: chceme aby každý vrchol měl stejný stupeň $d \geq 3$ a zároveň aby na hranici každé stěny byl stejný počet $k \geq 3$ vrcholů.

Označme n počet vrcholů, e počet hran a s počet stěn.

Máme k dispozici jednak vztah provazující stupně vrcholů s počtem hran:

$$dn = 2e$$

a podobně počítáme počet hran, které ohraničují jednotlivé stěny, a bereme v úvahu, že každá je hranicí dvou stěn, tj.

$$2e = ks.$$

Eulerův vztah pak říká

$$2 = n - e + s = \frac{2e}{d} - e + \frac{2e}{k}.$$

Úpravou odtud dostáváme pro naše známé d a k vztah

$$\frac{1}{d} + \frac{1}{k} = \frac{1}{2} + \frac{1}{e}.$$

Protože e a n musí být přirozená čísla (tj. zejména je $\frac{1}{e} > 0$) a minimum pro d i k je 3, dostáváme přímou diskusí všech možností tento výčet:

d	k	n	e	s
3	3	4	6	4
3	4	8	12	6
4	3	6	12	8
3	5	20	30	12
5	3	12	30	20

Tabulka zadává všechny možnosti. Ve skutečnosti ale také všechny odpovídající pravidelné mnohostěny existují - již jsme je viděli.

Maximální počet hran

Věta

Nechť (V, E, S) je rovinný graf s aspoň třemi vrcholy. Pak

$$|E| \leq 3|V| - 6.$$

Rovnost přitom nastává pro maximální rovinný graf, tj. rovinný graf, k němuž nejde při zachování rovinnosti přidat žádnou hranu. Pokud navíc uvažovaný graf neobsahuje trojúhelník (tj. K_3 jako podgraf), platí dokonce $|E| \leq 2|V| - 4$.

Maximální počet hran

Věta

Nechť (V, E, S) je rovinný graf s aspoň třemi vrcholy. Pak

$$|E| \leq 3|V| - 6.$$

Rovnost přitom nastává pro maximální rovinný graf, tj. rovinný graf, k němuž nejde při zachování rovinnosti přidat žádnou hranu. Pokud navíc uvažovaný graf neobsahuje trojúhelník (tj. K_3 jako podgraf), platí dokonce $|E| \leq 2|V| - 4$.

Důkaz.

Maximální rovinný graf má všechny stěny ohraničené kružnicí délky 3, z čehož plyne $3|S| = 2|E|$ a odtud již pomocí Eulerova vztahu dostáváme první tvrzení. Podobně v druhé části. □

Důsledek

- K_5 není rovinný;

Důsledek

- K_5 není rovinný;
- $K_{3,3}$ není rovinný;

Důsledek

- K_5 není rovinný;
- $K_{3,3}$ není rovinný;
- každý rovinný graf obsahuje alespoň jeden vrchol stupně nejvýše 5;

Důsledek

- K_5 není rovinný;
- $K_{3,3}$ není rovinný;
- každý rovinný graf obsahuje alespoň jeden vrchol stupně nejvýše 5;
- každý rovinný graf bez trojúhelníků obsahuje alespoň jeden vrchol stupně nejvýše 3.

Problém čtyř barev

Jedním z nejznámějších kombinatorických problémů je otázka:

Je možné každou mapu obarvit 4 barvami?

Tento problém sice na první pohled vypadá ryze geometricky, ale dá se přeformulovat do kombinatorické podoby.

Definice

Mapou nazýváme souvislý rovinný multigraf bez mostů. Normální mapou pak mapu, jejíž všechny vrcholy jsou stupně 3. Obarvení mapy je funkce, která každé stěně mapy přiřadí číslo (barvu).

Problém čtyř barev byl rozřešen teprve po více než sto letech bádání – mnoho matematiků na prezentovaný důkaz stále pohlíží s despektem, protože je založen na prověření velkého množství případů pomocí počítače. Elementárními kombinatorickými prostředky je možné alespoň dokázat možnost obarvení normálních map pěti barvami – viz literatura.

Věta (Appel, Haken (1976))

Každou normální mapu je možné obarvit pomocí čtyř barev.