Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE

Lukáš Jirovský

Vybrané problémy z teorie grafů ve výuce na střední škole

Katedra didaktiky matematiky

Vedoucí bakalářské práce: RNDr. Pavla Pavlíková, Ph.D.

Studijní program: Matematika, Matematika zaměřená na vzdělávání, matematika v kombinaci s informatikou

2008

Chtěl bych poděkovat RNDr. Pavle Pavlíkové, Ph.D., za pomoc při psaní práce.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 20. května 2008 Lukáš Jirovský

Obsah

Název práce: Vybrané problémy z teorie grafů ve výuce na střední škole

Autor: Lukáš Jirovský

Katedra: Katedra didaktiky matematiky

Vedoucí bakalářské práce: RNDr. Pavla Pavlíková, Ph.D.

E-mail vedoucího: pavla.pavlikova@mff.cuni.cz

Abstrakt: Práce představuje učební text zaměřený na problematiku teorie grafů v podobě webové aplikace použitelný např. pro matematický seminář na střední škole. Poukazuje na široké spektrum aplikací nejen v matematice, ale i ostatních předmětech (fyzika, chemie, geografie,...) a poskytuje matematický základ pro teorii grafů. Součástí práce je i řada příkladů na procvičení vhodných pro studenty středních škol.

Klíčová slova: teorie grafů, kostra grafu, barvení mapy, nejkratší cesta v grafu

Title: Selected problems from theory of graphs at high school teaching

Author: Lukáš Jirovský

Department: Department of Didactics of mathematics

Supervisor: RNDr. Pavla Pavlíková, Ph.D.

Supervisor's e-mail adress: pavla.pavlikova@mff.cuni.cz

Abstract: The educational website describing the wide usage of graph theory at high school teaching. It adverts to many possible applications in non-mathematical subjects (physical science, chemistry, geography…).

The work also contains many exercises and basics of graph theory.

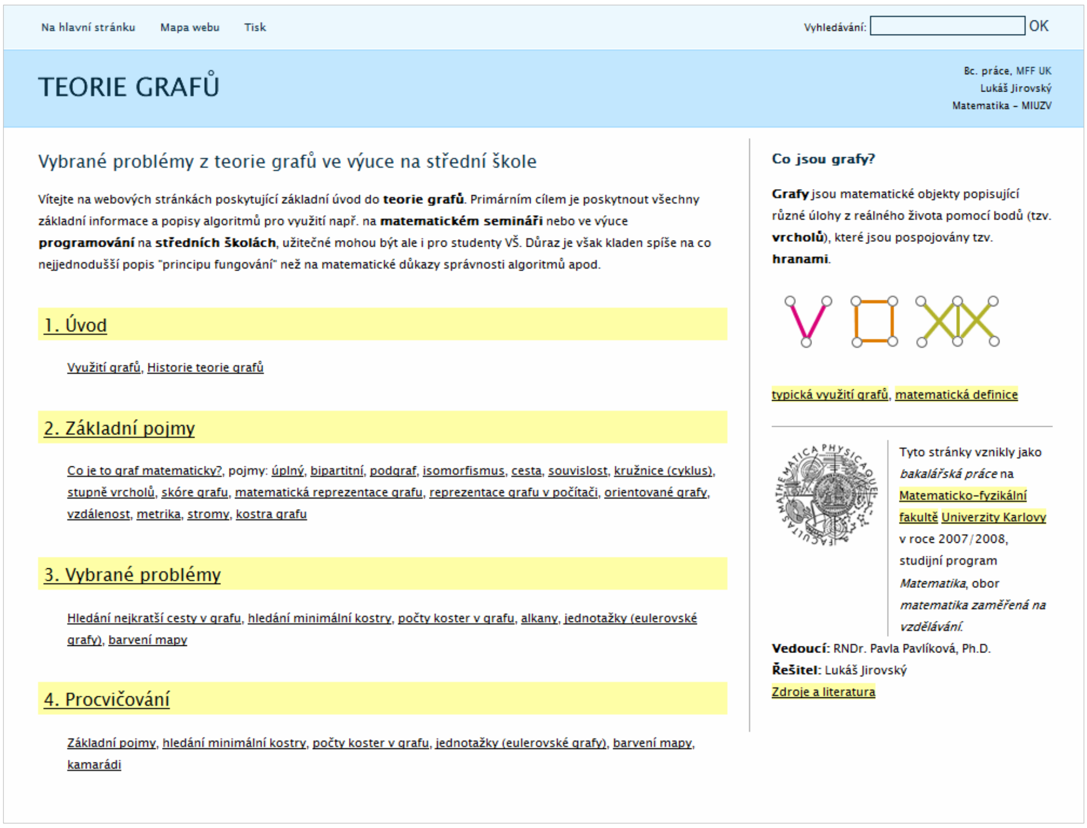
Keywords: Graph Theory, Spanning Tree, Map Colouring, Shortest Path Tree.

Úvod k práci

Cílem práce bylo vytvořit internetové stránky, které studentům středních škol poskytnou úvod do teorie grafů a ukáží nejznámější problémy spolu s praktickým využitím (například: hledání nejkratší cesty v grafu, minimální kostry apod.).

Do práce byly vybrány i problémy z různých oborů (mj. souvislost definice *stromů* a chemických vzorců *alkanů*) pro poukázání na přesahy matematiky do jiných oblastí. Kromě toho také práce nabídne matematický základ pro teorii grafů při jejím využití na semináři (nejčastěji v kombinatorice) či výuce programování. Vhodná může být také pro samostudium řešitelům úloh z korespondenčních seminářů Matematicko-fyzikální fakulty.

Práci ve formě webových stránek naleznete na přiloženém CD.



Náhled úvodní stránky webové verze práce

1. Úvod k teorii grafů

Co jsou grafy?

Ještě před matematickou definicí grafu je vhodné si říct, co **grafy** jsou a k čemu se využívají. Grafů rozeznáváme celou řadu typů, v tomto textu nám však nepůjde o grafy využívané ve statistice (sloupcový, koláčový...), ani o grafy funkcí.

Grafy si můžeme představit jako zjednodušení reálného světa, kde studovaný problém znázorníme pomocí bodů a čar, které je spojují, a tím popisují vlastnosti daného problému. Takovým bodům pak v teorii grafů říkáme **vrcholy grafu** a čáry, které je spojují, nazýváme **hrany grafu**.

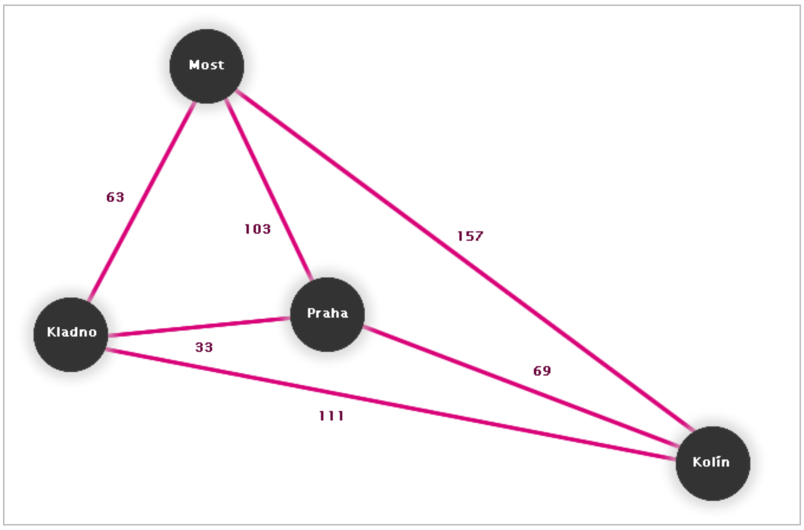
Příklad 1: města na mapě

Jedním z typických problémů teorie grafů je **hledání nejkratší cesty**.

Představme si, že chceme zjistit nejkratší cestu mezi vybranými městy (pro zjednodušení jsou na obrázku č. 2.28 jen čtyři města).

Jediná vlastnost, která nás bude zajímat, je vzdálenost mezi městy (vzdálenost = délka cesty, kterou bychom museli ujet po silnici). Jako **vrcholy** tedy použijeme **města**; **hrany** pak budou popisovat, že mezi městy existuje **cesta** a jak je dlouhá. Přitom zanedbáváme další, v tuto chvíli zbytečné informace - z kolika různých silnic se cesta skládá, jakého jsou tvaru nebo například zeměpisná poloha měst (sever, jih...). Jediné informace, které pak algoritmus (zjednodušeně: postup řešení problému) využije, budou čísla na hranách spojujících města (vzdálenosti mezi městy).

Podrobněji se k této úloze vrátíme v kapitole 2.

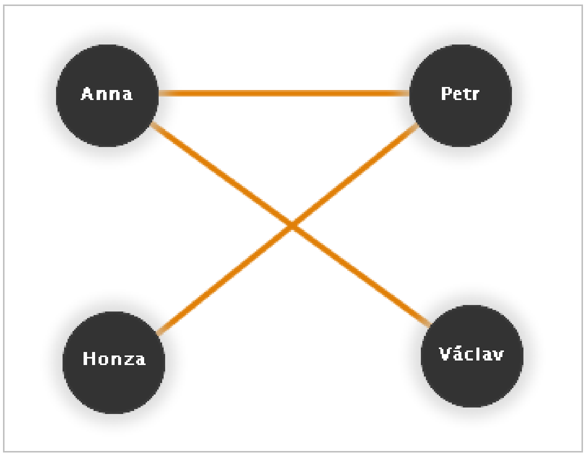


Znázornění vzdáleností mezi městy pomocí grafu [10] - obrázek z Kapitoly 2, 2.28

Příklad 2: kamarádi

Následující problém ukazuje možnosti popisu rozličných situací pomocí grafů.

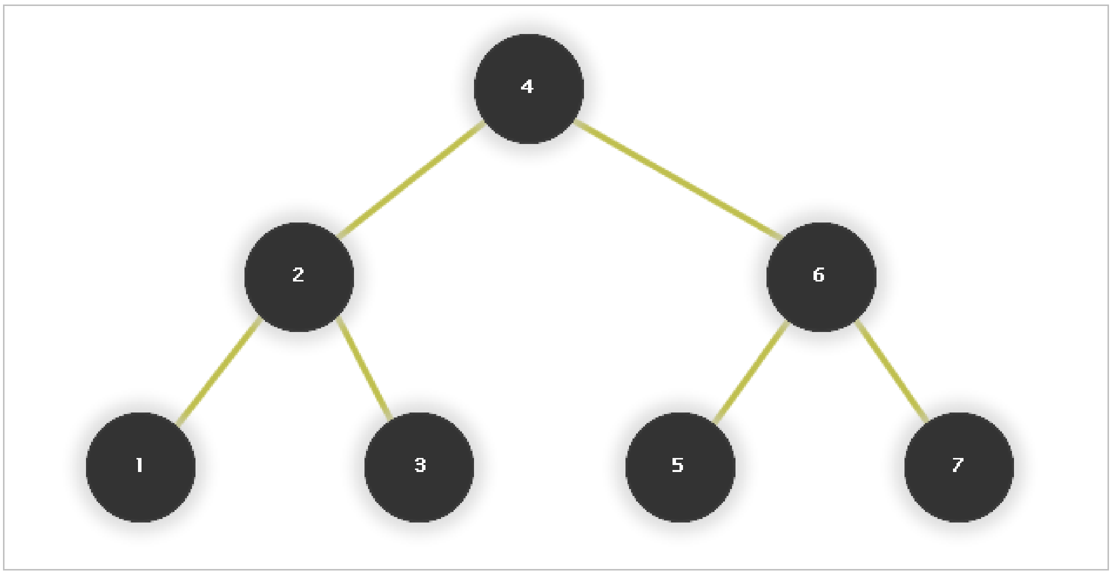
Mějme čtyři kamarády: Annu, Petra, Honzu a Václava. Pomocí grafu znázorníme, kdo koho zná (kdo s kým kamarádí). Znalostí grafů lze pak jednodušeji dokazovat různé souvislosti - podobnou úlohu z matematické olympiády si můžete vyzkoušet v kapitole 4.



Obr. č. 1.1 - Kamarádi

Příklad 3: stromy

Stromy jsou zvláštním typem grafů často používaným v informatice (více se o stromech dočtete v kapitole 2). Pomocí vhodně navrženého stromu lze např. jednodušeji vyhledávat či třídit čísla - příklad: *binární vyhledávací strom*.



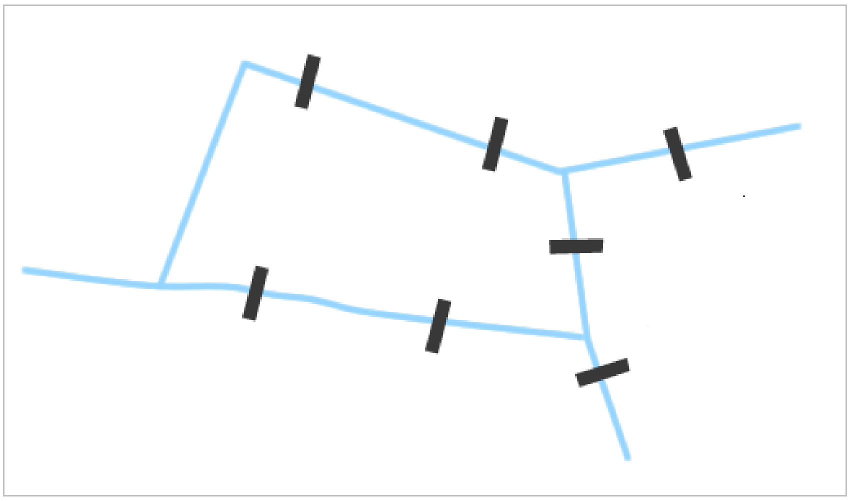
Binární vyhledávací strom - obrázek z Kapitoly 2, 2.30

Historie teorie grafů

18. století - Eulerova úloha / Sedm mostů města Königsbergu

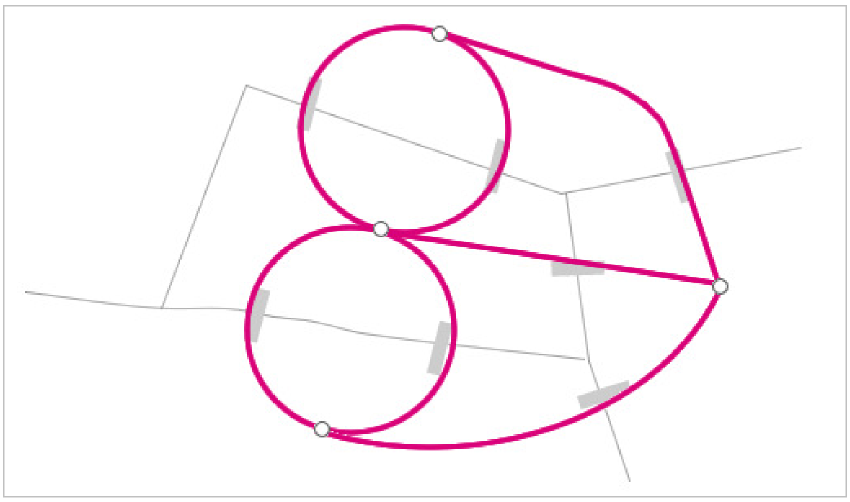
Za zakladatele teorie grafů je považován **Leonhard Euler** (1707-1783), který roku 1736 publikoval řešení příkladu **Sedmi mostů města Königsbergu** (Královce).

Zadání úlohy znělo, zda je možné projít každým mostem ve městě právě jednou a vrátit se zpět do původního místa.



Obr. č. 1.2 - Sedm mostů města Königsbergu

Převedení situace na graf provedl Euler tak, že si každý břeh představil jako vrchol a každý most použil jako hranu, která břehy spojuje. Matematicky dokázal, že úloha není řešitelná. Více o problému jednotažek naleznete v kapitole 3.

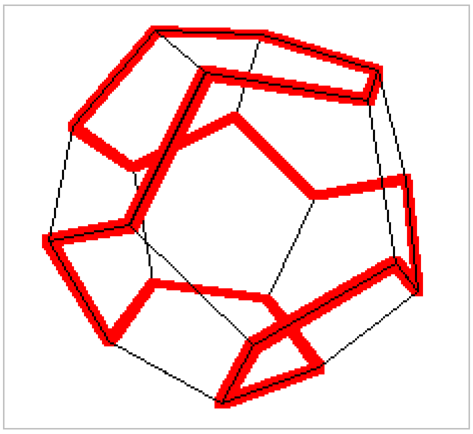


Obr. č. 1.3 - Sedm mostů města Königsbergu – znázornění pomocí grafu

19. století - fyzika a chemie

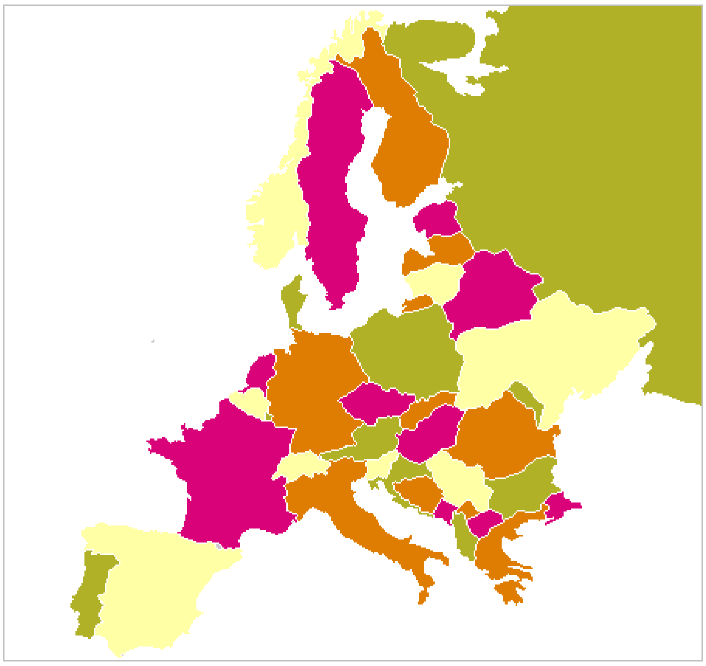
Grafy pak zůstávaly přes sto let na okraji zájmu matematiků - vrátili se k nim až roku 1847 **Gustav Kirchhoff** (1824-1887), když se zabýval výpočtem proudů v elektrických sítích pomocí počtu koster grafu, a 1857 **Arthur Cayley** (1821-1895), který pomocí grafů zjišťoval počty různých uskupení molekul alkanů (více o problému v kapitole 3).

V roce 1857 vymyslel **sir William Hamilton** (1805-1865) hru, jejímž úkolem bylo pospojovat všechny vrcholy pravidelného dvanáctistěnu tak, aby byl každý vrchol použit právě jednou (viz obrázek č. 1.4) - podle toho vznikl pojem **hamiltonovská kružnice**, jako kružnice, která projde právě jednou všemi vrcholy grafu (pojem *kružnice* je vysvětlen v kapitole 2). Hra byla pojmenována **The Icosian Game** (*icosian* = dvacítkový; dvanáctistěn má dvacet vrcholů).



Obr. č. 1.4 - Jedno z možných řešení The Icosian Game - [2]

Nejznámější úlohou teorie grafů v 19. století byl **problém čtyř barev** – otázka zněla, zda-li je možné každou mapu obarvit pomocí čtyř barev tak, aby sousední státy měly různé barvy (více o zadání a řešení problému se dočtete v kapitole 3), který byl kompletně vyřešen až roku 1976.



Obr. č. 1.5 - Obarvení mapy čtyřmi barvami

20. století u nás

Ve 20. století docházelo k velkému rozvoji teorie grafů. Významných poznatků bylo dosaženo i v Československu. V roce 1926 publikoval **Otakar Borůvka** (1899-1995) svůj algoritmus pro nalezení minimální kostry. Tento problém měl praktický důvod - co nejvýhodnější výstavbu elektrických sítí. Stejný problém vyřešil (jiným algoritmem) roku 1930 také **Vojtěch Jarník** (1897-1970).

2. Základní pojmy

Matematická definice grafu

Definice

Jak už bylo naznačeno v úvodu, grafy jsou vhodným prostředkem pro popis situací, které lze znázornit pomocí **konečného množství** bodů a vztahů mezi nimi znázorněnými pomocí hran.

Samotný graf *G* je definován jako dvojice těchto dvou množin - vrcholů (*V*) a hran (*E*).
Někdy také místo dvojice mluvíme o **uspořádané dvojici** – myslíme tím, že na prvním místě jsou uvedeny vrcholy a na druhém hrany. Zkratky *V* a *E* pocházejí z angličtiny - vrcholy jsou anglicky *VERTICES*, hrany *EDGES*.

> G = (V,E)

V dalších definicích používáme označení *V(G)* a *E(G)* pro množinu vrcholů *V* (resp. množinu hran *E*) grafu *G*.

Značení

> V ... množina vrcholů
> |V| ... velikost množiny vrcholů, tedy počet vrcholů (číslo)
> E ... množina hran
> |E| ... velikost množiny hran, tedy počet hran (číslo)

Důsledky

Množina hran *E* je podmnožinou množiny všech možných dvojic navzájem různých vrcholů – nepřipouštíme tedy „násobné“ hrany ani „smyčky“.

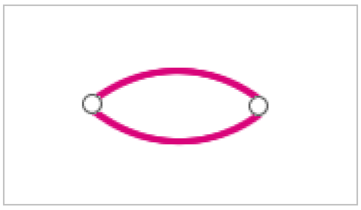
$$E \subseteq \binom{V}{2}$$

Hran může být v grafu (|*E*|) maximálně tolik jako v případě, kdyby byly každé dva vrcholy grafu byly spojené hranou.

$$|E| \leq 2^{\binom{|V|}{2}}$$

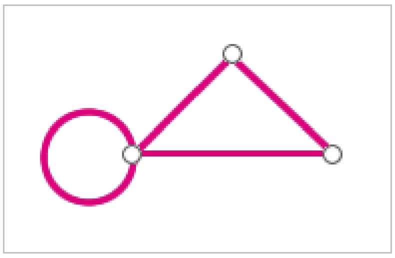
Poznámka

V této práci se budeme zabývat grafy, kde jsou každé dva vrcholy spojené **nanejvýš jednou hranou**. V případě, že jsou dva vrcholy spojené větším množstvím hran, hovoříme o takzvaných **násobných hranách**. Takovým grafům pak říkáme **multigrafy**.



Obr. č. 2.1 - Příklad násobných hran

Podobně také nebudeme připouštět grafy obsahující **smyčky** - hrany spojující jeden vrchol sám se sebou.



Obr. č. 2.2 - Příklad smyčky v grafu

Úplný graf

Popis

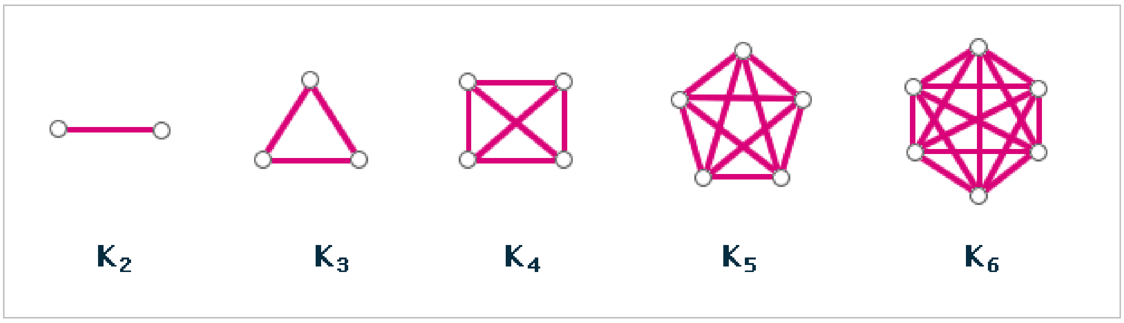
Úplný graf je takový graf, ve kterém jsou každé dva vrcholy spojené hranou.

$$|E| = \binom{|V|}{2}$$

Úplné grafy značíme K_n, kde n je počet vrcholů.

V úvodním příkladu (obr. 1.1) by úplnému grafu odpovídala skupina lidí, kde se zná každý s každým.

Příklady úplných grafů

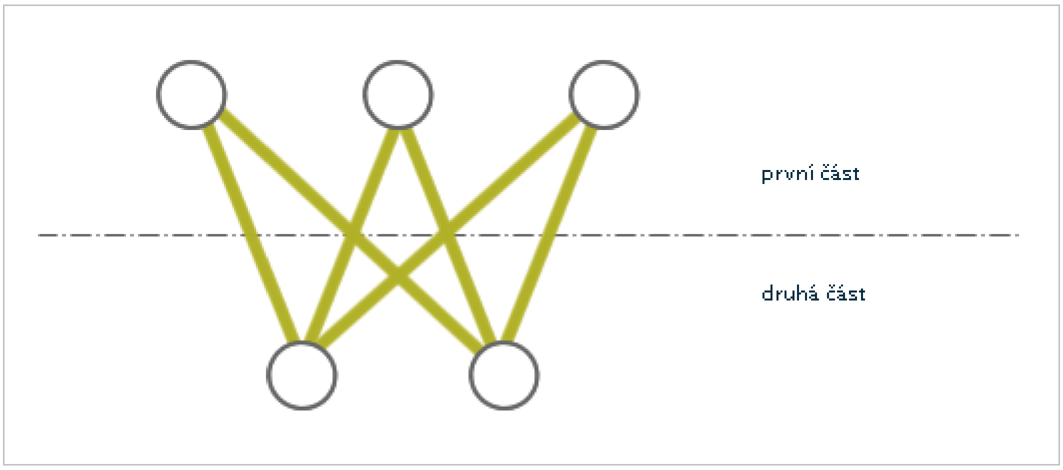


Obr. č. 2.3 - Příklady úplných grafů

Bipartitní graf

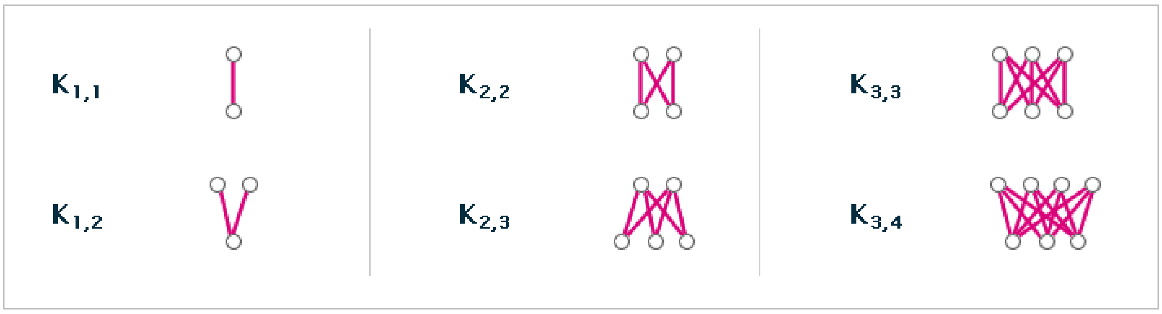
Popis

Bipartitní graf je takový graf, jehož množinu vrcholů lze rozdělit na dvě části, přičemž z každého vrcholu jedné části jde hrana pouze do vrcholů druhé části a naopak. Pokud jde z každého vrcholu jedné části hrana do **každého** vrcholu druhé části, mluvíme o **úplném bipartitním grafu**.



Obr. č. 2.4 - Bipartitní graf - ilustrace

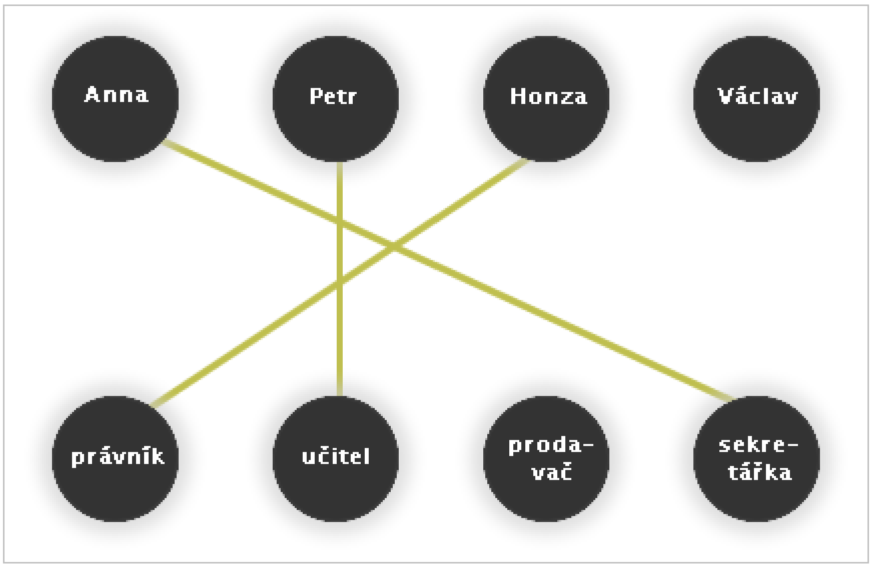
Podobně jako se úplné grafy značí K_n, kde n je počet vrcholů, pro označení **úplných bipartitních grafů** se používá značení $K_{m,n}$, kde m a n odpovídá počtu vrcholů jednotlivých částí.



Obr. č. 2.5 - Příklady úplných bipartitních grafů

Příklad v praxi

Jako příklad grafu, který musí být bipartitní, si můžeme představit graf znázorňující obsazení pracovních míst (obr. č. 2.6). V takovém grafu nikdy nespojíme dvě pracovní místa, ani dva zaměstnance.



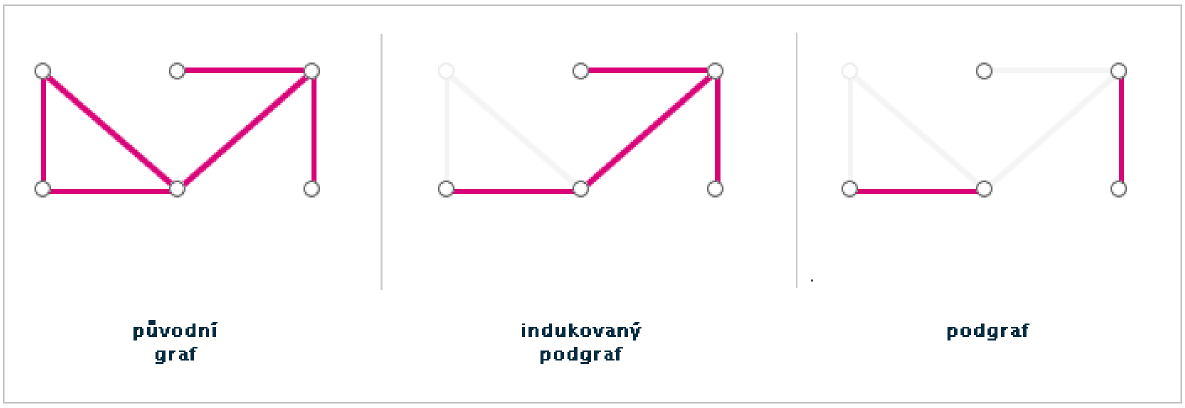
Obr. č. 2.6 - Příklad popisu obsazení pracovních míst pomocí bipartitního grafu

Podgraf

Co je podgraf?

Podgraf grafu *G* je graf *H*, který vznikl odebráním některých vrcholů a hran z původního grafu *G*.

Při odebrání vrcholu je nutné vymazat všechny hrany vedoucí do (z) tohoto vrcholu. Pokud byly odebrány jen tyto hrany, nazývá se podgraf **indukovaný**. Pokud byly odebrány i jiné hrany, jde obecně o podgraf.



Obr. č. 2.7 - Podgrafy

Definice

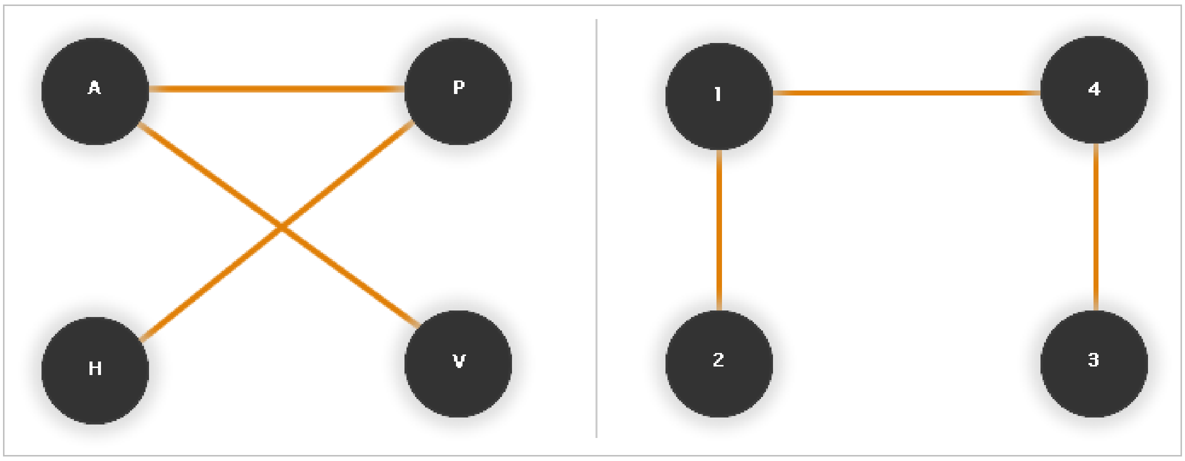
Graf *H* je **podgrafem** grafu *G*, jestliže $V(H) \subseteq V(G)$ a $E(H) \subseteq E(G)$.

Graf *H* je **indukovaným podgrafem** grafu *G*, jestliže $V(H) \subseteq V(G)$ a

$$E(H) = E(G) \cap \binom{V(H)}{2}.$$

Isomorfismus

Podle definice je graf zadán svými vrcholy (množina *V*) a hranami (množina *E*). Pokud ale budeme konkrétní situaci popisovat pomocí grafů, může se nám stát, že dva grafy odpovídající téže situaci budou mít různě označené nebo umístěné vrcholy. Například v úloze o kamarádech z úvodu použijeme pro označení vrcholů jednou první písmena křestních jmen a podruhé kamarády očíslujeme. Grafy jsou na první pohled různé - ale popisují stejnou situaci (kamarádi jsou stále ti samí). V takovém případě říkáme, že dané dva grafy jsou **isomorfní**.



Obr. č. 2.8 - Isomorfní grafy

Zda jsou grafy isomorfní můžeme ověřit tak, že najdeme způsob, jak by bylo nutné vrcholy prvního grafu přejmenovat, aby odpovídaly vrcholům druhého grafu - přejmenujeme-li vrcholy *A* a *B* na *1* a *2*, musí být *1* a *2* spojeny hranou, pokud byly *A* a *B* spojeny hranou (a samozřejmě *1* a *2* nesmí být spojeny, pokud *A* a *B* nebyly spojeny hranou).

Takovému "přejmenování" pak říkáme **bijektivní zobrazení**. *Bijektivní* je synonymum pro *vzájemně jednoznačné* - každý vrchol se zobrazuje na právě jeden (dva původní vrcholy se nikdy nezobrazí na jeden nový, ani jeden původní na víc nových).

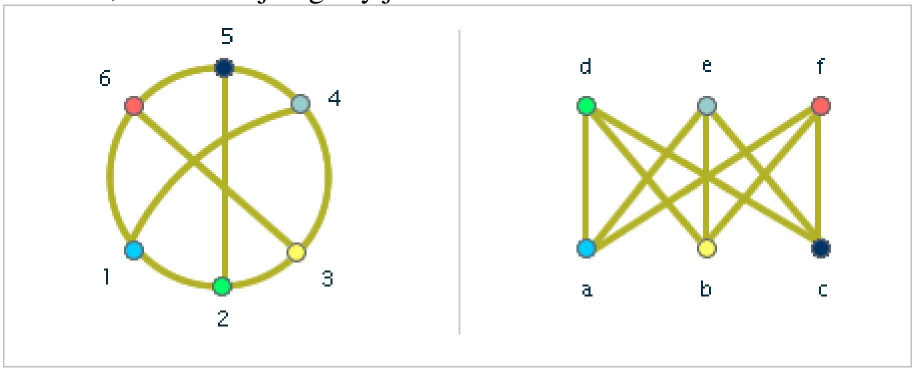
Příklad takového přejmenování pro grafy z obr. 6 by bylo: A→1, H→3, P→4, V→2.

Definice

> Dva grafy $G=(V,E)$ a $G'=(V',E')$ nazýváme **isomorfní**, jestliže existuje bijektivní (vzájemně jednoznačné) zobrazení $f : V \to V'$ tak, že platí:
> $\{x,y\} \in E$, právě když $\{f(x), f(y)\} \in E'$.

Příklad

Ukažme, že následující grafy jsou isomorfní:



Obr. č. 2.9 - Isomorfní grafy

Hledaným isomorfismem je například zobrazení: a→1, b→3, c→5, d→2, e→4, f→6.

Cesta a souvislost v grafu

Cesta

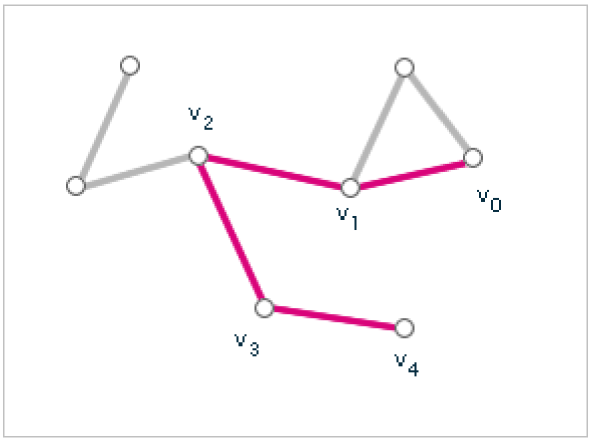
Definice

> **Cestu v grafu** můžeme chápat jako posloupnost vrcholů a hran (v_0, e_1, v_1,..., e_t, v_t), kde vrcholy v_0,..., v_t jsou navzájem různé vrcholy grafu G a pro každé $i = 1,2,...,t$ je $e_i = \{v_{i-1}, v_i\} \in E(G)$.

Poznámka

Tato definice povoluje i cestu délky 0.

Příklad cesty v grafu



Obr. č. 2.10 – Příklad cesty v grafu (v_0, v_1, v_2, v_3, v_4)

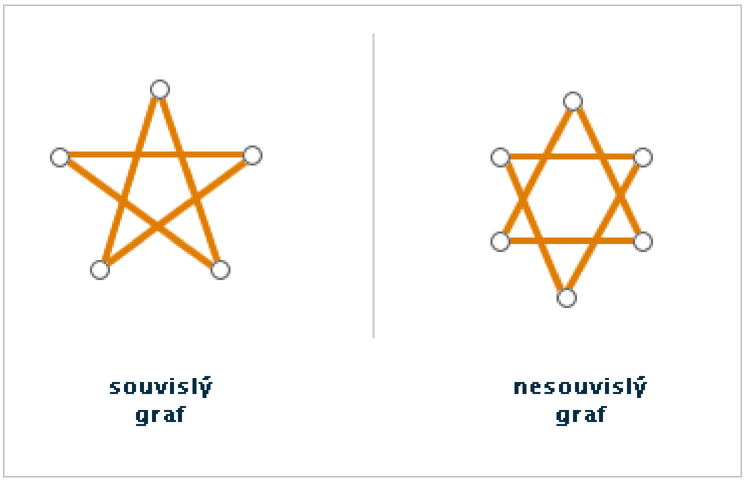
Souvislost

Někdy potřebujeme vyjádřit, jestli je graf "jedním celkem" - můžeme-li se dostat z každého vrcholu nějakou cestou do jiného, nebo zda jde o více na sebe nenavazujících částí. Proto se zavádí pojem **souvislost grafu**:

Definice

> Řekneme, že (neorientovaný) graf G je **souvislý**, jestliže pro každé jeho dva vrcholy x a y existuje v G cesta z x do y.

Příklad souvislého a nesouvislého grafu



Obr. č. 2.11 - Souvislý a nesouvislý graf

Pokud graf není souvislý, části, ze kterých se skládá (a které jsou samy o sobě souvislé), se nazývají **komponenty souvislosti**. Na obr. č. 2.11 vpravo by byly komponentami dva trojúhelníky.

Kružnice (cyklus) v grafu

Definice

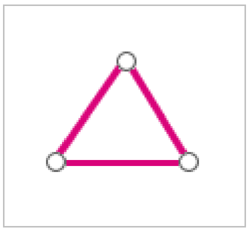
> **Kružnicí** (cyklem) v grafu rozumíme posloupnost vrcholů a hran (v_0, e_1, v_1,..., e_t, v_t = v_0), kde vrcholy v_0,..., v_{t-1} jsou navzájem různé vrcholy grafu *G* a pro každé *i* = 1,2,...,*t* je $e_i = \{v_{i-1}, v_i\} \in E(G)$.

Poznámka

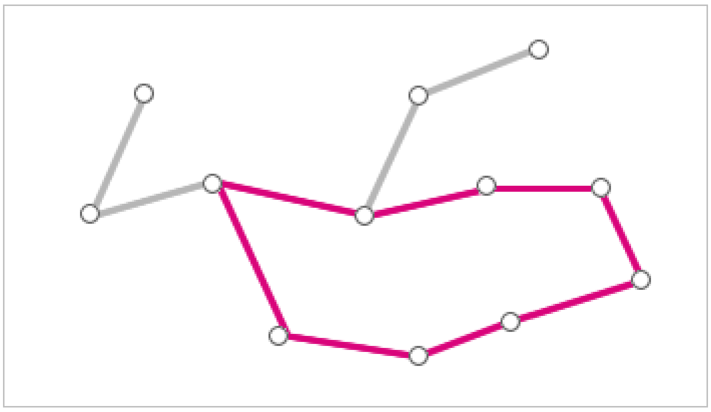
Nejkratší kružnicí je trojúhelník (K_3 - úplný graf se třemi vrcholy).

Povšimněte si, že definice kružnice se podobá definici cesty - opět jde o posloupnost hran a vrcholů. Na rozdíl od cesty je ale první a poslední vrchol posloupnosti stejný. V cestě ale povolujeme i délku 0 (prázdnou posloupnost). Kružnice má přitom minimální délku 3.

Příklady



Obr. č. 2.12 - Nejkratší kružnice (trojúhelník)



Obr. č. 2.13 - Kružnice v obecném grafu

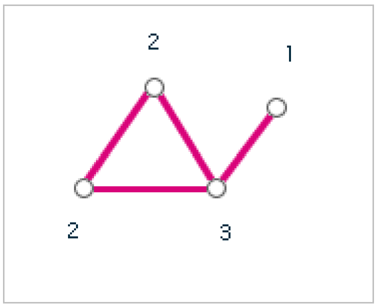
Stupně vrcholů / skóre grafu

Stupně vrcholů

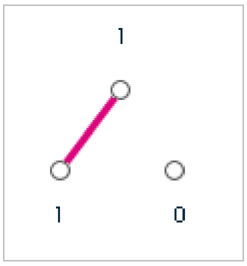
Definice

> Nechť *G* je graf, *v* jeho vrchol. Symbolem $\mathbf{deg}_G(v)$ označme **počet hran** grafu *G* **obsahujících vrhchol *v***. Číslo $\deg_G(v)$ nazveme **stupněm vrcholu *v*** v grafu *G*.

Příklad



Obr. č. 2.14 - Příklad stupňů vrcholů v grafu



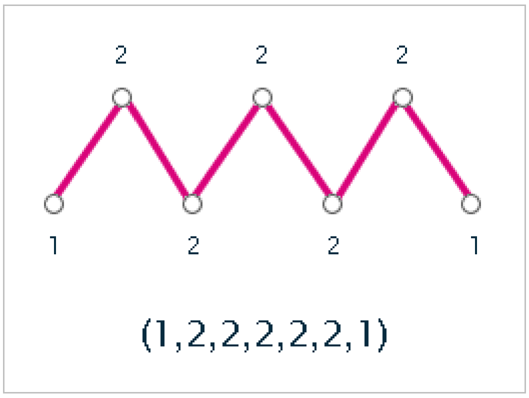
Obr. č. 2.15 - Příklad grafu s vrcholem stupně 0

Skóre grafu

Definice

> Označme vrcholy grafu G $v_1, v_2, \dots, v_n$ (v libovolně zvoleném pořadí). Posloupnost
> $(\mathbf{deg}_G(v_1), \mathbf{deg}_G(v_2), \dots, \mathbf{deg}_G(v_n))$
> nazýváme **posloupnost stupňů** grafu G nebo **skóre** grafu *G*.
>
> Dvě skóre považujeme za stejná, pokud jedno můžeme dostat z druhého přerovnáním čísel (permutací).

Příklad

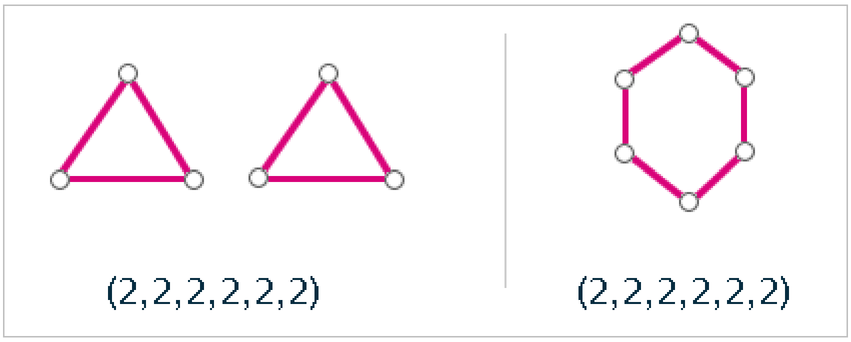


Obr. č. 2.16 - Příklad skóre grafu

Důsledek

> Pokud mají dva grafy různé skóre, nejsou isomorfní.

Neplatí ale opak tohoto tvrzení (stejné skóre → isomorfní grafy). Příklad, který by takovéto tvrzení nesplnil, je na obr. č. 14 – zkuste si rozmyslet, proč grafy nejsou isomorfní, ačkoliv mají stejné skóre.



Obr. č. 2.17 - Příklad dvou neisomorfních grafů se stejným skóre

Matematická reprezentace grafu

Dosud jsme všechny grafy popisovali pomocí obrázků. Často ale potřebujeme graf zadat pomocí čísel, například pro výpočty na počítači.

Matice sousednosti

Nejtypičtějším zápisem grafu je použitím **matice**.

Co je to matice?

Matice je matematické schéma, pomocí kterého se zapisují čísla (nebo jiné matematické objekty) do obdélníkového tvaru.

Matice se používají v algebře např. pro řešení soustav rovnic. Pro účely teorie grafů nám postačí představit si matici jako **tabulku čísel**, kterou uzavřeme do závorek.

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$$

Obr. č. 2.18 - Příklad obecné matice

Často potřebujeme čísla z matice číst (nebo zapisovat) pomocí určení pozice prvku dvojicí (řádek, sloupec). Například na obr. 2.18 by číslo 10 bylo na pozici (3,2).

V této části se budeme zabývat neorientovanými grafy - zápis orientovaných grafů je popsán později v této kapitole.

Postup vytváření matice sousednosti

- Nejprve si všechny vrcholy očíslujeme.
- Za rozměr matice musíme zvolit počet vrcholů. Kdybychom měli například graf se 4 vrcholy ale matici rozměrů jen 3×3, nemohli bychom zapsat hrany vedoucí do/z čtvrtého vrcholu.

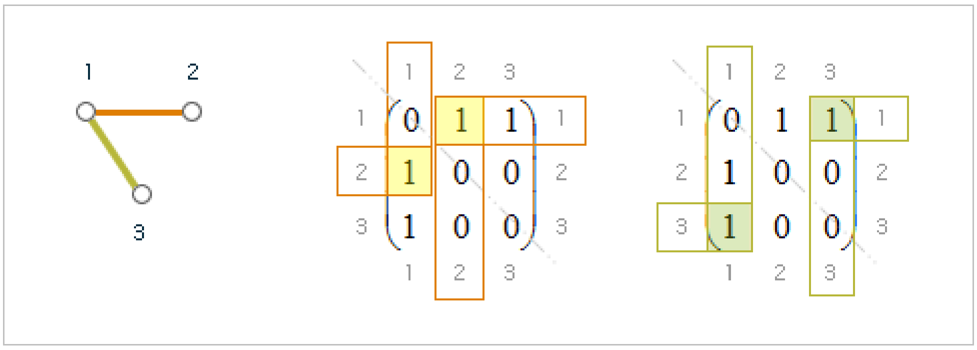
- Pokud vede mezi dvěma vrcholy hrana, zapíšeme do matice na pozici [*číslo jednoho vrcholu*, *číslo druhého vrcholu*] a na pozici [*číslo druhého vrcholu*, *číslo prvního vrcholu*] číslo 1. Jinak zapíšeme 0.

Definice

Nechť G=(V,E) je graf s *n* vrcholy. Označme vrcholy $v_1,\ldots,v_n$ (v nějakém libovolném pořadí). **Matice sousednosti** grafu G je čtvercová matice $A_G = (a_{ij})_{i,j=1}^{n}$ definovaná předpisem:

$$a_{ij} = \begin{cases} 1 & \text{pro } \{ v_i, v_j \} \in E \\ 0 & \text{jinak.} \end{cases}$$

Příklad



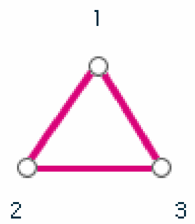
Obr. č. 2.19 – Graf a jeho matice sousednosti

Poznámka

Pro neorientované grafy platí, že jejich matice sousednosti jsou symetrické.

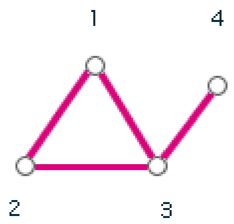
Pokud je graf G úplný, obsahuje matice A_G s výjimkou hlavní diagonály samé jedničky.

Příklady



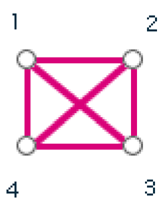
$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Obr. č. 2.20 – Graf a jeho matice sousednosti



$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Obr. č. 2.21 – Graf a jeho matice sousednosti

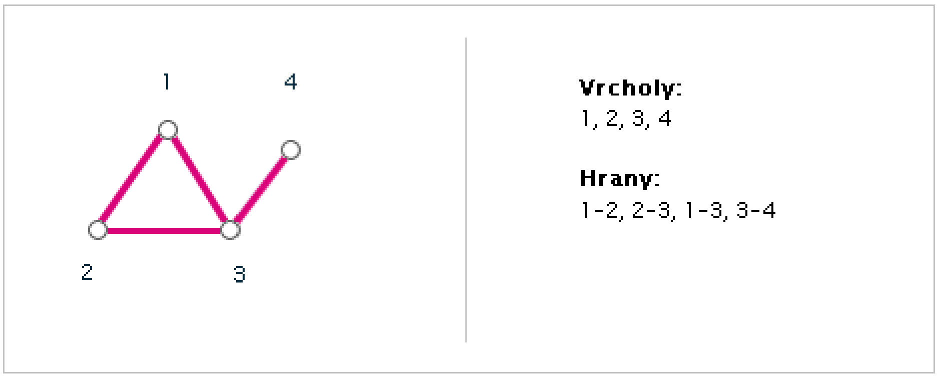


$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Obr. č. 2.22 – Úplný graf a jeho matice sousednosti

Jiné možnosti reprezentace grafu

Kromě matice sousednosti je možné použít i jiné reprezentace - pokud bude graf mít hodně vrcholů, které budou spojeny relativně malým počtem hran mezi nimi, je výhodnější použít například **seznam sousedů** či **seznam vrcholů se seznamem hran**.



Obr. č. 2.23 - Příklad zápisu grafu pomocí seznamů vrcholů a hran

Reprezentace grafu v počítači

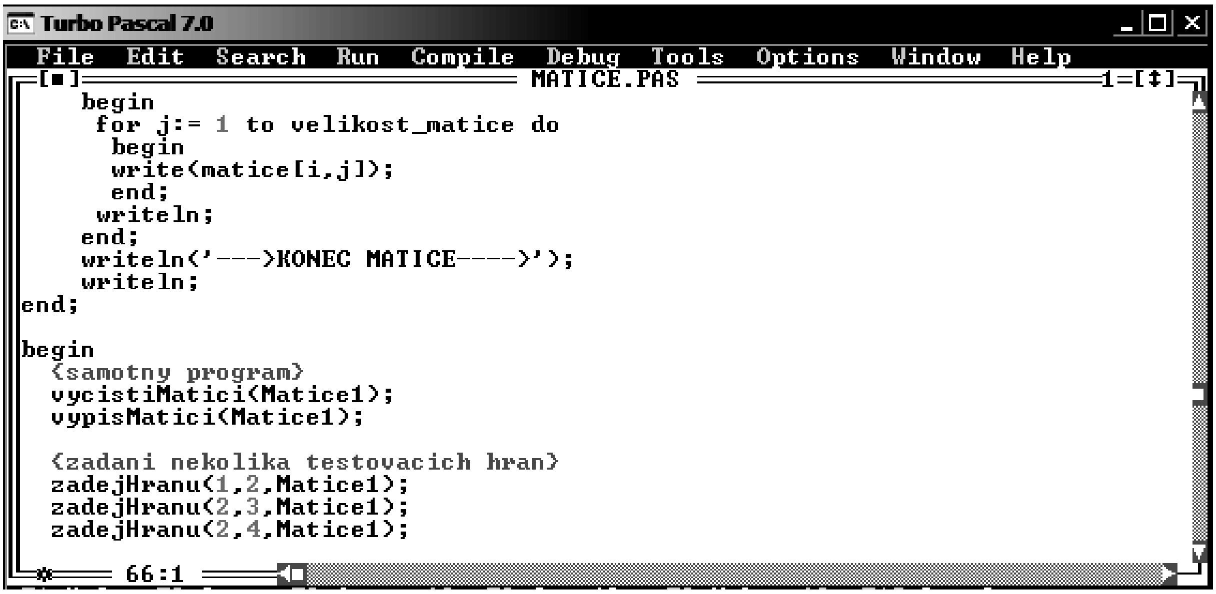
Konkrétní reprezentace grafu se při práci na počítači často liší podle vlastností grafu. Například víme-li, že graf bude mít hodně vrcholů a relativně málo hran, může být matice zbytečné plýtvání pamětí počítače a vyplatí se použít seznam sousedů. Speciální kapitolou jsou také dynamické datové struktury, kde se grafy (ještě častěji přímo stromy) za běhu programu velmi mění a celý graf se může několikanásobně zvětšit. Pro používání takovýchto struktur je ale třeba velmi dobře ovládat práci s pamětí (s ukazateli, příkazy jako *new* (Pascal) či *malloc* (C)).

Graf reprezentovaný maticí

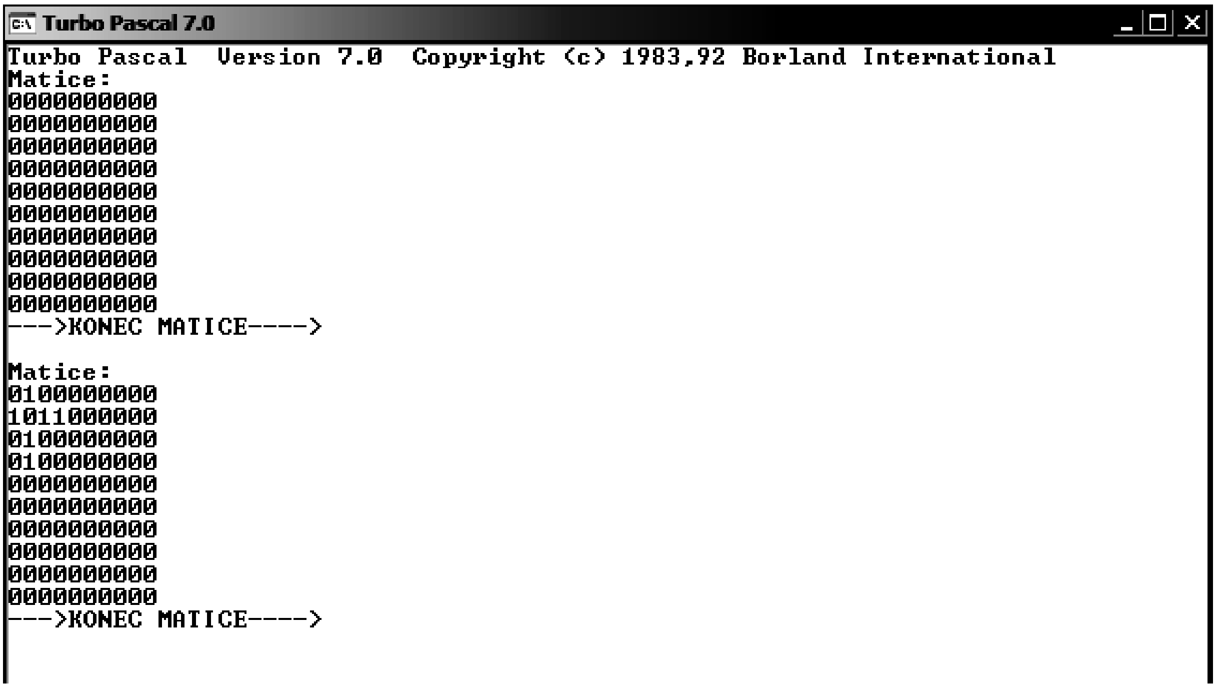
Příklad popisuje graf reprezentovaný maticí sousednosti. Jde o jednoduchý neorientovaný graf bez jakékoliv metriky. Programovací jazyk (prostředí): Pascal (Turbo Pascal 7).

Program přímo žádný algoritmus nepočítá, pouze ukazuje práci s grafem (přidání hrany, odebrání...).

Ukázky programu



Obr. č. 2.24 - Turbo Pascal 7 (programovací prostředí, obrázek upraven pro tisk)



Obr. č. 2.25 - Turbo Pascal 7 (běh programu, obrázek upraven pro tisk)

Poznámky pro programátory

- Pro matici je použito dvourozměrné pole (typ *typ_matice*).
- Rozměry matice jsou určené konstantou *velikost_matice*. Takovéto využití pole o dopředu naalokované paměti brání zvětšování grafu (přidávání nových vrcholů) - oproti dynamickým datovým strukturám.
- Matice se předává procedurám *odkazem* (nikoliv *hodnotou*) - matice je tak globální proměnnou, do které procedury zasahují a upravují ji.
- Zaznamenání hrany i odebrání hrany je jednoduché - pouze se změní číslo na souřadnici určené pořadovými čísly vrcholů.

Kód programu

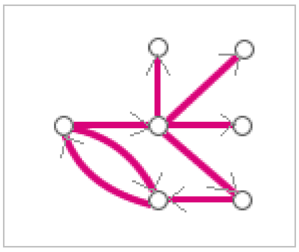
```
program GRAFMATICE1;

const
  velikost_matice = 10;

type
  typ_matice = array[1..velikost_matice,1..velikost_matice] of integer;
```

```
var
  matice1: typ_matice;

procedure vycistiMatici(var matice:typ_matice);
var
  i, j: integer;
begin
  for i:=1 to velikost_matice do
    begin
    for j:= 1 to velikost_matice do
      begin
      matice[i,j]:= 0;
      end;
    end;
end;

procedure zadejHranu(odkud: integer; kam: integer; var matice:typ_matice);
begin
  matice[odkud,kam]:= 1;
  matice[kam,odkud]:= 1;
end;

procedure odeberHranu(odkud: integer; kam: integer; var matice:typ_matice);
begin
  matice[odkud,kam]:= 0;
  matice[kam,odkud]:= 0;
end;

function jeHrana(odkud: integer; kam: integer; var matice:typ_matice):boolean;
begin
  if(matice[odkud,kam] = 1) then jeHrana:=true else
                                 jeHrana:=false;
end;

procedure vypisMatici(var matice:typ_matice);
var
   i,j: integer;
begin
   writeln('Matice:');
   for i:=1 to velikost_matice do
    begin
     for j:= 1 to velikost_matice do
      begin
```

```
      write(matice[i,j]);
      end;
     writeln;
    end;
    writeln('--->KONEC MATICE---->');
    writeln;
end;

begin
  {samotny program}
  vycistiMatici(Matice1);
  vypisMatici(Matice1);

  {zadani nekolika testovacich hran}
  zadejHranu(1,2,Matice1);
  zadejHranu(2,3,Matice1);
  zadejHranu(2,4,Matice1);

  vypisMatici(Matice1);

  readln;
end.
```

Orientované grafy

Zatím jsme používali pouze tzv. *neorientované grafy* - Pokud vedla hrana z vrcholu *A* do vrcholu *B*, vedla hrana také z *B* do *A* (nerozlišovali jsme směr hrany). Při popisu reálných situací pomocí grafů se nám často může stát, že hrana vede pouze jedním směrem - např. u silniční sítě by takový stav nastal u jednosměrné silnice. Z takových důvodů zavádíme tzv. **orientované grafy**.



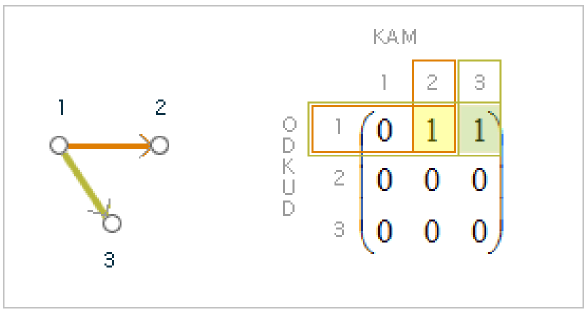
Obr. č. 2.26 - Příklad orientovaného grafu

Definice

Orientovaný graf *G* je dvojice (*V*,*E*), kde *E* je podmnožina kartézského součinu V × V. Prvky *E* nazýváme **šipky** nebo **orientované hrany**. Orientovaná hrana *e* má tvar (*x*, *y*). Říkáme, že tato orientovaná hrana vychází z *x* a končí v *y*.

Reprezentace pomocí matice sousednosti

Pokud pro reprezentaci grafu použijeme matici sousednosti, bude situace velmi podobná jako u neorientovaných grafů. Musíme si ovšem u souřadnic prvku uvědomit, že první číslo znamená *odkud* a druhé *kam* hrana vede. Přestává totiž platit, že hodnota prvku na souřadnicích např. (1,2) je stejná jako hodnota prvku na pozici (2,1). Proto také matice sousednosti orientovaného grafu není nutně symetrická.

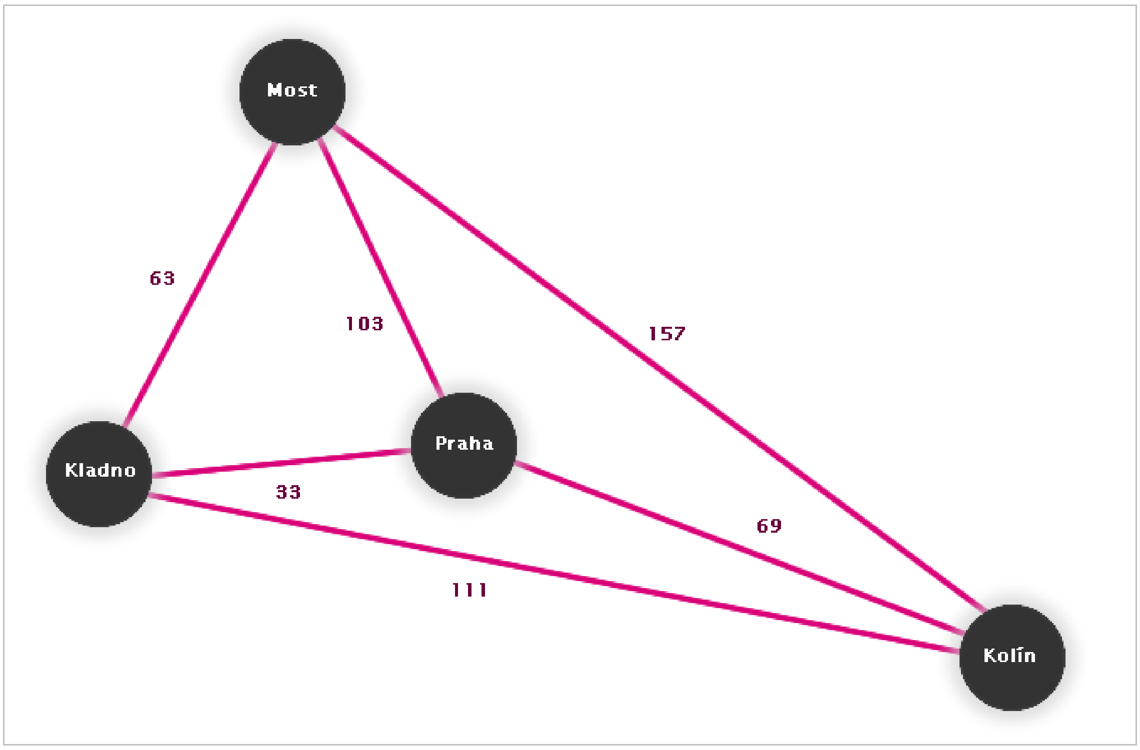


Obr. č. 2.27 - Reprezentace orientovaného grafu pomocí matice sousednosti

Vzdálenost / metrika

Ohodnocení hran

Již v úvodu k teorii grafů (Kapitola 1) jsme ukázali, že hrany k sobě mohou mít přiřazené číslo, které si lze představit jako vzdálenost mezi městy nebo např. cenu, kterou musíme zaplatit za průchod hranou. Říkáme, že graf má **ohodnocené hrany**.



Obr. č. 2.28 - Znázornění vzdáleností mezi městy pomocí grafu

Definice ohodnocení

Funkci, která k hranám přiřadí čísla (budeme pro jednoduchost pracovat s kladnými reálnými čísly) nazveme **ohodnocením hran** a označíme ji w.

> $w\colon E(G) \rightarrow (0, +\infty)$

Grafová vzdálenost / metrika

Úvod

Pokud máme ohodnocené hrany, jsme schopni říct, jaká je vzdálenost mezi každými dvěma městy (reprezentovanými vrcholy). (Musíme se ovšem umět z každého města dostat do jiného (např. cestou přes nějaké třetí) - graf musí být souvislý.) Funkci, která nám dosadí k vrcholům **nejkratší vzdálenosti**, nazveme **metrika**.

Definice vzdálenosti v grafu

> Nechť $G = (V,E)$ je souvislý graf. Pro vrcholy v, v' definujme číslo $d_G(v,v')$ jako délku nejkratší cesty z v do v' v grafu G. Číslo $d_G(v,v')$ se nazývá **vzdálenost vrcholů *v* a *v'* v grafu G**.

Vlastnosti metriky

Funkce d_G: $V \times V \rightarrow R_0^+$, kterou nazýváme **metrika grafu G**, má následující vlastnosti:

- $d_G(v,v') \geq 0$ pro každou dvojici v, v'
- $d_G(v,v') = 0$ právě tehdy, když $v = v'$ pro každou dvojici v, v'
- $d_G(v,v') = d_G(v',v)$ pro každou dvojici v, v'
- $d_G(v,v'') \leq d_G(v,v') + d_G(v',v'')$ pro každou trojici v, v', v''

Poslední vlastnost je tzv. **trojúhelníková nerovnost** - říká nám, že jsme v grafu získali vždy nejkratší vzdálenosti - pokud bychom se pokusili jít z města (vrcholu) v do v'' přes v', bude cesta stejně dlouhá nebo delší (→ v tabulce metriky známe nejkratší cesty).

Příklad metriky

Pro ukázkový graf (Obr. č. 2.28) by výsledek metriky vypadal následovně:

-	Most	Kladno	Praha	Kolín
Most	0	63	96	157
Kladno	63	0	33	102
Praha	96	33	0	69
Kolín	157	102	69	0

Poznámka

Všimněte si, že **grafová vzdálenost** vyšla menší než standardní (přímá) vzdálenost - algoritmus pro hledání nejkratší cesty sám správně spočítal, že např. cesta z Kladna do Kolína nemusí měřit 111km, ale 102km (33+69), pokud by cesta vedla přes Prahu.

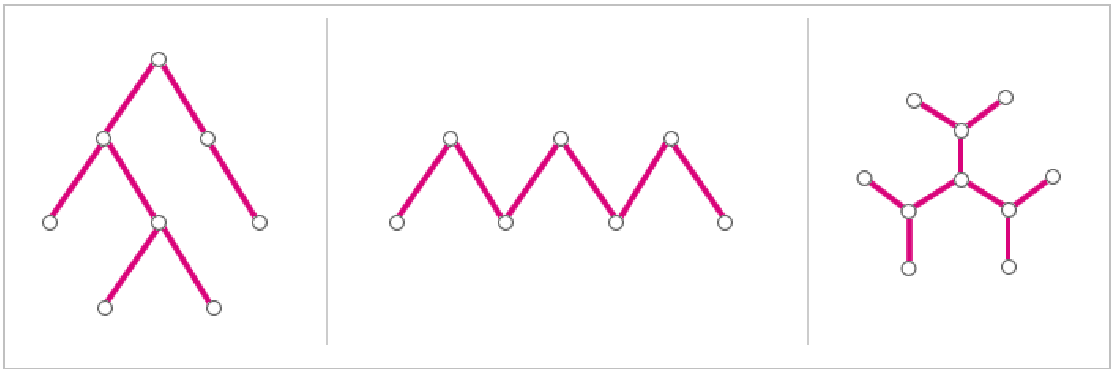
Více informací o algoritmu pro hledání nejkratší cesty naleznete v kapitole 3.

Stromy

Definice

> **Strom** je souvislý graf neobsahující kružnici.

Z definice stromu vyplývá, že mezi každými dvěma vrcholy existuje právě jedna cesta (alespoň jedna cesta, protože je souvislý; nemůže nastat situace více cest, protože díky neexistenci kružnice není možné zvolit "objížďku").



Obr. č. 2.29 - Příklady stromů

Eulerův vzorec (pro stromy)

Pro stromy také platí následující vztah mezi počty hran a vrcholů:

> $|V| = |E| + 1$

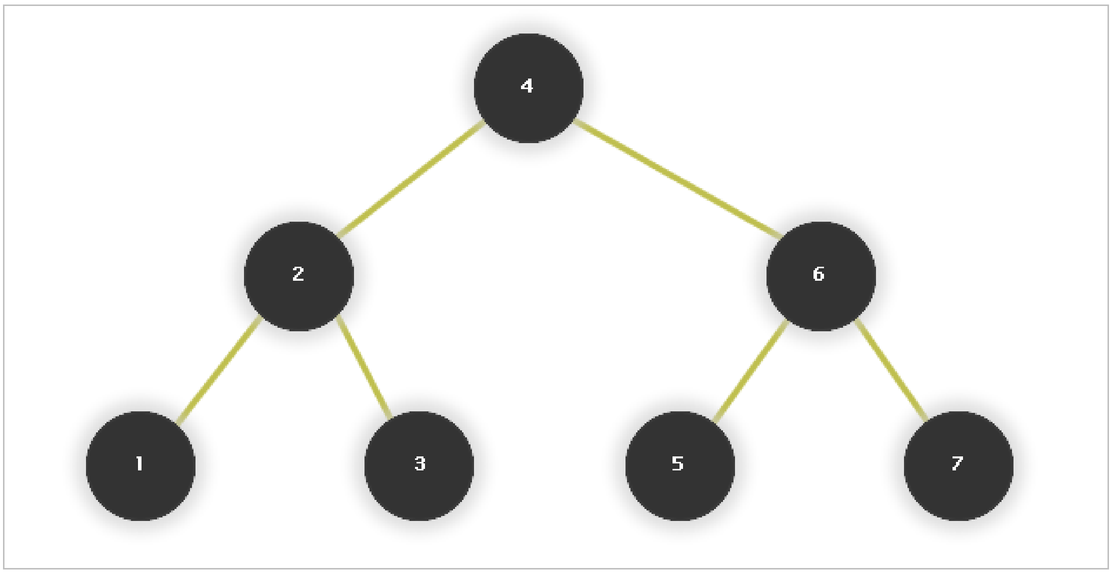
Využití stromů v informatice

Stromy se často využívají pro ukládání dat (typicky čísel) v informatice jako tzv. *datová struktura* (zjednodušeně způsob uložení dat).

Takovou typickou datovou strukturou je **binární vyhledávací strom**. Jde o strom, kde z každého vrcholu vedou nanejvýš tři hrany (jedna "seshora" - říkáme od "otce" - a maximálně dvě "dolů" - k dvěma "synům") a kde pro každý prvek platí, že jeho levý syn je menší nebo roven otci, pravý syn větší. Pojmy levý a pravý zde jsou jednoznačné, musí být splněna podmínka, že směrem vlevo dolů je číslo menší nebo rovno otci, směrem vpravo je číslo větší.

Důvodem, proč se takové struktury používají, je např. rychlejší nalezení prvku, než kdyby byly všechny prvky "za sebou" (v poli, spojovém seznamu).

V tomto typu úloh se nepoužívá matice sousednosti, ale dynamické datové struktury (ukazatelé + *malloc* (C), ukazatelé + *new* (Pascal)) - nechceme ukládat graf pomocí čísel, ale naopak čísla pomocí grafu.



Obr. č. 2.30 - Binární vyhledávací strom

Pro více informací hledejte...

Při špatném použití binárního vyhledávacího stromu může nastat situace, že z každého vrcholu půjde směrem dolů pouze jedna hrana s vyšším prvkem - tím by se ztrácela výhoda rychlého nalezení prvku. Z tohoto důvodu se zavádějí vylepšení binárního vyhledávacího stromu, která se sama starají o tzv. **vyvažování**. Takovými strukturami jsou např. **AVL stromy** (viz [11]).

Kostra grafu

Úvod

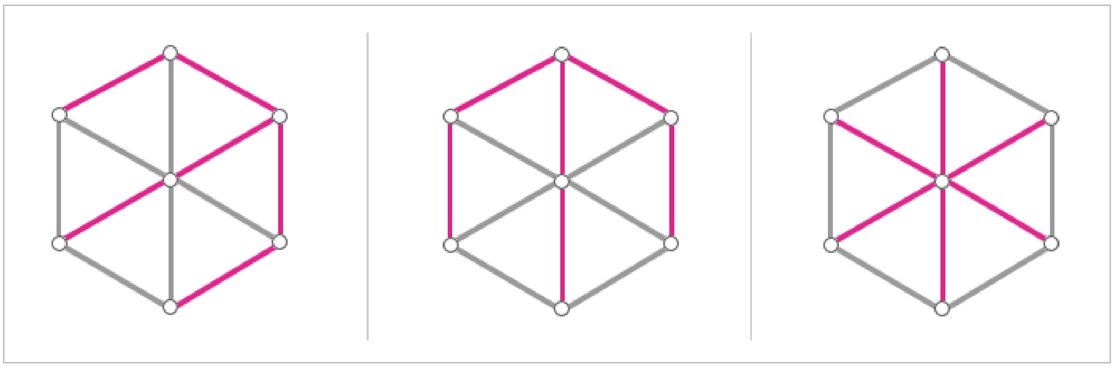
Kostrou grafu budeme rozumět libovolný podgraf, který hranami spojuje **všechny vrcholy** původního grafu a zároveň sám neobsahuje žádnou kružnici (→ jde o strom).

Definice kostry

> Nechť $G = (V,E)$ je graf. Libovolný strom (V,E'), kde $E' \subseteq E$, nazveme **kostrou grafu**.

Příklady

Pokud původní graf (graf, ke kterému hledáme kostru) obsahuje kružnici, pak máme více možností, jak kostru zvolit.



Obr. č. 2.31 - Příklad různých koster stejného grafu

3. Vybrané problémy

Hledání nejkratší cesty v grafu

Úvod

Hledání **nejkratší cesty** je jedním ze základních problémů teorie grafů - podobné algoritmy se používají např. v plánovačích tras v GPS nebo v jízdních řádech.

Mějme souvislý graf s metrikou. Tím jsme získali "mapu" (víme, odkud a kam vedou silnice a jak jsou dlouhé) - jediné informace, které algoritmus potřebuje.

Floyd-Warshallův algoritmus

Jedním z algoritmů pro hledání nejkratší cesty je **Floyd-Warshallův algoritmus**. Jeho výhodou je velmi snadné použití.

Princip algoritmu

Algoritmus postupně projde všechny možnosti cest mezi každými dvěma městy a pokud zjistí, že cesta přes třetí město by byla kratší, použije ji.

> Vzdálenost(A,B) = min(Vzdálenost(A,B), Vzdálenost(A,C)+Vzdálenost(C,B)).

Užitečnou vlastností tohoto algoritmu je také, že najde vzdálenost mezi každými dvěma městy (i pokud mezi nimi původně nevedla přímá silnice).

Implementace

Podobně jako u matice sousednosti v počítači použijeme dvourozměrné pole. Mezi městy, která nejsou spojena silnicí, je použita vysoká konstanta představující nekonečno (v konkrétním případě *999*, konstanta musí být dostatečně vysoké číslo, často se používá *součet délek silnic + 1*); mezi městy, která silnicí spojeny nejsou, bude v matici (poli) uložená délka cesty.

Samotný algoritmus je vlastně užití standardní funkce *minimum ze dvou čísel* spuštěné ve třech *for cyklech* v sobě.

Řešení ve formě programu

```
procedure nejkratsiCesta(var matice:typ_matice);
{ Samotne hledani nejkratsi cesty }
var
 i, j, k: integer;
 {pomocne promenne}
begin
  for i:=1 to velikost_matice do
    begin
    for j:=1 to velikost_matice do
      begin
      for k:=1 to velikost_matice do
        begin
        matice[i,j]:= minimum(matice[i,j], (matice[i,k] + matice[k,j]));
        end; { k }
      end; { j }
    end; { i }
end;
```

Nevýhody

Velkým problémem tohoto algoritmu je ovšem jeho výpočetní složitost - s tím, jak roste počet vrcholů, mezi kterými nejkratší vzdálenost počítáme, roste počet kroků algoritmu podobně rychle jako graf funkce $y = x^3$.

Pro více informací hledejte...

Jedním z lépe použitelných algoritmů je **Dijkstrův algoritmus**. Dosahuje rychlejšího výpočtu, samotná implementace je ovšem náročnější než u Floyd-Warshallova algoritmu.

Výpočetní náročnost algoritmů podle velikosti vstupních dat se v informatice zkoumá pomocí tzv. **(časové) složitosti** algoritmu (viz [9]).

Hledání minimální kostry v grafu

Úvod

Úloha hledání minimální kostry nám popisuje, jak máme spojit **všechny vrcholy grafu** "co nejlevněji" - hranami s nejnižší váhou (ohodnocením). Praktickým využitím mohou být například rozvody elektřiny mezi městy - jak propojit města co nejmenší délkou elektrického vedení.

Definice minimální kostry

Hledání minimální kostry má smysl u ohodnocených grafů. Podobně jako u vzdáleností si zavedeme funkci *w*, která hranám přiřadí číslo - tzv. *váhu*.

Zadání problému pak můžeme matematicky zapsat takto:

> Pro souvislý graf $G = (V,E)$ s nezáporným ohodnocením hran *w* najděte souvislý podgraf (V,E') takový, že výraz:
>
> $$w(E') = \sum_{e \in E'} w(e)$$
>
> nabývá minimální hodnoty.

Kruskalův algoritmus

Kruskalův algoritmus je jedním z algoritmů pro hledání minimální kostry grafu. Pracuje na principu spojování hran s nejmenším ohodnocením, dokud tyto hrany nespojí vrcholy celého grafu. Díky jeho snadnému postupu jej lze snadno použít i bez počítače při "ručním výpočtu" – v části věnované procvičování (kapitola 4) si můžete zkusit některé úlohy vyřešit sami za použití zvýrazňovače.

Popis Kruskalova algoritmu

- Všechny hrany si seřadíme podle velikosti (vzestupně - od hrany s nejmenší váhou).
- Hranu s nejmenší váhou použijeme jako první hranu kostry.
- Pokud jsme tím už vytvořili kostru (graf měl jen dva vrcholy), končíme.
  V opačném případě vezmeme hranu s druhou nejmenší váhou a přidáme ji do kostry.
  POZOR! Pokud by nám v grafu vznikla kružnice, hranu nepoužijeme.

- Opakujeme minulý krok, dokud vznikající kostra nespojí všechny vrcholy grafu.

Kruskalův algoritmus matematicky

> Je dán souvislý graf $G = (V,E)$ s n vrcholy, m hranami a s ohodnocením w.
>
> Očíslujme hrany $e_1, e_2,\dots,e_m$ tak, aby:
>
> $w(e_1) \leq w(e_2) \leq \dots \leq w(e_m)$.
>
> Nyní budeme postupně konstruovat množiny hran $E_0, E_1, E_2,\dots \subseteq E$.
>
> Položme $E_0 = \emptyset$.
>
> Byla-li již nalezena množina E_{i-1}, určíme množinu E_i následovně:
>
> $$E_i = \begin{cases} E_{i-1} \cup \{e_i\} & \text{neobsahuje-li graf } (V, E_{i-1} \cup \{e_i\}) \text{ kružnici} \\ E_{i-1} & \text{jinak.} \end{cases}$$
>
> Algoritmus se zastaví, jestliže buď E_i již obsahuje $n-1$ hran, nebo $i = m$, tj. probraly se všechny hrany grafu G. Nechť E_t značí množnu, pro níž se algoritmus zastavil, a nechť T značí graf (V,E_t). T je pak hledanou minimální kostrou.

Kruskalův algoritmus je tzv. *hladový*.

Co jsou hladové algoritmy?

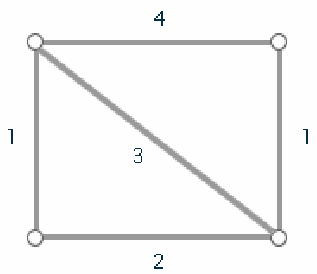
Za **hladové** algoritmy jsou označované takové algoritmy, které vždy volí v danou chvíli nejvýhodnější volbu (aniž by se snažily "myslet do budoucnosti").
Anglicky jsou označovány *greedy search* (*greedy* = chamtivý).

Příklad špatného užití podobného principu by bylo např. při hraní šachů - podobným postupem by hráč téměř jistě prohrál. Naopak u některých úloh (zmiňované hledání minimální kostry) funguje takový algoritmus úspěšně.

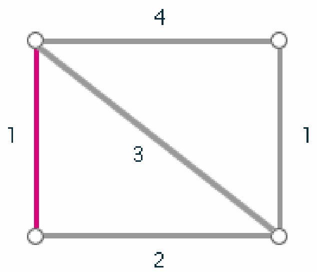
Animace č. 1

(celá animace je zobrazena krok po kroku na přiloženém CD)

Krok 1/4

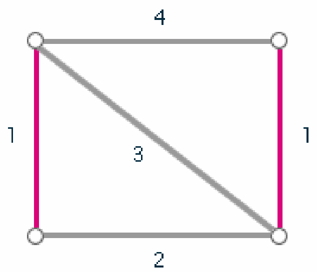


Krok 2/4



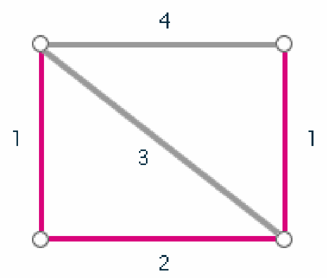
Hrany s váhou 1 jsou nejmenší, kostra začne vznikat z nich.

Krok 3/4



Hrany s váhou 1 jsou nejmenší, kostra začne vznikat z nich.

Krok 4/4

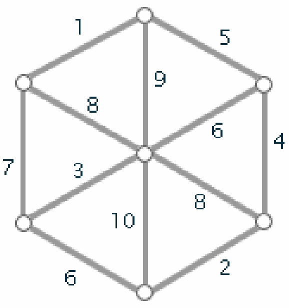


Následuje hrana s váhou 2. Kostra spojila všechny vrcholy – konec.

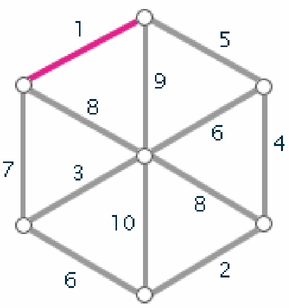
Animace č. 4

(pro uvedení složitějšího příkladu i v tištěné verzi)

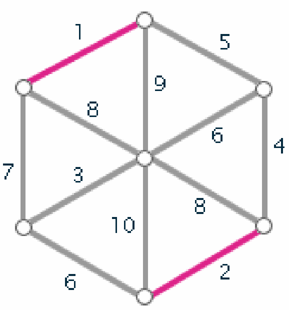
Krok 1/7



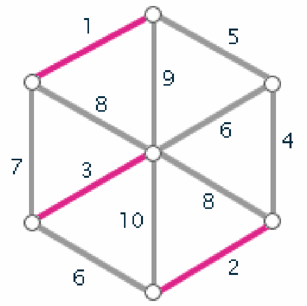
Krok 2/7



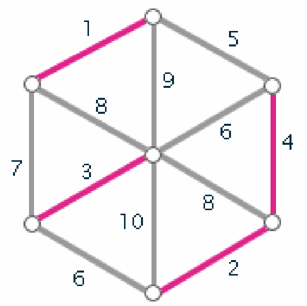
Krok 3/7



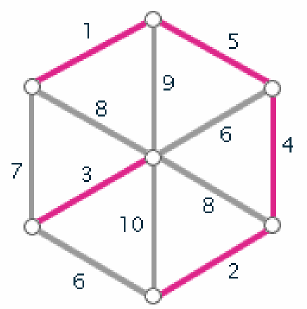
Krok 4/7

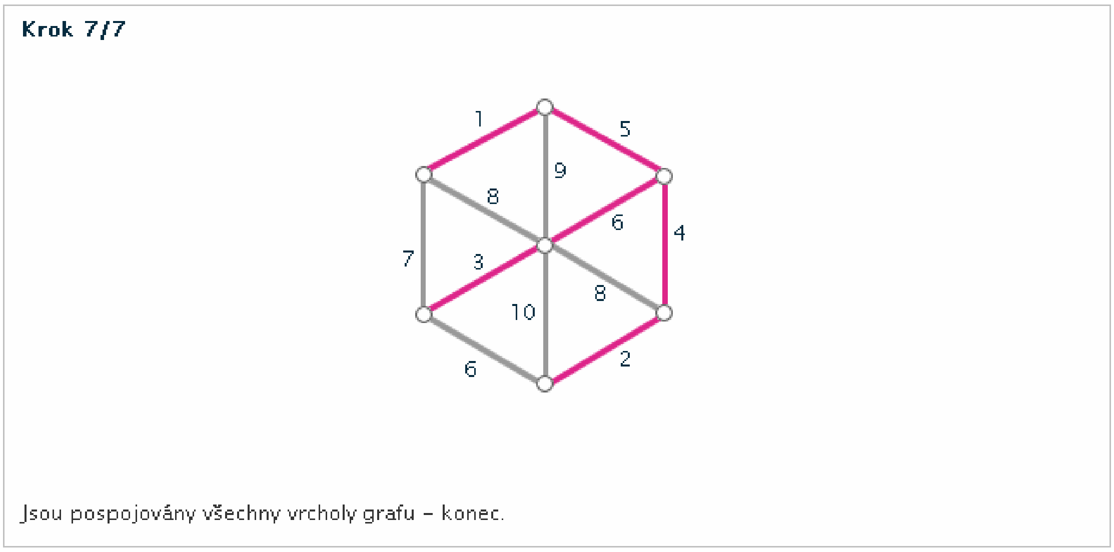


Krok 5/7



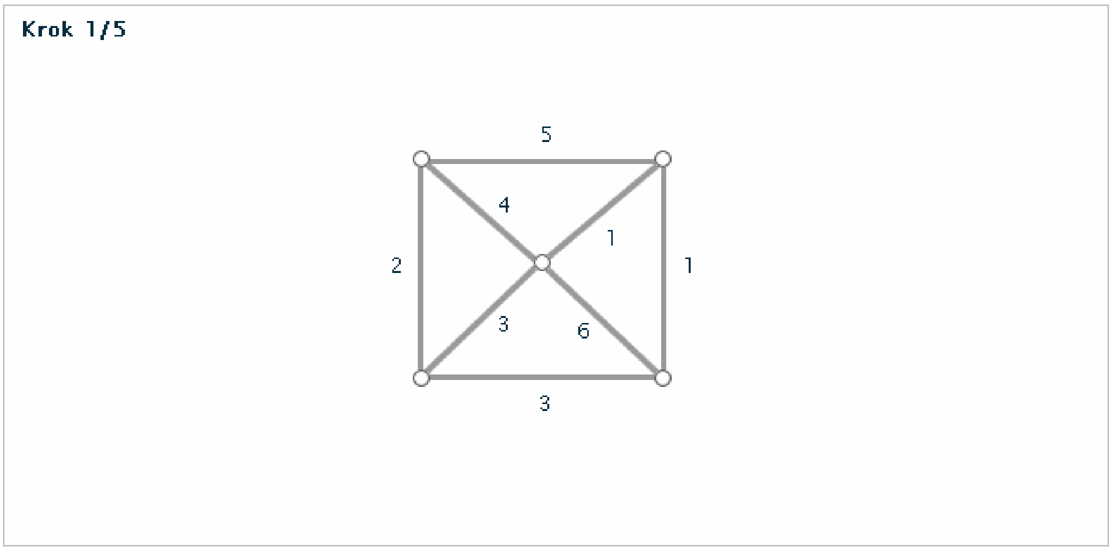
Krok 6/7

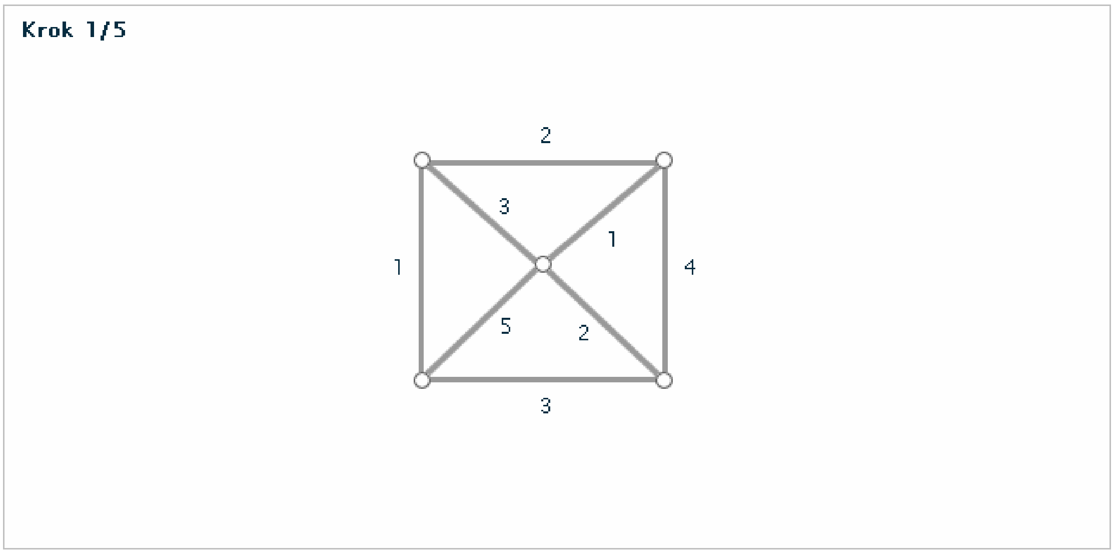




Animace č. 2 a 3

(celé animace jsou zobrazeny krok po kroku na přiloženém CD)





Počty koster grafu

Pro počet koster grafu neplatí žádné jednoduché univerzální pravidlo - většinou si musíme všechny možnosti představit podobně jako obdobné příklady v kombinatorice.

Platí však několik speciálních pravidel podle toho, o jaký graf se jedná:

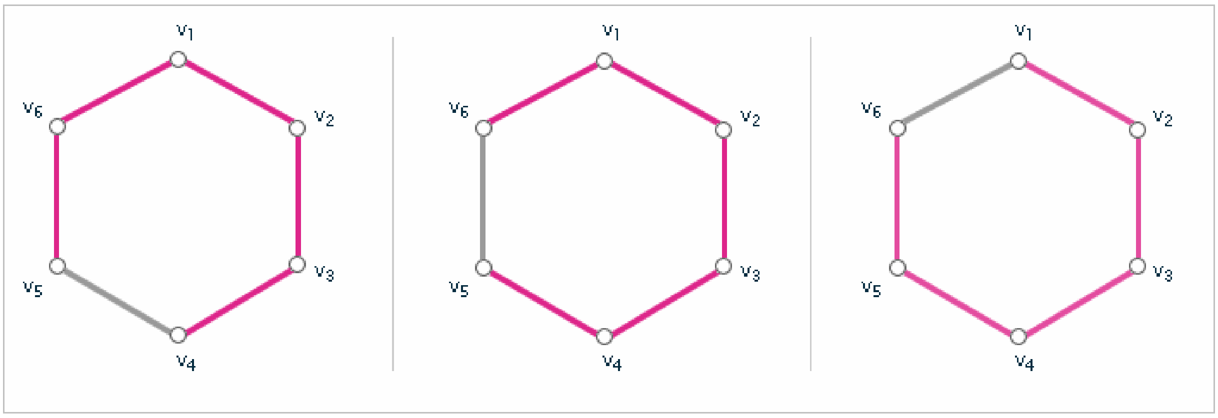
Počty koster úplného grafu (Cayleyho formule)

Pomocí *Cayleyho formule* můžeme určit počet stromů na daných *n* vrcholech - tedy počet koster úplného grafu.

> Pro každé $n \geq 3$ je počet stromů na daných n vrcholech roven n^{n-2}.

Počty koster kružnice

> Pokud je graf kružnicí, je počet koster grafu roven $|V|$ ($|V| = |E|$, můžeme vynechat vždy právě jednu hranu).



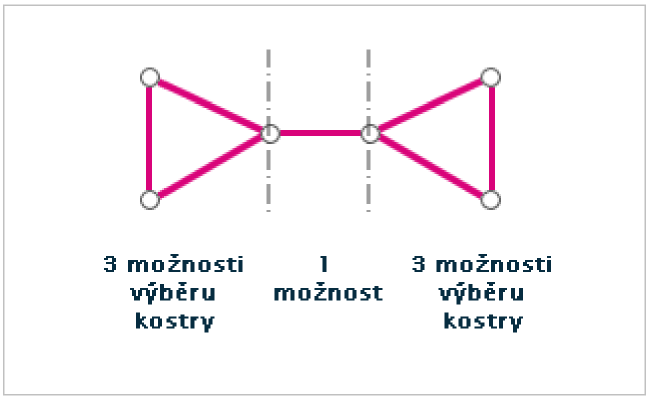
Obr. č. 3.1 - Příklad různých koster kružnice

Počty koster stromu

> Pokud je graf stromem, má právě jednu kostru (samotný graf je kostrou).

Příklad

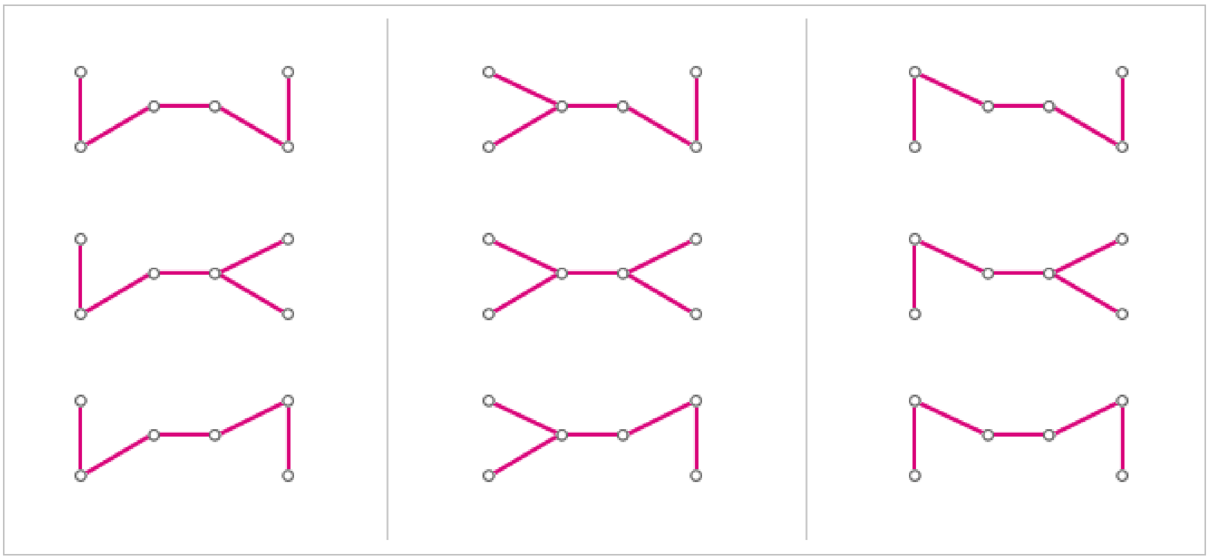
Budeme-li chtít zjistit, kolik různých koster má následující graf (obr. č. 3.2), můžeme si ho rozdělit na jednotlivé podgrafy a výsledek pak získáme vynásobením počtů koster jednotlivých podgrafů (kombinatorické pravidlo součinu).



Obr. č. 3.2 - Příklad výpočtu počtu koster pomocí podgrafů

Na obr. č. 3.2 máme u prvního podgrafu (podgrafy jsou odděleny přerušovanými čarami) 3 možnosti, jak zvolit kostru (jde o kružnici). V druhém podgrafu je po pouze jedna možnost výběru hrany (pokud bychom tuto jedinou hranu nevybrali, graf by přestal být souvislým a tím by porušil definici kostry). V třetím podgrafu máme opět 3 možnosti.

Výsledkem je tedy **součin 3×1×3 = 9 –** máme 9 různých koster grafu.



Obr. č. 3.3 - Všech různé kostry daného grafu

Úlohy na hledání počtu možných koster grafu naleznete v kapitole 4.

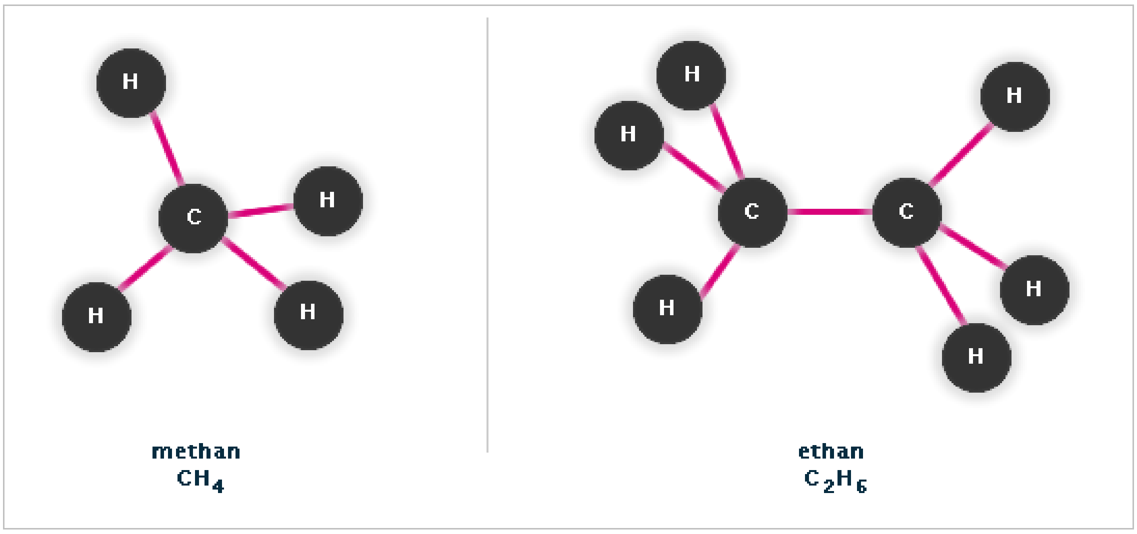
Alkany

Úvod

Pomocí grafů můžeme také popisovat chemické sloučeniny. Za vrcholy grafu použijeme jednotlivé atomy, hrana bude znázorňovat chemickou vazbu mezi nimi. (Pro znázornění problematiky nám postačí uvažovat pouze jednoduchou vazbu.)

Na následujícím příkladu si lze pomocí teorie grafů ukázat, proč strukturní vzroce alkanů jsou stromy.

> Alkany: C_nH_{2n+2}



Obr. č. 3.4 - Příklady dvou nejjednoduších alkanů (methan, ethan)

Co jsou alkany?

Alkany jsou uhlovodíky neobsahující žádné násobné vazby mezi svými molekulami. Obsahuje-li molekula více než čtyři atomy uhlíku, může vznikat více **izomerů** (molekul se stejným souhrnným vzorcem, ale různými strukturními vzorci).

Postup

- Molekuly alkanů tvoří stromy (jsou souvislé a neobsahují cyklus).
- Pro každý strom platí Eulerův vzorec: $|V| = |E| + 1$
- Nyní ověříme, že graf popisující alkan se vzorcem C_nH_{2n+2} je stromem.

- Molekuly vodíku a uhlíku znázorňujeme jako vrcholy, tzn. $|V| = n + (2n + 2) = 3n + 2$.
- $|E|$ je rovno počtu vazeb. Z každé molekuly uhlíku vycházejí 4 vazby (tj. celkem 4n hran) a z každé molekuly vodíku vychází jedna hrana (tj. celkem 2n+2 hran).
- Tímto postupem jsme každou vazbu započetli dvakrát, proto musíme výsledek vydělit dvěma:

$$|E| = \frac{4n + (2n+2)}{2} = \frac{6n+2}{2} = 3n + 1$$

- Ukázali jsme, že alkany splňují Eulerův vzorec , tedy jejich vzorce můžeme v teorii grafů považovat za stromy.

Jednotažky (eulerovské grafy)

Úvod

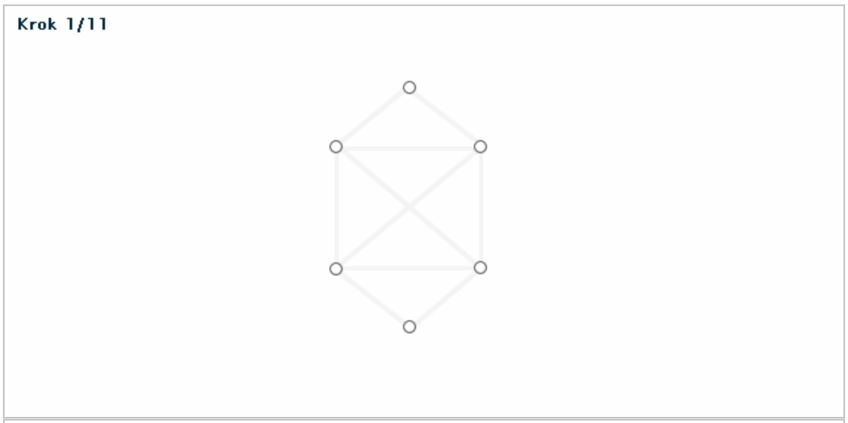
Kreslení grafů jedním tahem je jednou ze základních úloh teorie grafů - viz Eulerova úloha o procházení po mostech. Pro zjednodušení si úlohu předvedeme na neorientovaných grafech, podobný problém lze řešit i na orientovaných grafech.

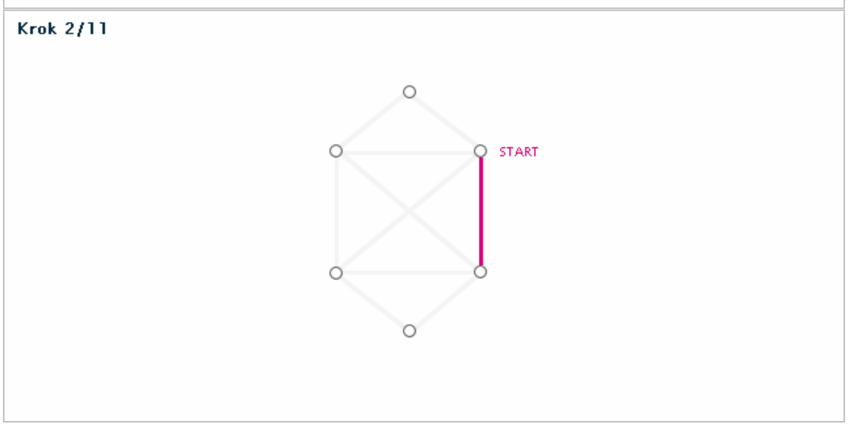
Zadání

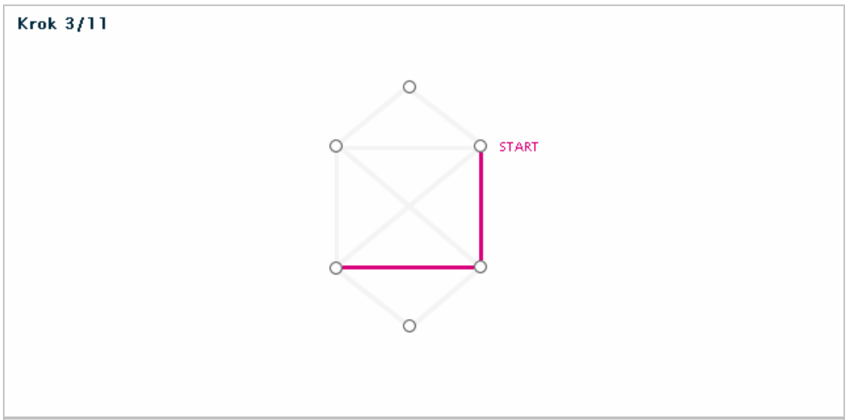
> Nakreslete daný graf $G = (V,E)$ jedním uzavřeným tahem bez zvednutí tužky papíru (žádna hrana se neobtahuje dvakrát). Tento tah začíná i končí ve stejném vrcholu.

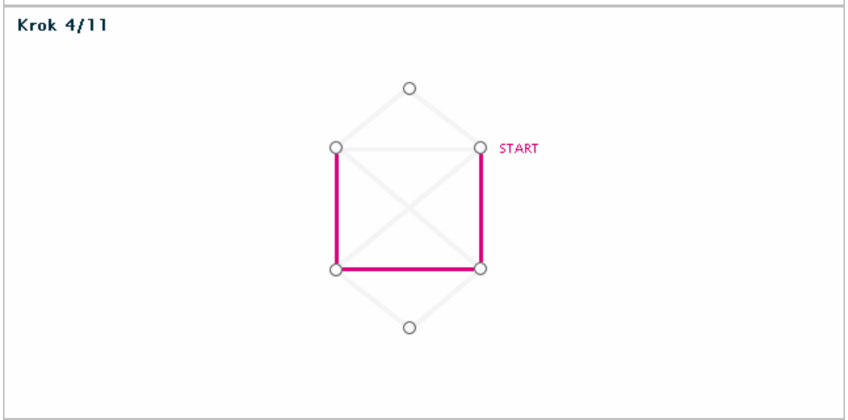
Animace

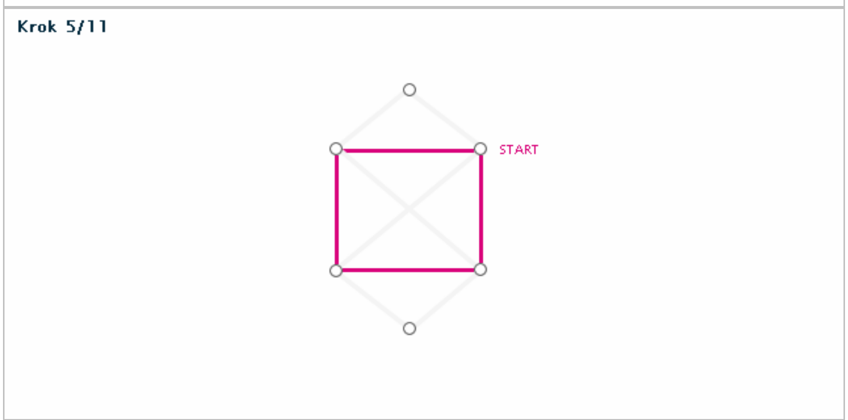
(celá animace je zobrazena krok po kroku na přiloženém CD)

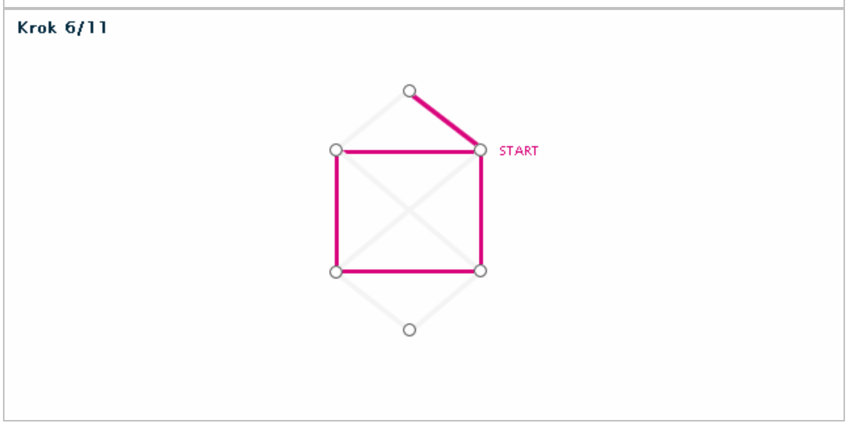


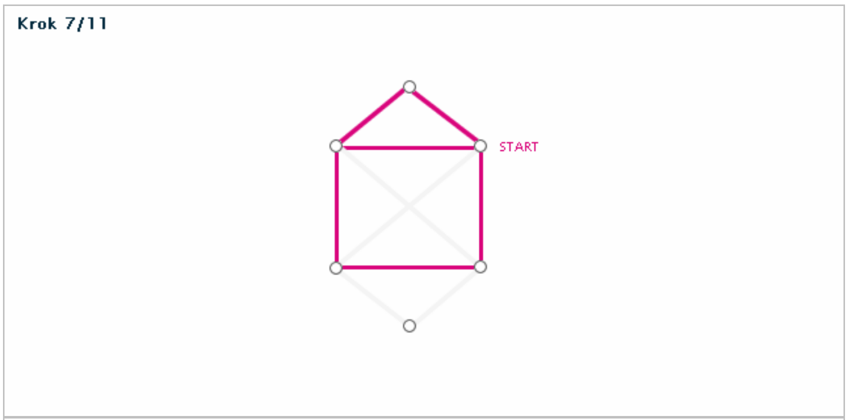


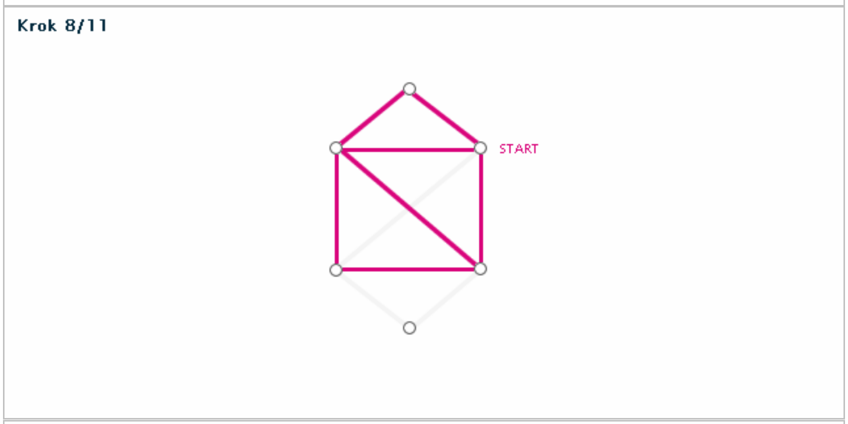

Krok 3/11
START

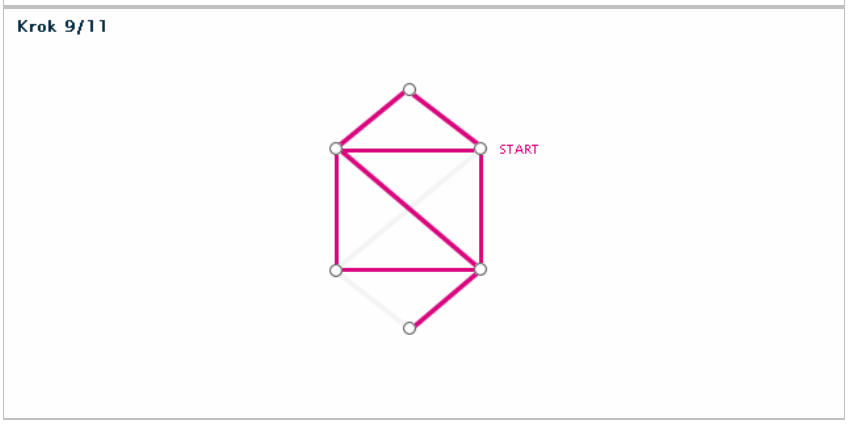

Krok 4/11
START

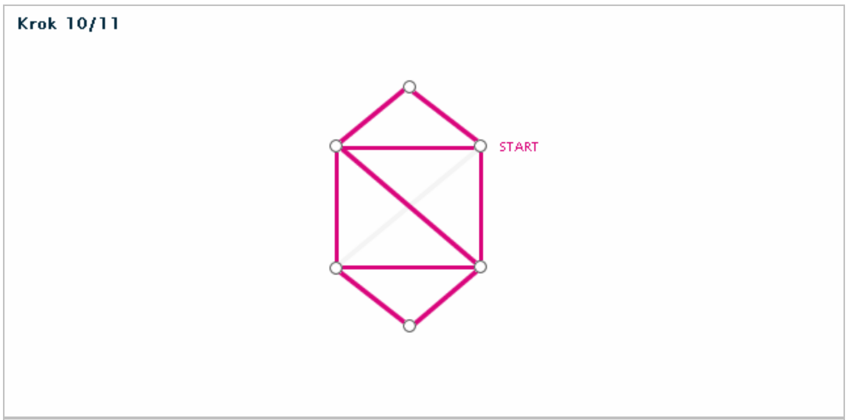

Krok 5/11
START

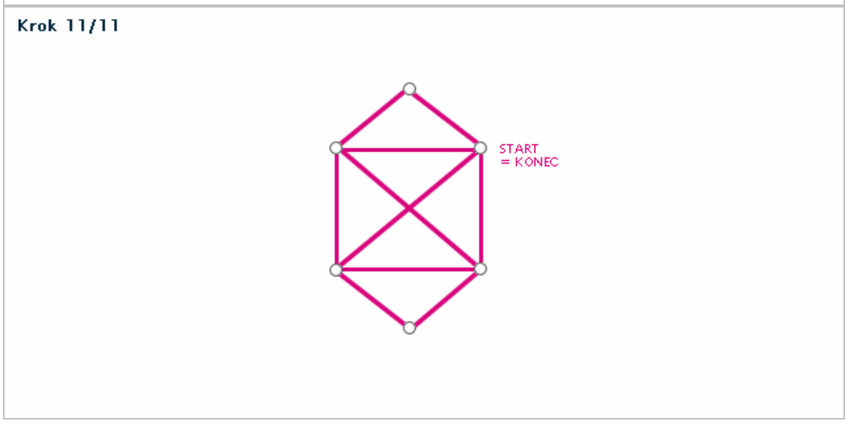

Krok 6/11
START


Krok 7/11
START


Krok 8/11
START


Krok 9/11
START





Poznámka

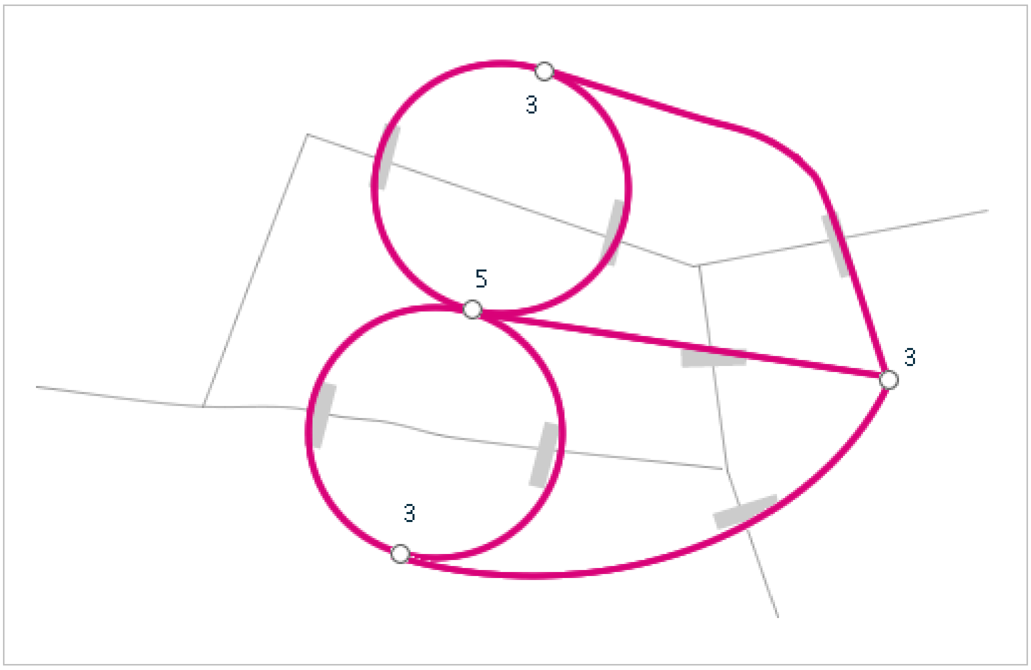
Nalezený tah **není jediný možný**. Tah může začínat (resp. končit) v libovolném vrcholu.

Návod

> Graf *G* je eulerovský, právě když je souvislý a každý vrchol *G* má sudý stupeň.

Eulerova úloha Sedm mostů Königsbergu

Předchozí věta také vysvětluje, proč je úloha *Sedmi mostů města Königsbergu* (viz historie teorie grafů, kapitola 1) neřešitelná - ačkoliv je graf souvislý, neplatí, že by všechny vrcholy měly sudý stupeň (naopak všechny mají lichý stupeň).

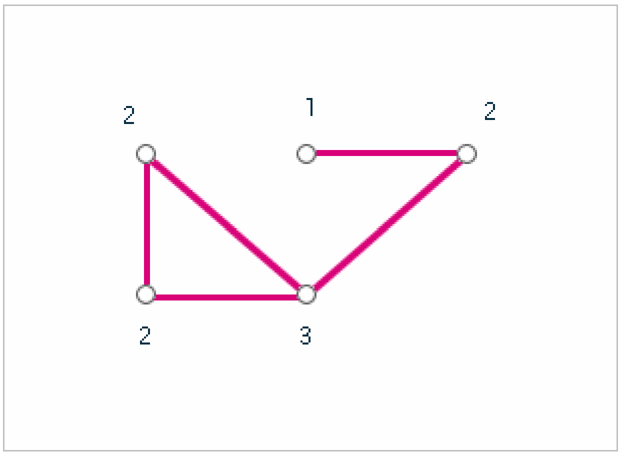


Obr. č. 3.5 - Proč nelze vyřešit úlohu Sedmi mostů města Königsbergu?

Otevřené tahy

V literatuře také můžeme najít termín **otevřený eulerovský tah** - je definován stejně jako tah uzavřený, pouze se nevrací do původního bodu, kde začal.

> Jsou-li v grafu právě dva vrcholy lichého stupně, pak existuje otevřený eulerovský tah začínající v jednom z těchto vrcholů a končící v druhém. [5]



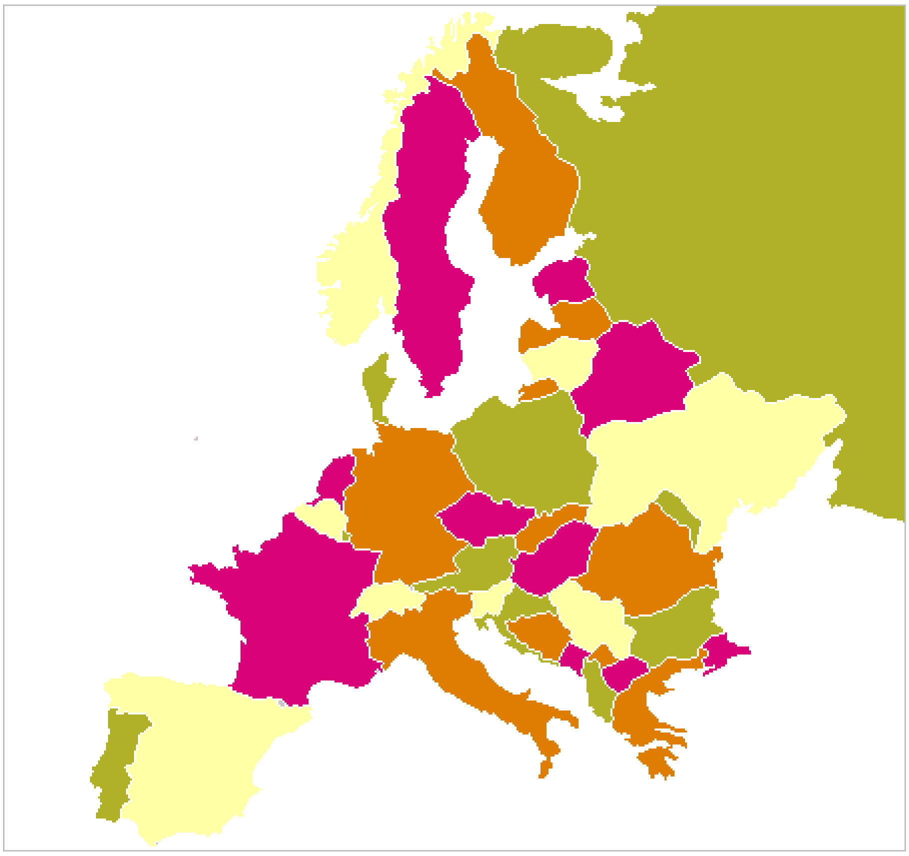
Obr. č. 3.6 – Otevřený eulerovský tah

Barvení mapy

Úvod

Pomocí teorie grafů lze dokázat, že pro obarvení libovolné mapy tak, aby dvě sousední země nebyly obarveny stejnou barvou, nám postačí čtyři barvy.

Příklad mapy obarvené čtyřmi barvami



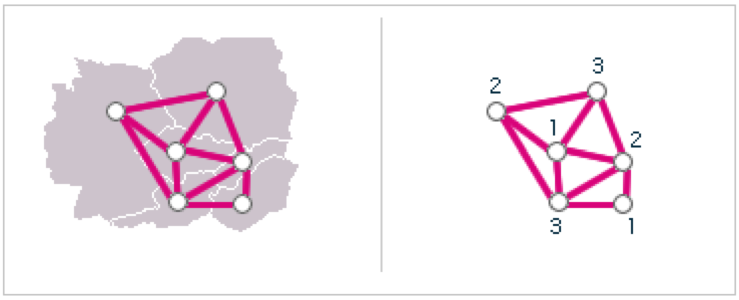
Obr. č. 3.7 - Politická mapa obarvená čtyřmi barvami (viz [8])

Spojitost s teorií grafů

Pro řešení tohoto problému použijeme podobný princip jako u Eulerovy úlohy - každý stát si představíme jako vrchol grafu a hranou spojíme státy, které spolu sousedí. Místo obarvení si také můžeme představit dosazování čísel k vrcholům. Díky tomu lze také formulovat problém matematicky (následuje).

Poznámka

Barvení grafu se řeší pro souvislé grafy, proto jsme z mapy vymazali ostrovy (např. Velkou Británii).



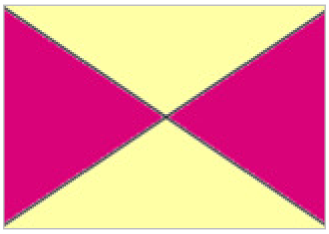
Obr. č. 3.8 - Spojitost mezi barvením mapy a grafem, který této mapě odpovídá

Zadání matematicky

> Nechť $G = (V,E)$ je graf, k přirozené číslo. Zobrazení b: $V \rightarrow \{1,2,\ldots,k\}$ nazveme **obarvením grafu G** pomocí k barev, pokud pro každou hranu $\{x,y\} \subseteq E$ platí $b(x) \neq b(y)$.

Poznámka

Někdy může nastat situace, že státy sousedí jedním bodem. V takovém případě není nutné, aby měly všechny státy různé barvy - vždy pouze po dvou, sdílí-li delší hranici.



Obr. č. 3.9 - Státy sousedící jedním bodem

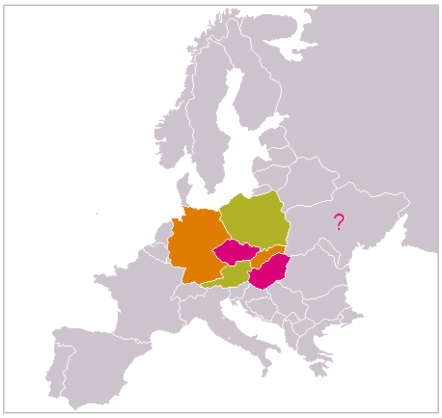
Proč čtyři barvy?

Nejprve se podívejme, proč potřebujeme **nejméně 4 barvy**:

- **Mějme 1 barvu:** Úlohu bychom nemohli vyřešit. Mají-li mít sousední státy rozdílné barvy, nemohli bychom obarvit ani dva sousedy (např. ČR a SR).
- **Mějme 2 barvy:** Podobná situace - dejme ČR první barvu, SR i Německu druhou. Polsko ovšem sousedí i s ČR (barva 1), SR (barva 2) i Německem (barva 2). Proto potřebujeme třetí barvu.

- **Mějme 3 barvy:** Na obrázku č. 3.10 vidíme, že pro Ukrajinu (označena otazníkem) se nám opět nedostává barev - máme již obarvené tři sousedy (každý má jinou barvu) - znovu potřebujeme další, čtvrtou barvu.

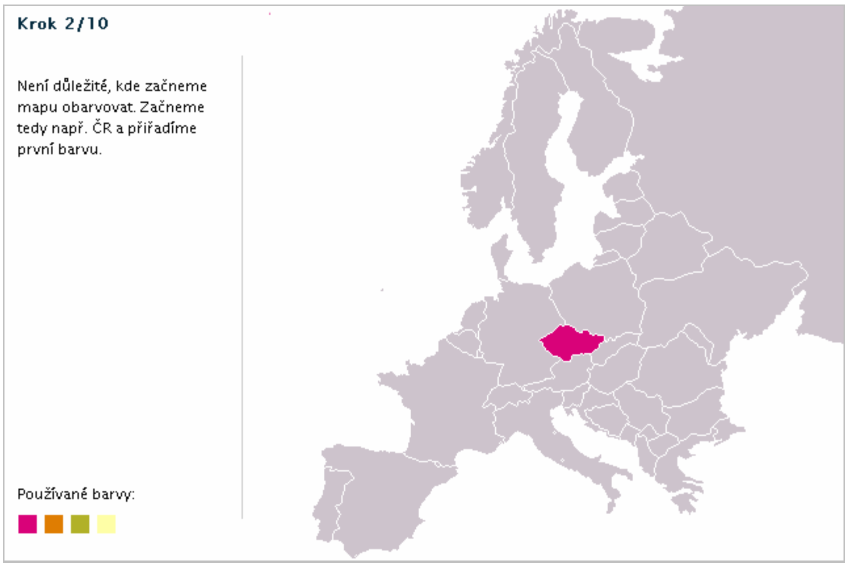
Proč nám čtyři barvy už stačí? - viz rámeček *Důkaz*....

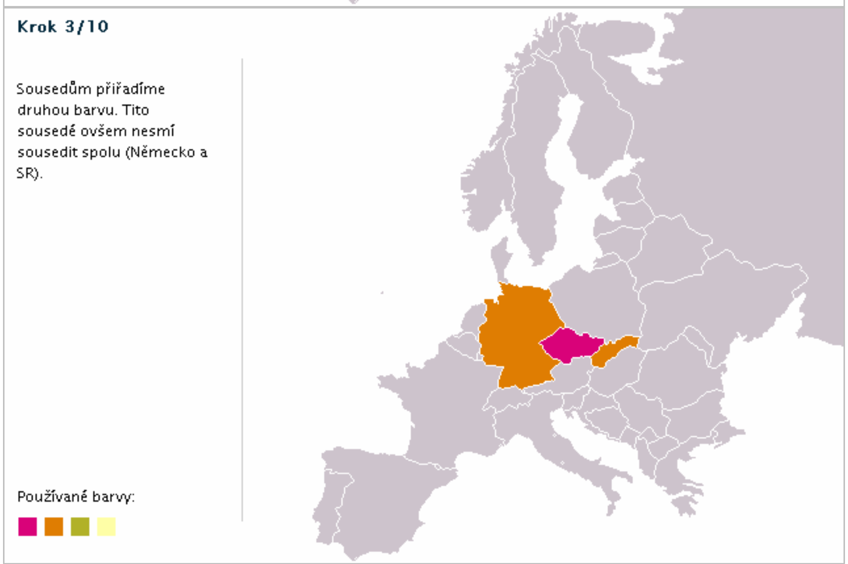


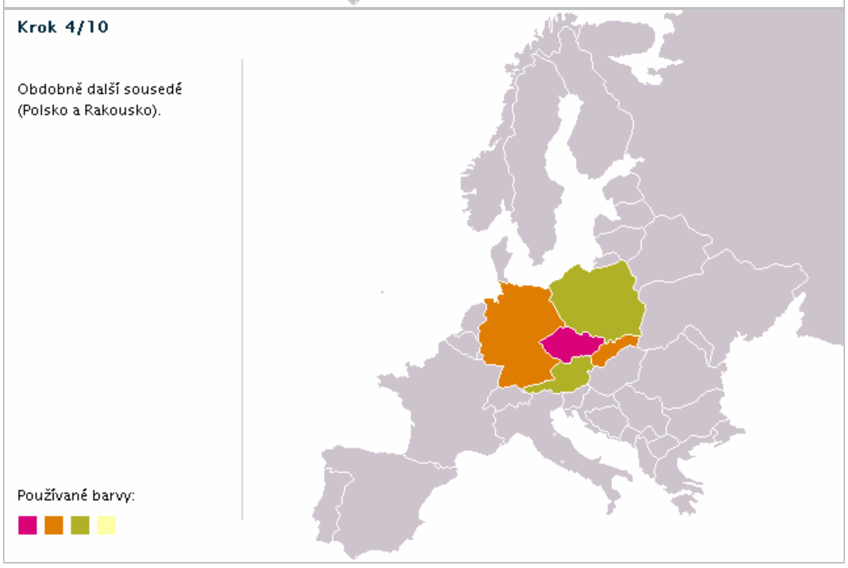
Obr. č. 3.10 - Proč potřebujeme při barvení mapy čtyři barvy?

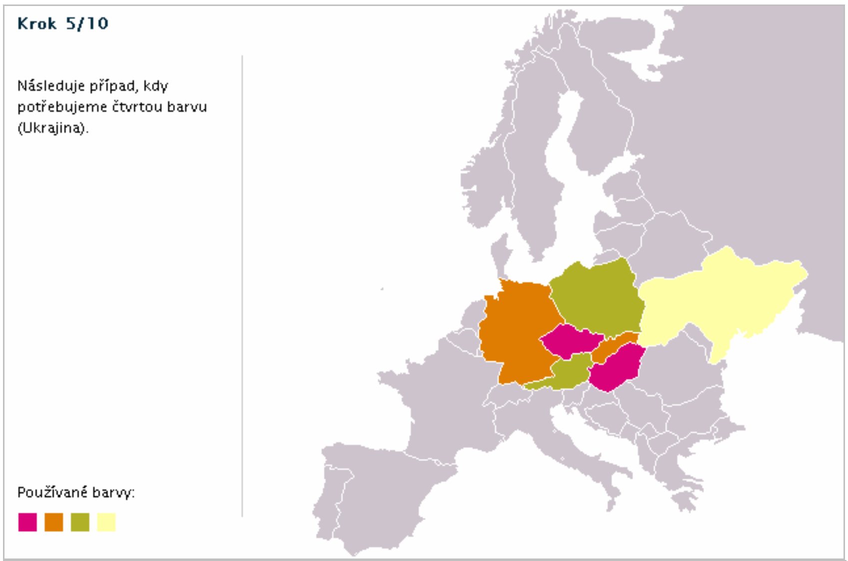
Animace

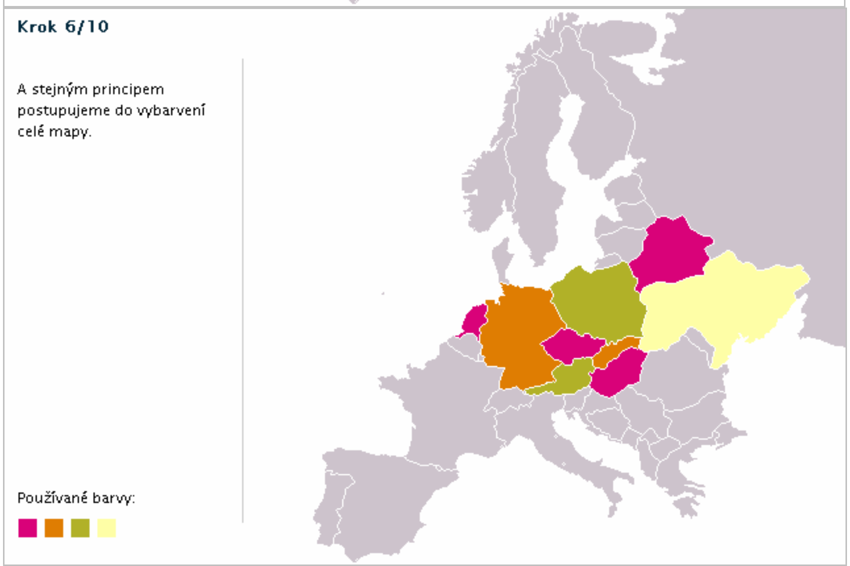
(celá animace je zobrazena krok po kroku na přiloženém CD)

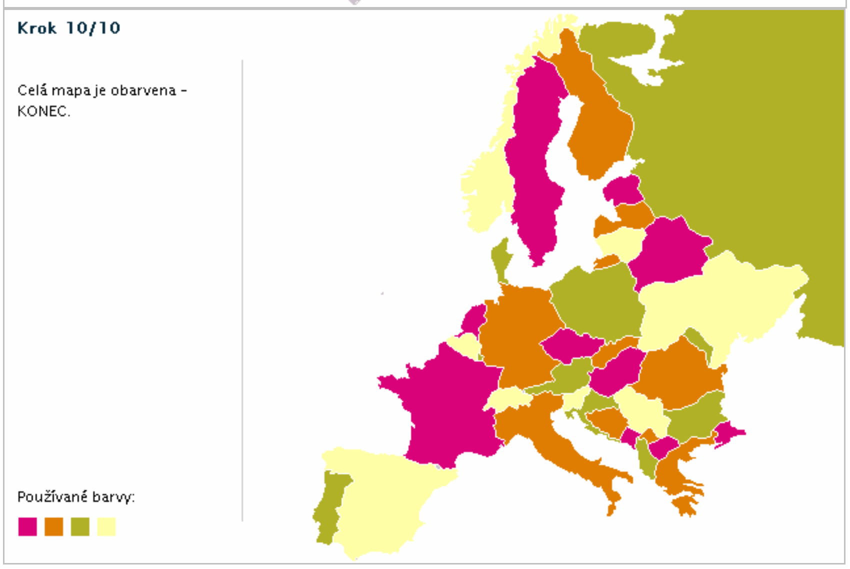

Krok 2/10
Není důležité, kde začneme mapu obarvovat. Začneme tedy např. ČR a přiřadíme první barvu.
Používané barvy:


Krok 3/10
Sousedům přiřadíme druhou barvu. Tito sousedé ovšem nesmí sousedit spolu (Německo a SR).
Používané barvy:


Krok 4/10
Obdobně další sousedé (Polsko a Rakousko).
Používané barvy:


Krok 5/10
Následuje případ, kdy potřebujeme čtvrtou barvu (Ukrajina).
Používané barvy:


Krok 6/10
A stejným principem postupujeme do vybarvení celé mapy.
Používané barvy:


Krok 10/10
Celá mapa je obarvena - KONEC.
Používané barvy:

Důkaz, že pro obavení mapy stačí čtyři barvy

Dokázat, že pro obarvení libovolné mapy stačí čtyři barvy, je matematicky poměrně náročný. Problém byl vyřešen v 70. letech 20. století, přičemž velká část problému byla dokázána pomocí počítače - viz [12] a [13].

Důkaz jednoduššího tvrzení (pro obarvení stačí 5 barev) můžete najít v knize [1], str. 214.

Praktické využití

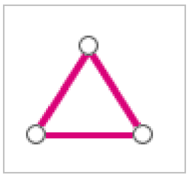
Podobný problém (na složitější úrovni) se řeší v mobilních sítích. Oblast je rozdělena na mnoho malých buňek, ve kterých telefony komunikují se základnovými stanicemi na určité frekvenci. Snahou operátorů je využít co nejmenší počet frekvencí (odpovídající co nejmenšímu počtu barev) a přitom respektovat nutnou podmínku, že sousední buňky nemohou být nastaveny na stejnou frekvenci (což by odpovídalo obarvení sousedních států stejnou barvou).

4. Procvičování

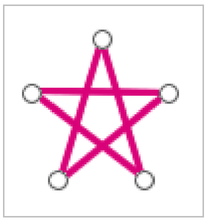
(Řešení všech úloh je uvedeno na konci kapitoly.)

Základní pojmy

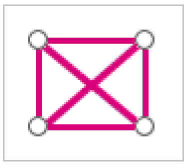
1. Je následující graf úplný?



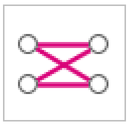
2. Je následující graf úplný?



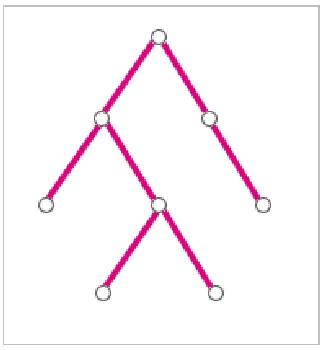
3. Je následující graf bipartitní?



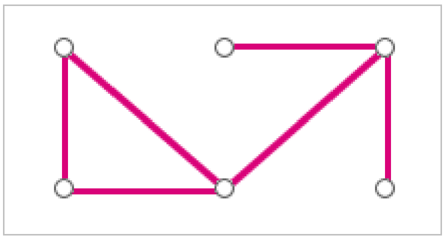
4. Je následující graf bipartitní?



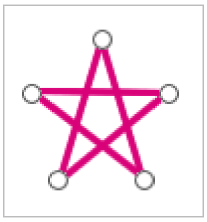
5. Obsahuje graf cyklus?



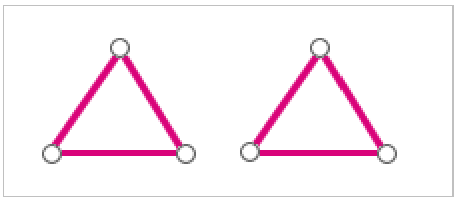
6. Obsahuje graf cyklus?



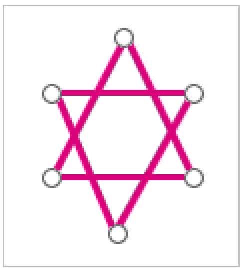
7. Je graf souvislý?



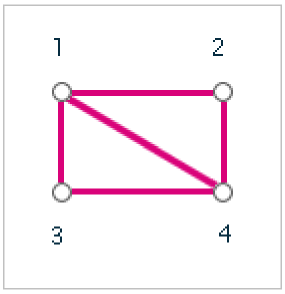
8. Je graf souvislý?



9. Je graf souvislý?



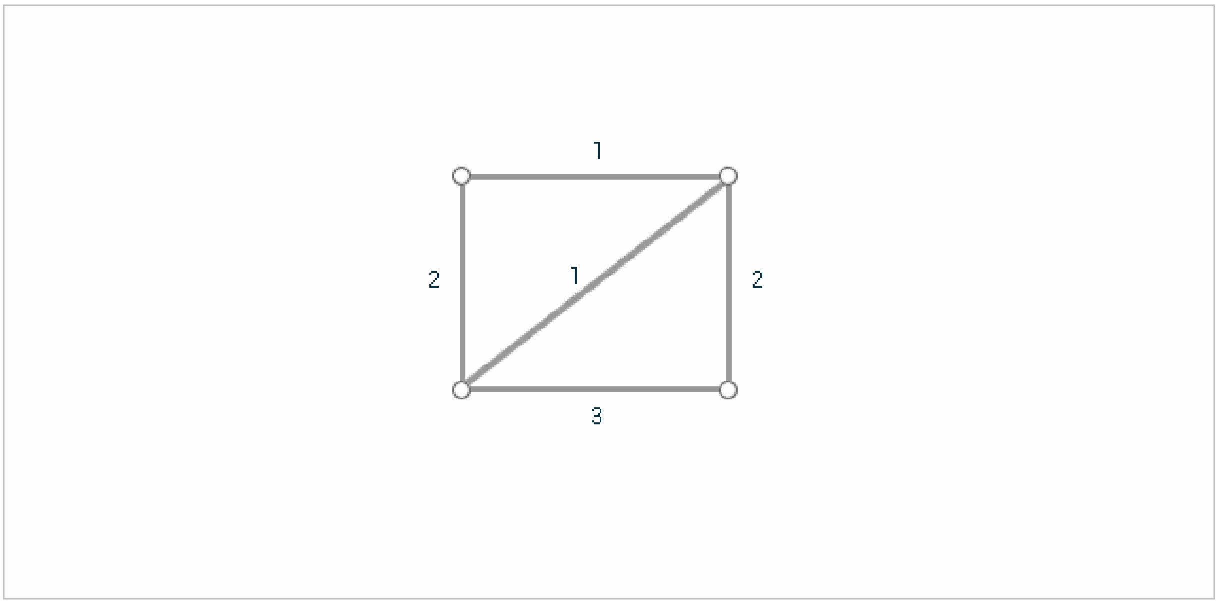
10. Jak bude vypadat matice sousednosti grafu?



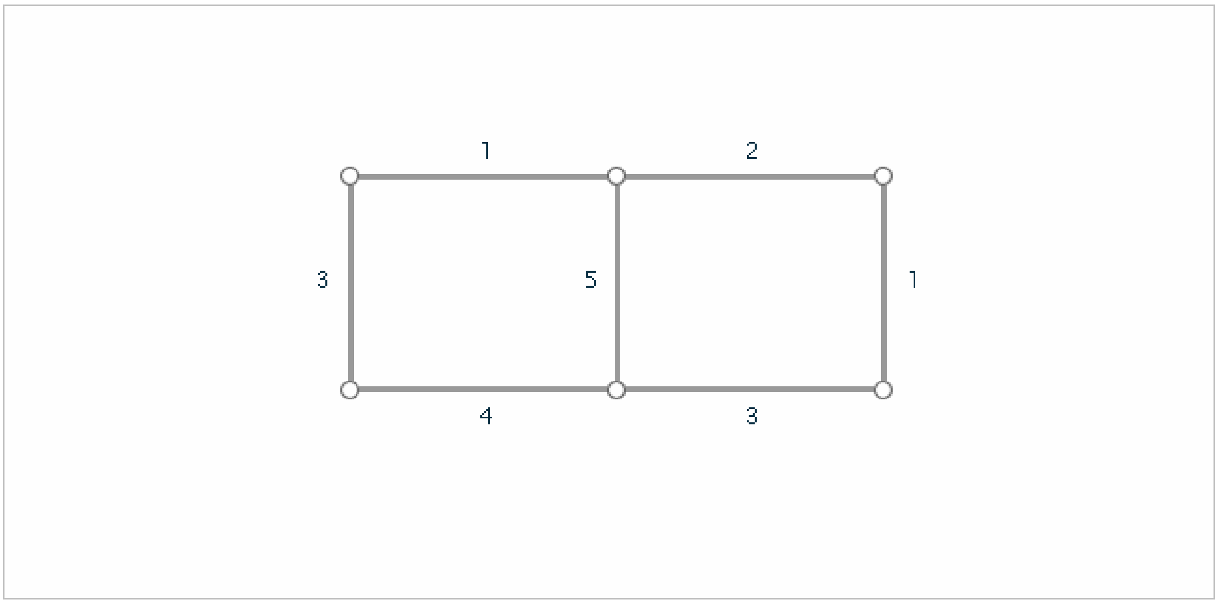
Minimální kostra

Nalezněte minimální kostry grafů:

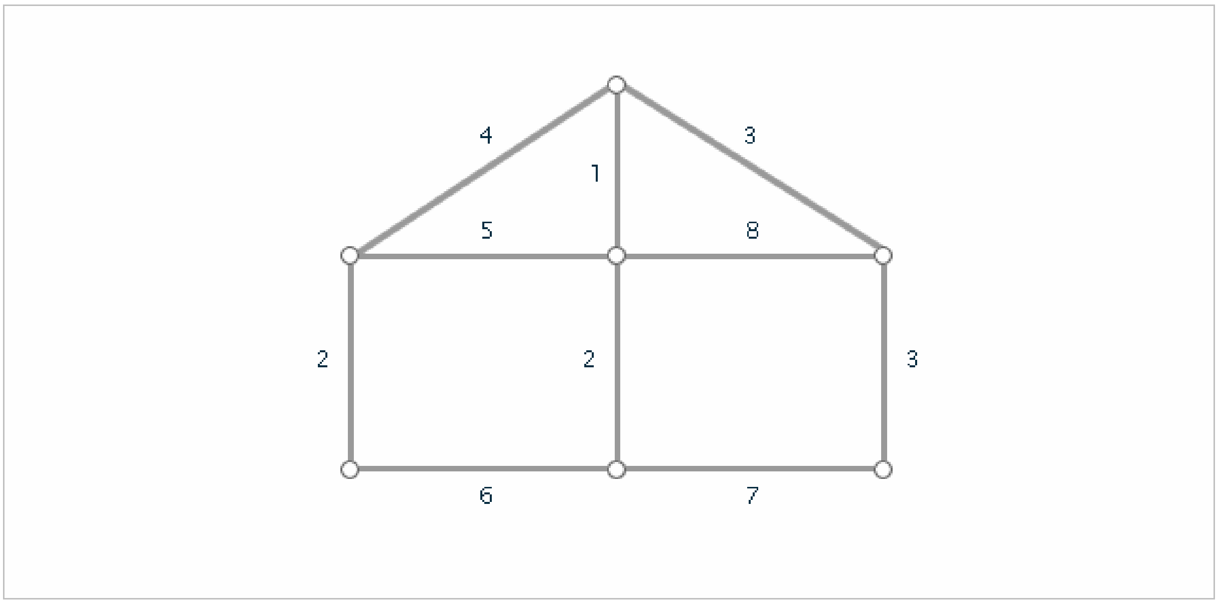
1.



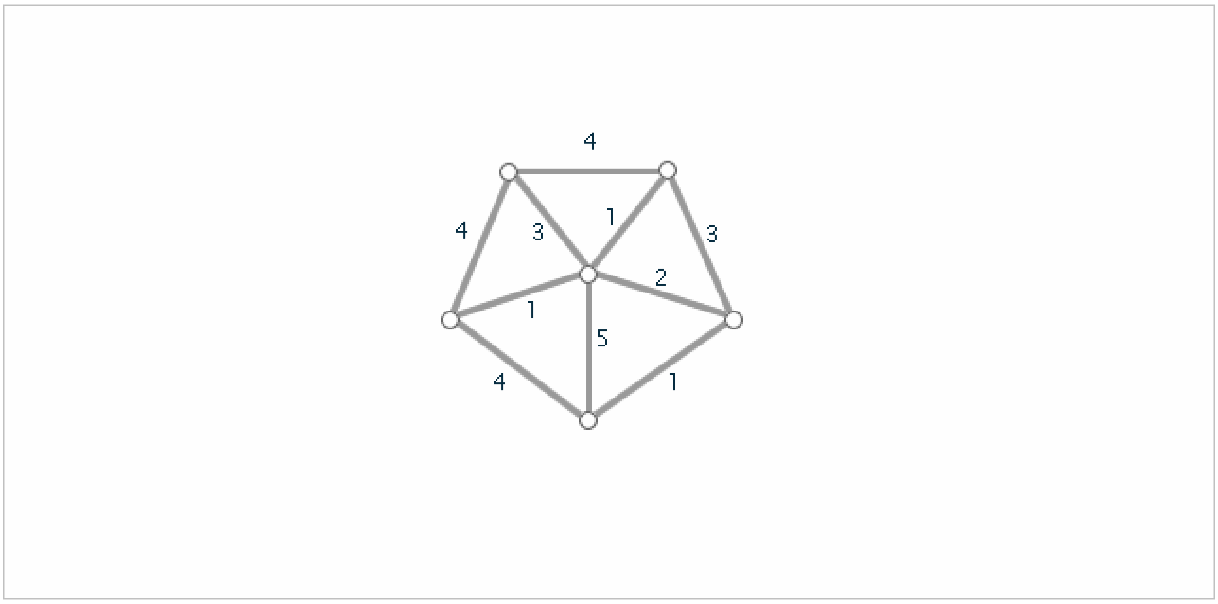
2.



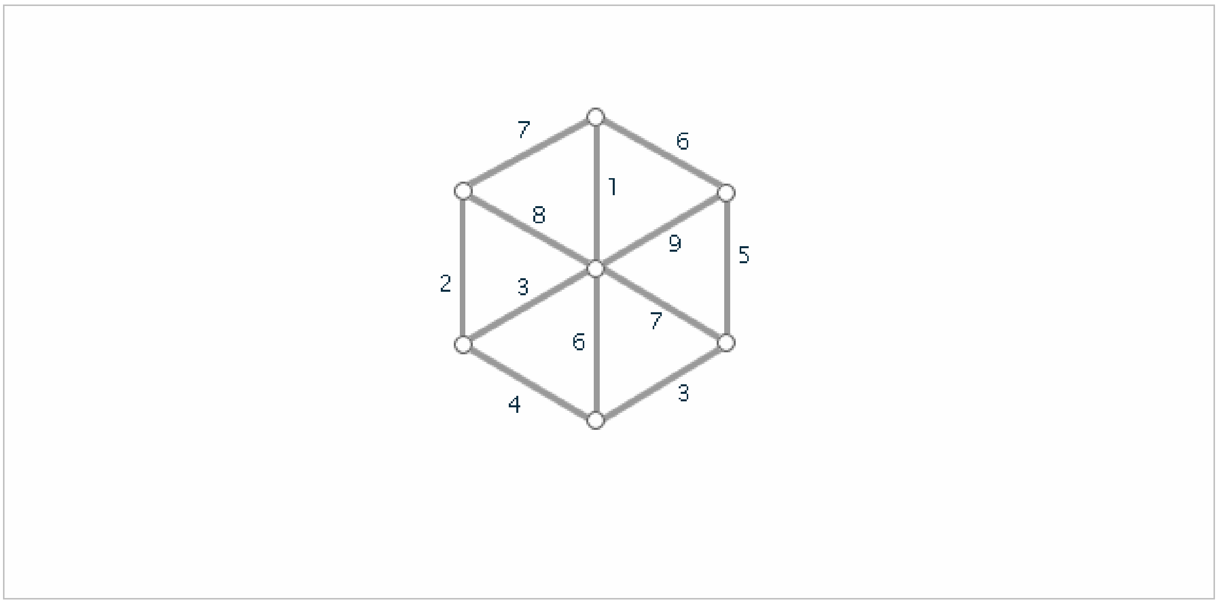
3.



4.



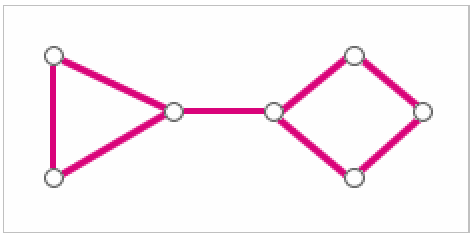
5.



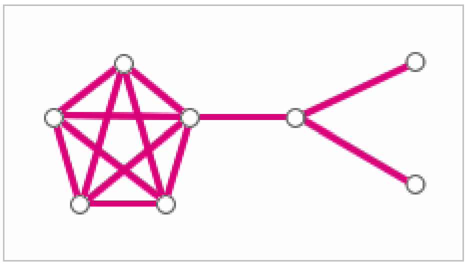
Počty koster

Určete počty všech možných koster následujících grafů:

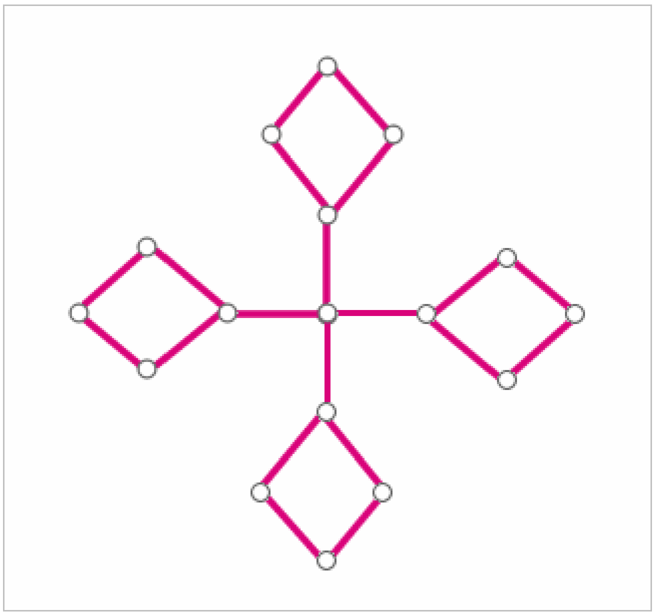
1.



2.



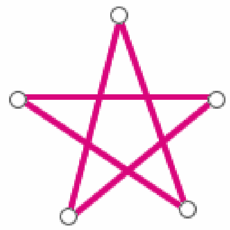
3.



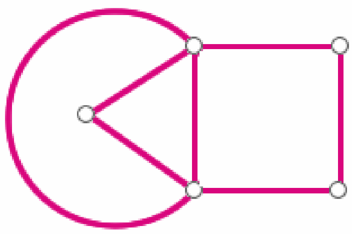
Jednotažky

Nakreslete graf jedním uzavřeným tahem nebo zdůvodněte, proč to není možné:

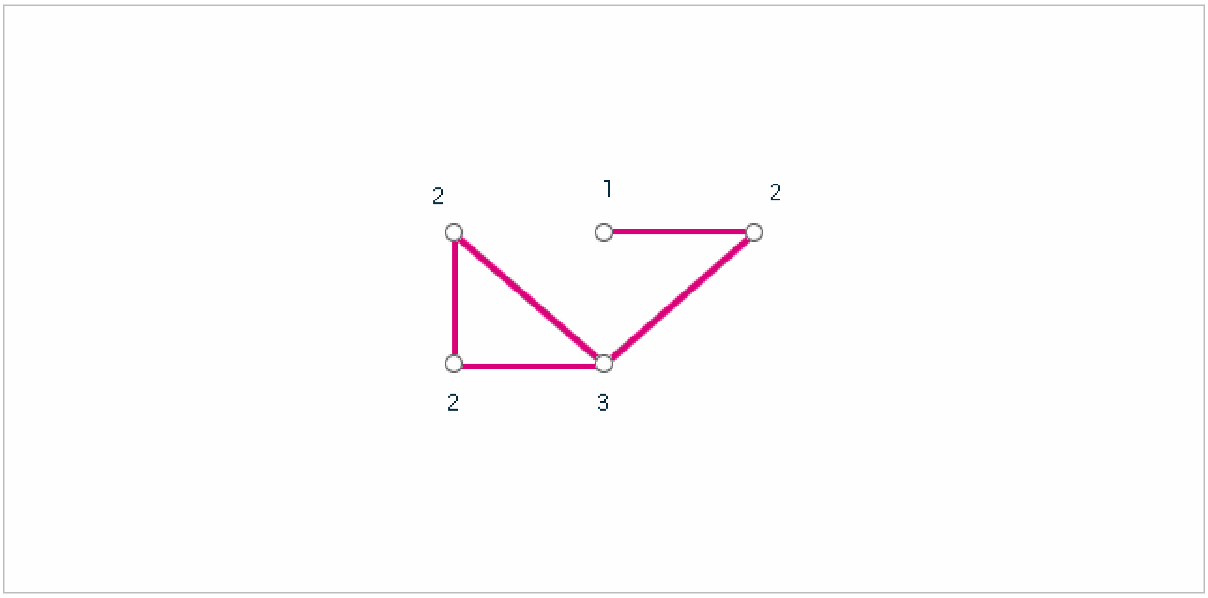
1.



2.

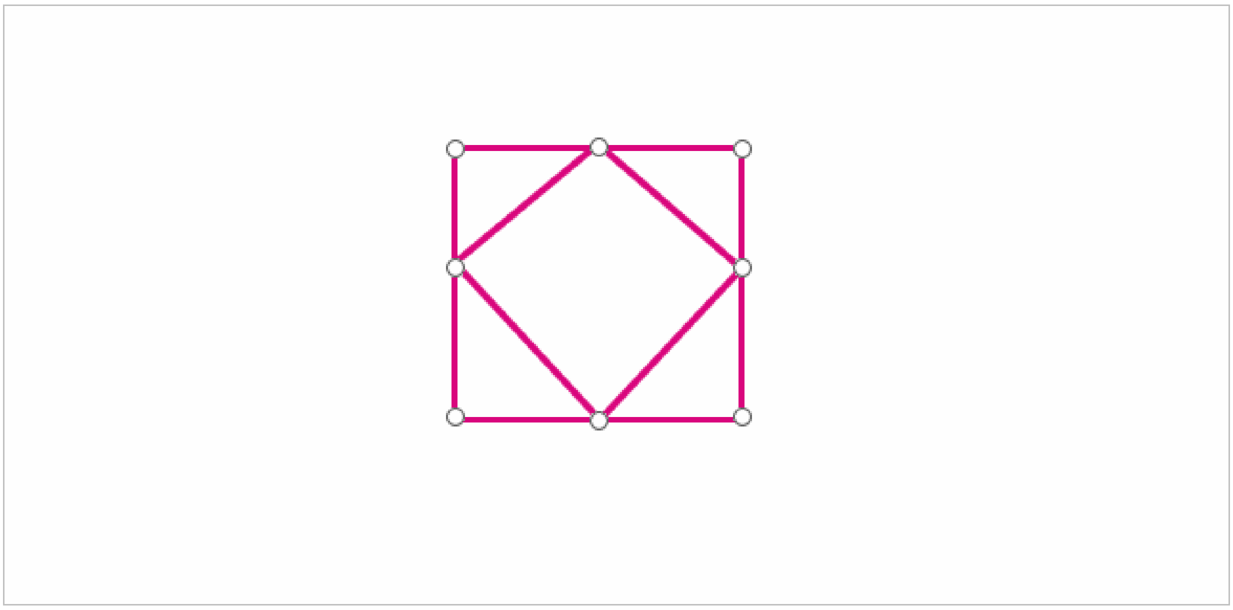


3.



4.

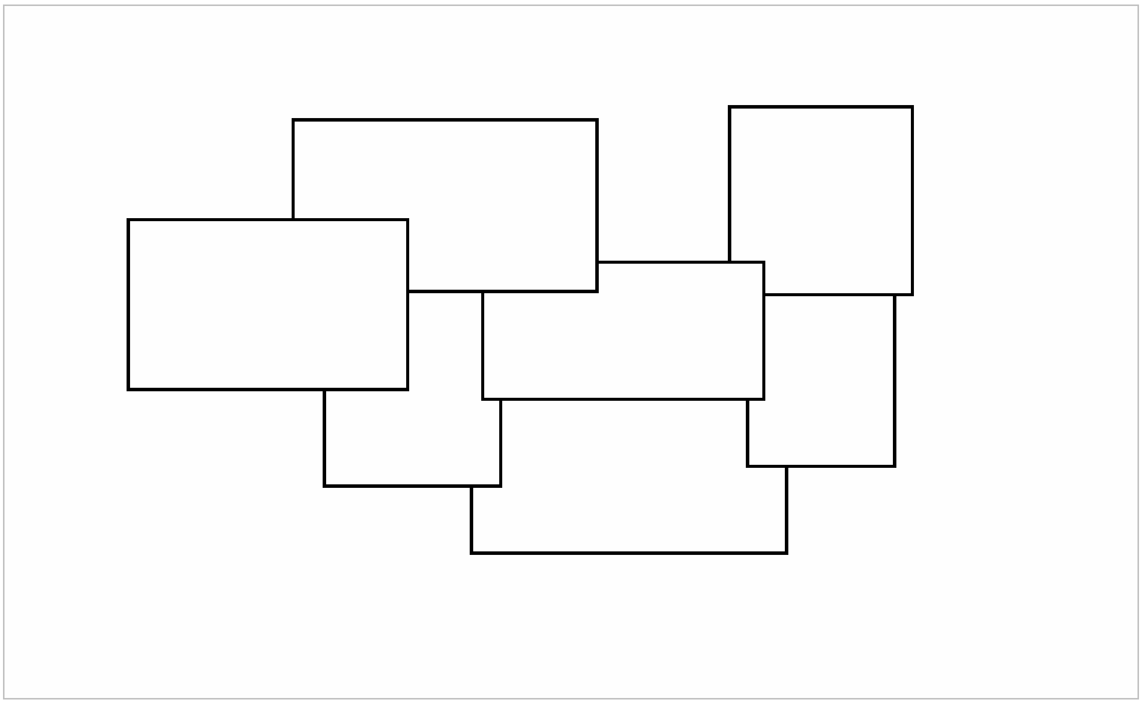
5.



Barvení mapy

1. Vybarvěte mapu [7] pomocí čtyř barev:

2. Vybarvěte mapu pomocí tří barev:



Úloha s kamarády

Úloha z matematické olympiády [3]

Ve skupině šesti lidí existuje právě 11 dvojic známých. Vztah "znát se" je vzájemný, tzn. jestliže osoba *A* zná osobu *B*, pak *B* zná *A*. Pokud se kdokoliv ze skupiny dozví nějakou zprávu, řekne ji všem svým známým. Dokažte, že se tímto způsobem zprávu dozví nakonec všichni.

Nápověda:

Kamarády si zakreslete jako vrcholy, vztah "znát se" použijte pro hrany a využijte vlastnosti úplných grafů.

Řešení úloh

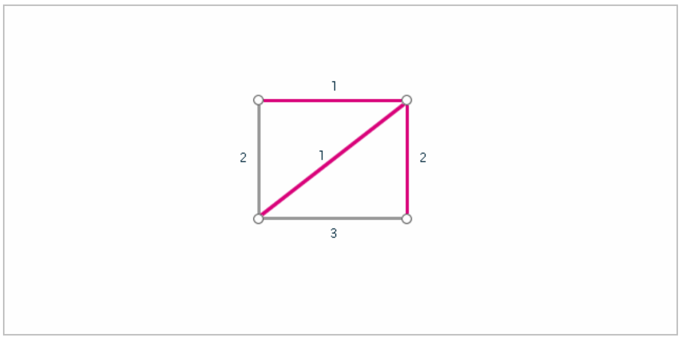
Základní pojmy

1. ANO
2. NE
3. NE
4. ANO
5. NE, je to strom.
6. ANO
7. ANO
8. NE
9. NE
10. Matice:

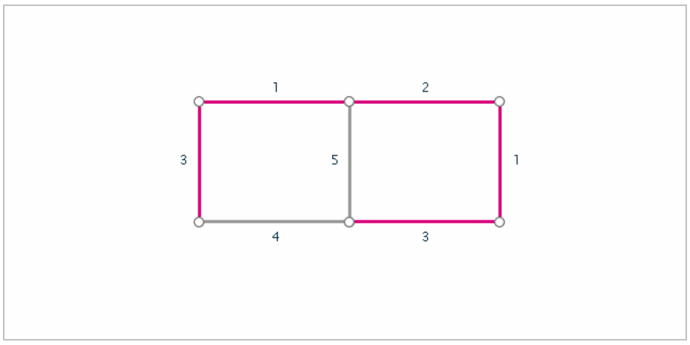
$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Minimální kostra

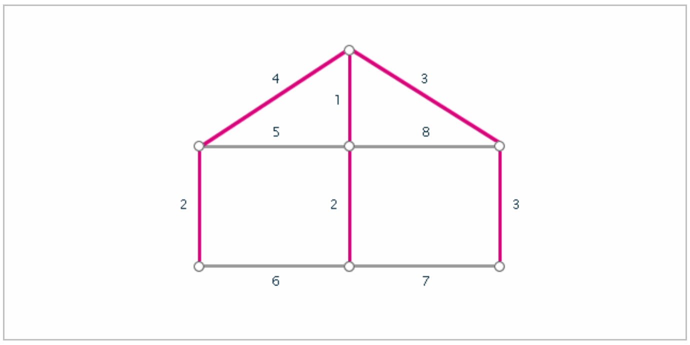
1.



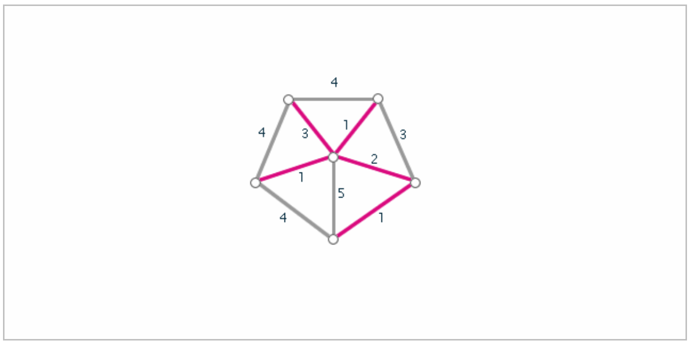
2.



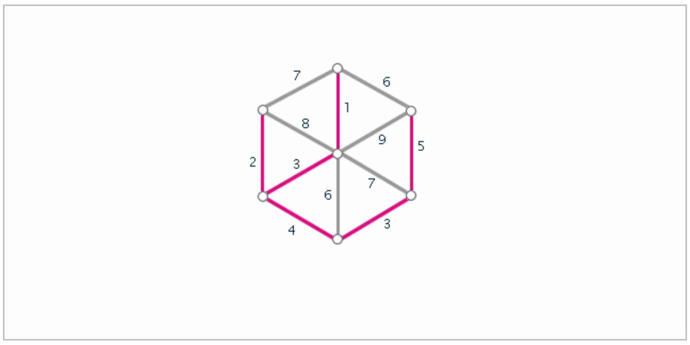
3.



4.

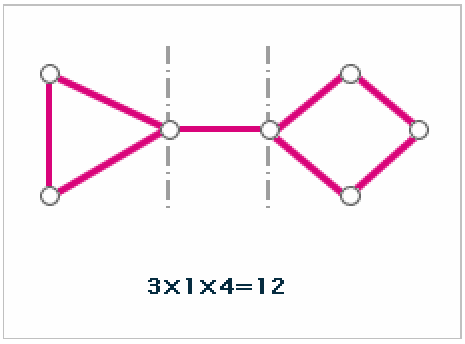


5.

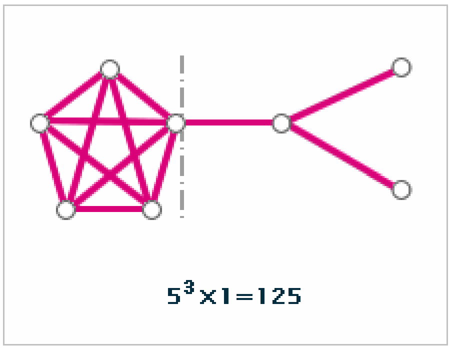


Počty koster

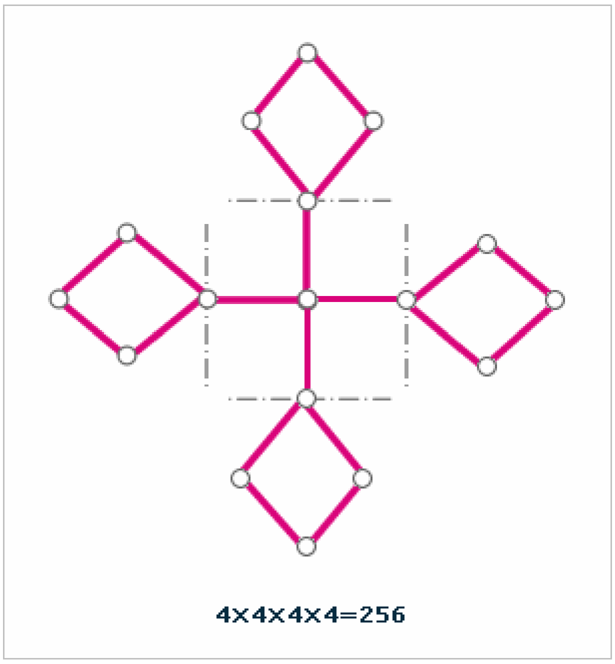
1.



2.

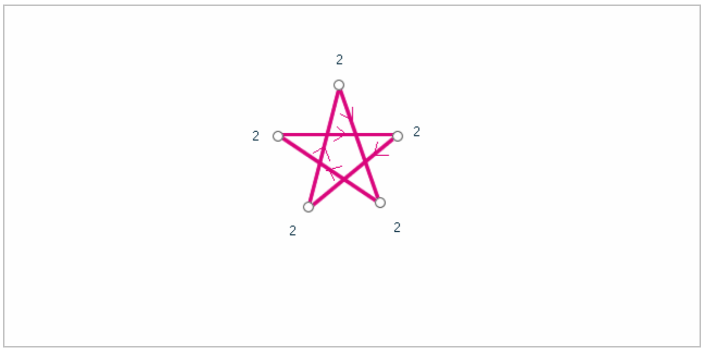


3.

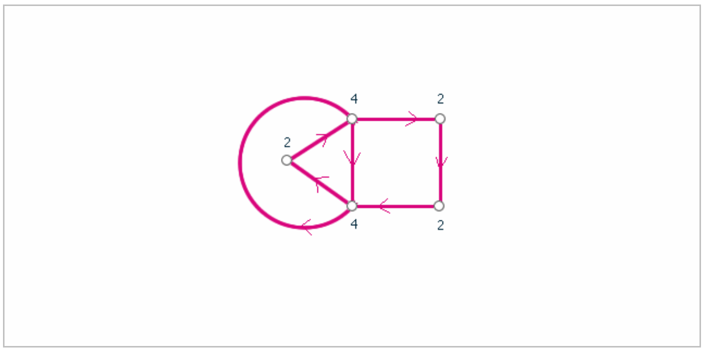


Jednotažky

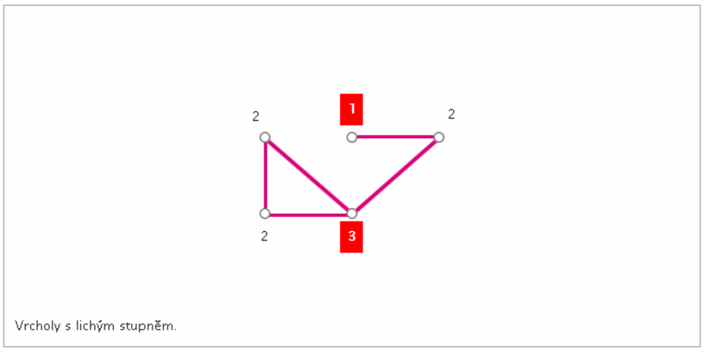
1.



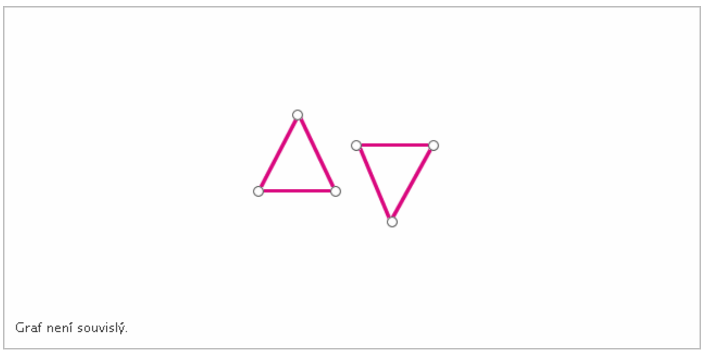
2.



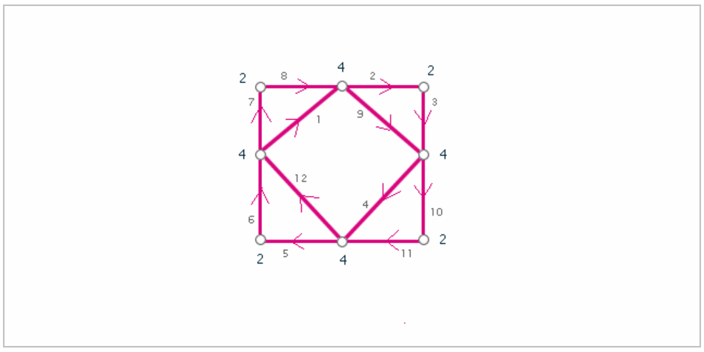
3.



4.

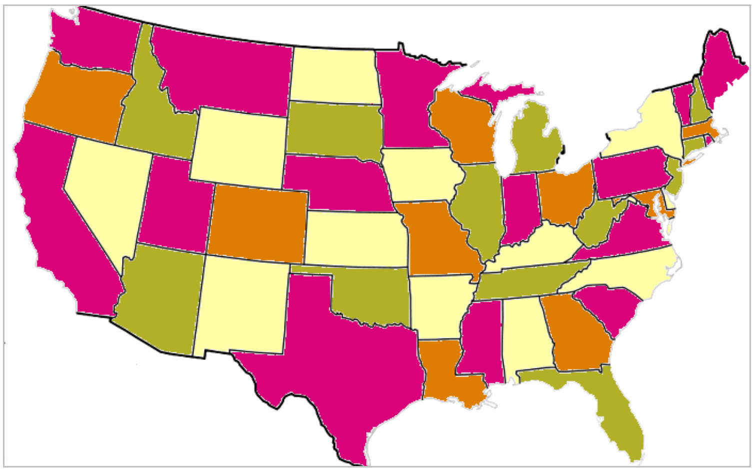


5.

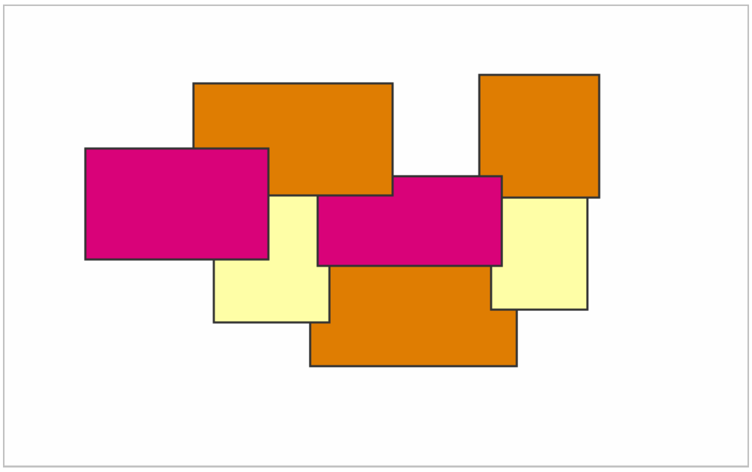


Barvení mapy

1. Jedno z možných řešení:



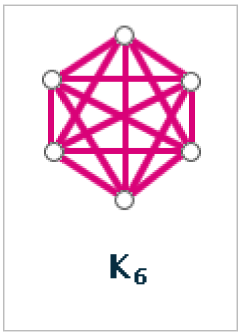
2. Jedno z možných řešení:



Kamarádi

Kamarády (známé) si zakreslíme jako graf o šesti vrcholech. Graf bude neorientovaný.

Představme si situaci, že se zná každý s každým - půjde o úplný graf o 15 vrcholech (stačí dosadit do vzorce pro výpočet počtu hran v úplném grafu).



K_6 - úplný graf na šesti vrcholech

Ze zadání víme, že hran v grafu bude 11, tzn. z grafu 4 hrany odebereme. Každý vrchol je ale spojen s ostatními 5 vrcholy, takže i pokud odebereme 4 hrany jednomu vrcholu, graf stále zůstane souvislý (→ každá zpráva se dostane ke všem kamrádům).

Závěr

Hlavním cílem práce bylo vytvořit webové stránky, na kterých si studenti středních škol (a případní jiní zájemci) osvojí základní pojmy teorie grafů a kde se seznámí s různými možnostmi praktického využití této části matematiky.

V současných učebnicích matematiky pro střední školy tato partie zcela chybí, proto je práce vhodná pro zájemce na seminářích, účastníky korespondenčních kurzů a řešitele Matematické olympiády.

Přínosem webových stránek na přiloženém CD je, že při tvorbě stránek byl kladen velký důraz na provázání informací, takže po prvotním seznámení s problematikou není bezpodmínečně nutné číst látku od začátku do konce, ale je možné mezi kapitolami přeskakovat podle potřeby díky hypertextovým odkazům, a tak například rychle najít vysvětlení pojmu použitého v pozdější definici apod.

Další výhodou je malá náročnost na vybavení počítače uživatele (stačí základní internetový prohlížeč, do kterého není nutné instalovat rozšiřující software, např. pluginy pro Javu či Flash).

Webové stránky jsou napsány v XHTML s použitím kaskádových stylů a skriptovacího jazyka PHP na straně serveru (pro vložení menu či ovládání animací).

Animace jsou řešené přepínáním obrázků (jazyk JavaScript) a všechny obrázky grafů (vyjma map, u kterých jsou uvedeny citace) byly nakresleny v programu Macromedia Fireworks.

Vytvořené stránky doplňují jiné středoškolské internetové učebnice umístěné na stránkách Katedry didaktiky matematiky MFF UK [11] v rámci aplikací informačních technologií ve výuce; vzhledem k tomu, že v případě teorie grafů jde o rozšiřující učivo nad rámec standardní výuky na SŠ, byl zvolen vlastní způsob ovládání animací, pro které jsou použity standardní obrázky a ne zmiňované náročnější Java applety, a vlastní grafický design odlišný od citovaných prací, díky čemuž došlo k lepší optimalizaci pro tisk.

Literatura a zdroje

Literatura

[1] Matoušek J., Nešetřil J. (2007): Kapitoly z diskrétní matematiky, Karolinum, Praha

[2] Wolfram Mathworld: Icosian Game, elektronický text, obrázek citován 12. 4. 2008, http://mathworld.wolfram.com/IcosianGame.html

[3] 57. ročník matematické olympiády (2007/2008), Úlohy klauzurní části školního kola kategorie C, úloha 3

[4] Zagorová P. (2001): Historie a vývoj teorie grafů, seminární práce, Přírodovědecká fakulta, Masarykova Univerzita, Brno – dostupné na www: http://www.math.muni.cz/~xzagorov/hist_mat/Hist_mat.doc, citováno 19. 4. 2008

[5] Šišma P. (1997): Teorie grafů 1736-1963, Prometheus, Praha

[6] Opava Z. (1989): Matematika kolem nás, Albatros, Praha

[7] Anderson I. (2001): A First Course in Discrete Mathematics, Springer, London

Další zdroje

[8] Wikipedia (EN verze), slepá mapka Evropy, obrázek citován 30. 3. 2008, http://en.wikipedia.org/wiki/Image:BlankMap-Europe.png

[9] Wikipedia (EN verze), slepá mapka států USA, obrázek citován 30. 3. 2008, http://en.wikipedia.org/wiki/Wikipedia:Blank_maps#North_America_2

[10] Mapy.cz (vzdálenosti mezi městy), elektronická služba, údaje citovány 1. 2. 2008, http://mapy.cz

[11] Webové stránky Katedry didaktiky matematiky MFF UK – studentské práce, http://www.karlin.mff.cuni.cz/katedry/kdm/diplomky/index.php

Odkazy

[12] Vaníček, J. a kolektiv (2008): Teoretické základy informatiky, Kernberg Publishing, s.r.o., Praha

[13] Robertson N., Sanders D.P., Seymour P.D., Thomas R. (1997): The Four Color Theorem, Journal of Combinatorial Theory Ser. B 70, 2-44

[14] Gonthier G. (2005): A computer-checked proof of the Four Color Theorem, Technical Report, Microsoft Research