# Základy informatiky

Teorie grafů

Zpracoval: Pavel Děrgel
Úprava: Daniela Szturcová

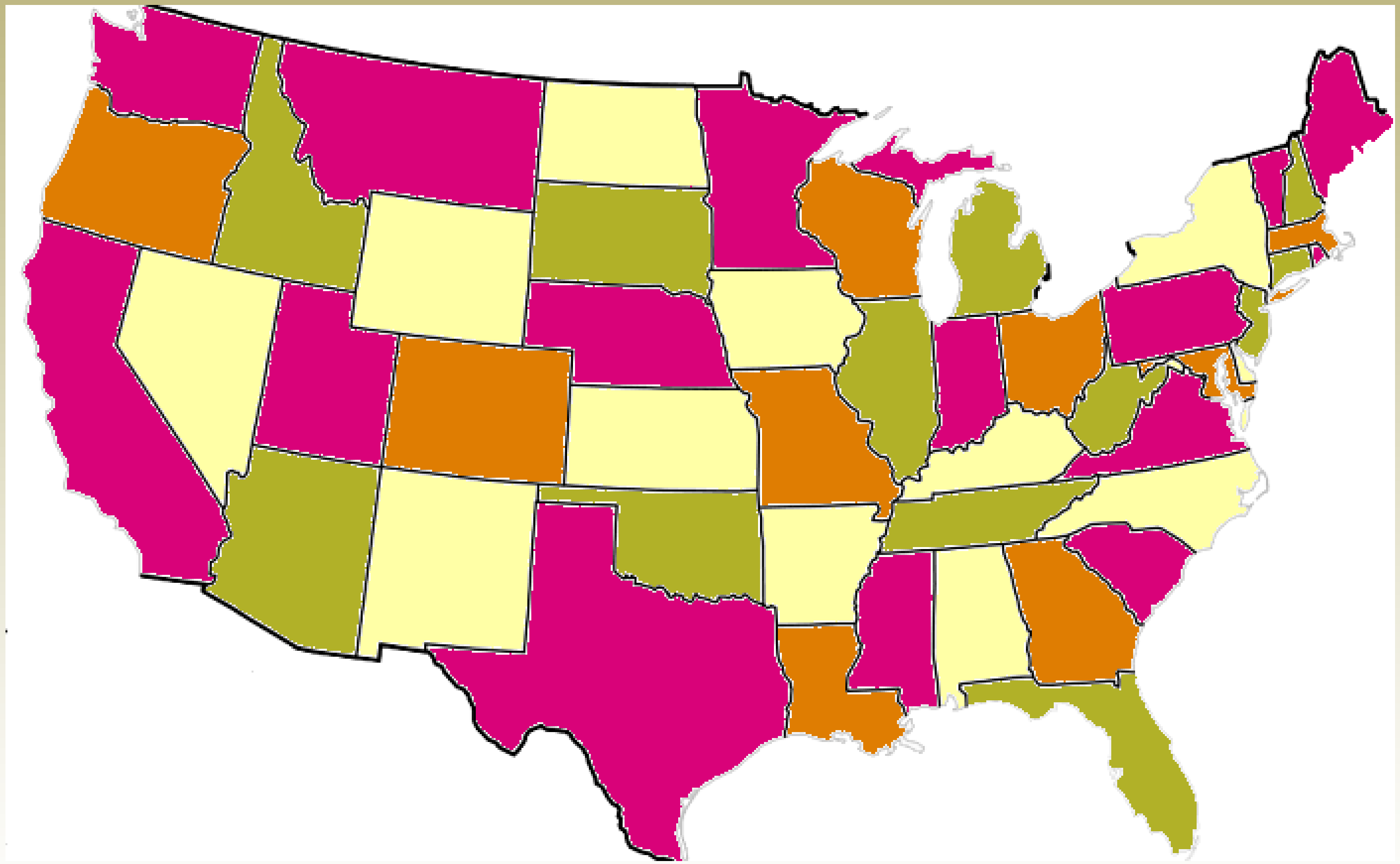
# Obsah přednášky

- Barvení mapy
- Teorie grafů
  - Definice
  - Uzly a hrany
  - Typy grafů
  - Cesty, cykly, souvislost grafů

# Barvení mapy

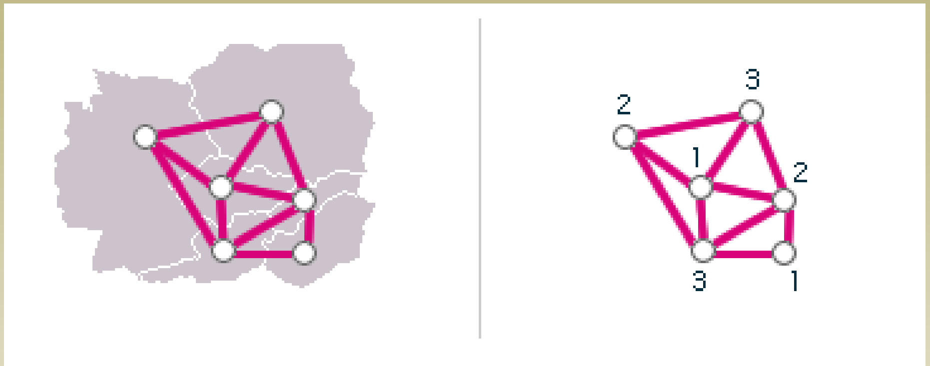
http://teorie-grafu.elfineer.cz/

# Barvení mapy



http://teorie-grafu.elfineer.cz/

# Barvení mapy – spojitost s grafy




mobilní sítě – počet frekvencí

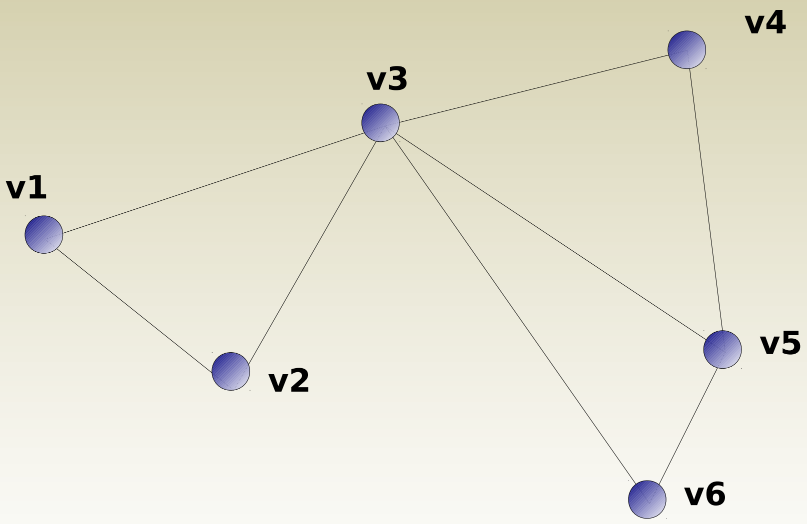
# Teorie grafů

- Teorie grafů zkoumá vlastnosti struktur, zvaných grafy. Grafy nám umožňují jednoduše a přehledně popisovat reálné systémy, které jsou reprezentovány pomocí sítí (počítačové sítě, silniční sítě, atd.) nebo mohou být na grafovou reprezentaci převedeny.

- Grafové algoritmy – umožňují zpracovávat grafy, hledat nejkratší nebo nejrychlejší cesty, počítat propustnost sítě, atd.

# Definice grafu

- Graf je definován jako dvojice G = (V, E)
- V je množina vrcholů
- E je množina hran

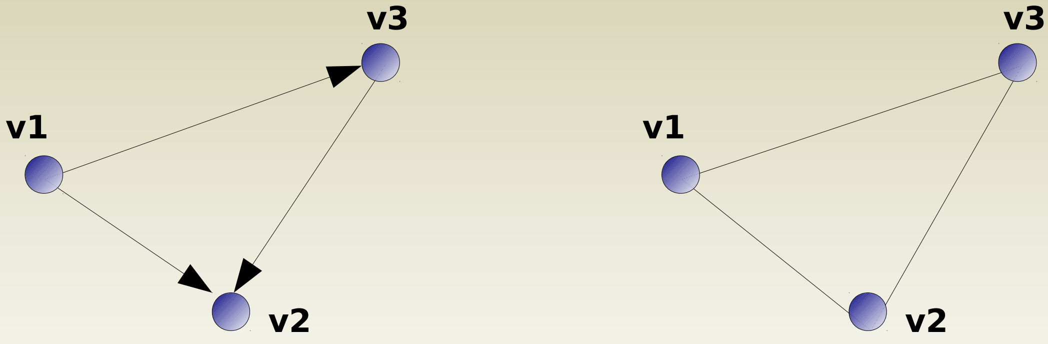


# Konvence

- $V(G)$ – množina vrcholů v grafu $G$
- $E(G)$ – množina hran v grafu $G$
- $|G|$ - Počet vrcholů v grafu $G$
- $||G||$ - Počet hran v grafu $G$
- $G = (\emptyset, \emptyset)$ - Prázdný graf
- $v \in e$ – vrchol $v$ náleží hraně $e$ (v je incidentní s hranou e)
- $d_v$ – stupeň vrcholu (počet hran, se kterými je vrchol incidentní)

# Typy grafů

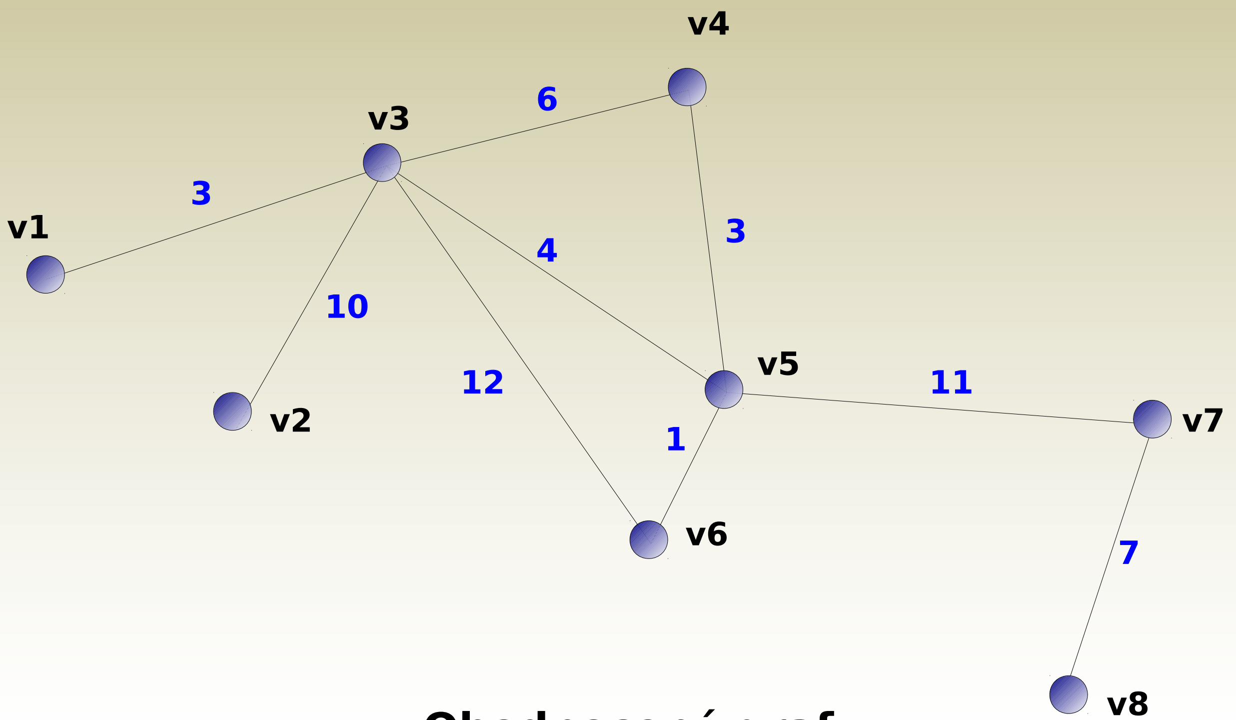
- Orientovaný – jeho hrany jsou uspořádané dvojice
- Neorientovaný – jeho hrany jsou dvouprvkové množiny (nezáleží na orientaci).




**Orientovaný graf** **Neorientovaný graf**

# Ohodnocený graf

- Ohodnocený graf – jeho hrany jsou opatřeny číselnými nebo jinými hodnotami (reprezentace vzdáleností na silnici, možnost různých časových průjezdů v opačných směrech - stoupání).



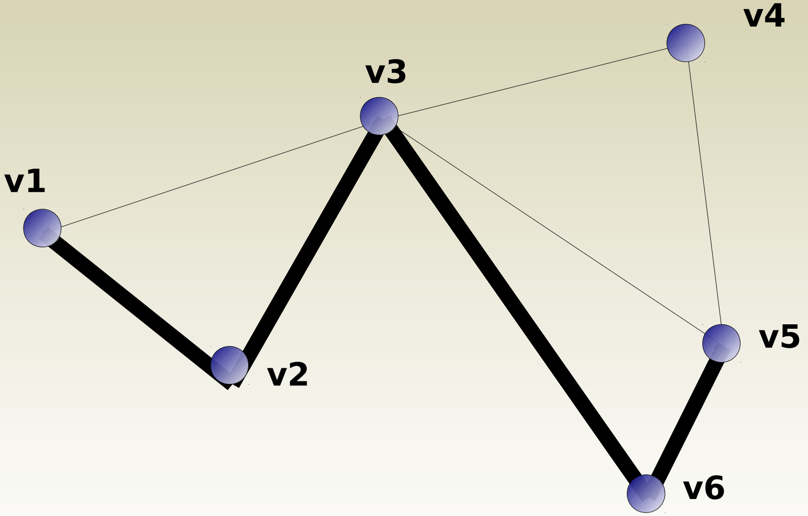

**Ohodnocený graf**

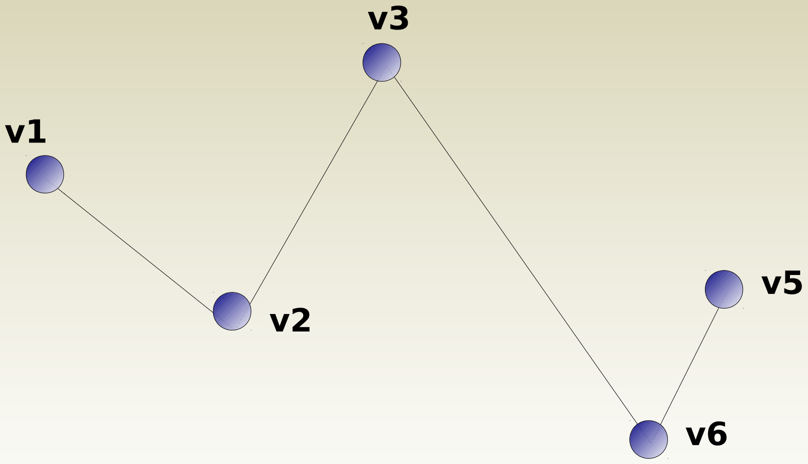
# Cesta

Cesta P je neprázdný graf $P = (V, E)$, kde

$V = \{x_1, x_2, ..., x_k\}$ $E = \{x_0x_1, x_1x_2, ..., x_{k-1}x_k\}$

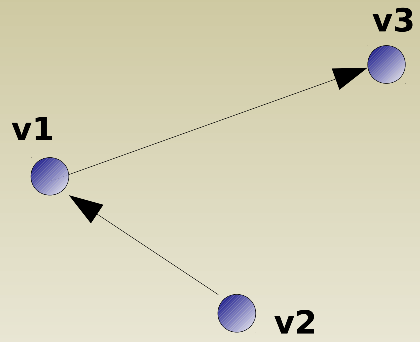
Cesta je posloupnost vrcholů a hran.



# Dosažitelnost vrcholů

Vrchol B je dosažitelný z vrcholu A, pokud existuje alespoň jedna cesta z vrcholu A do vrcholu B.



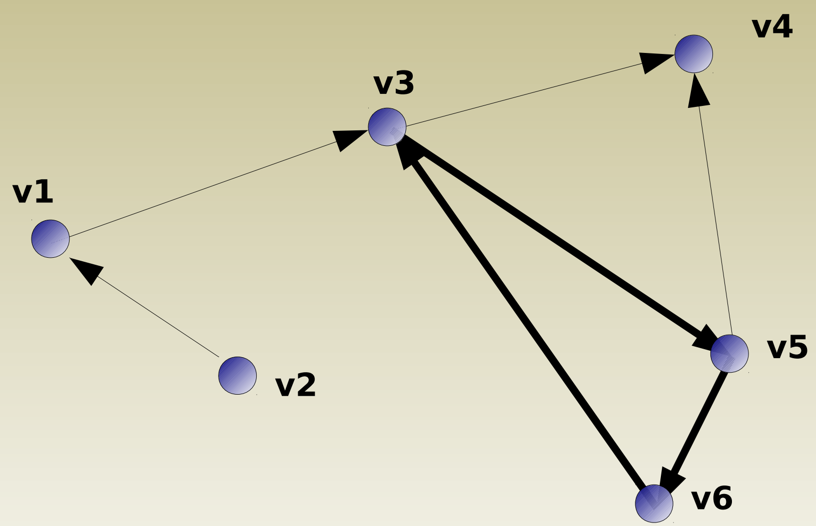

Příklad:

vrchol $v_3$ je dosažitelný z $v_2$

(naopak to neplatí)

# Cyklus

Cesta, jejíž první vrchol je totožný s posledním se nazývá cyklus.

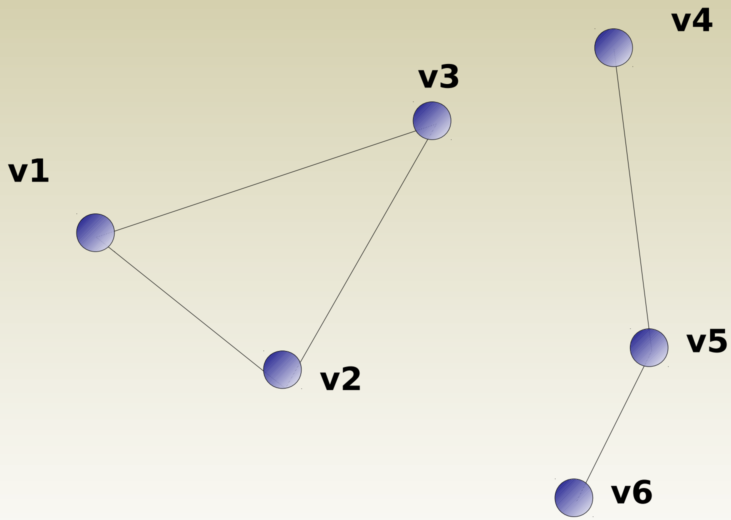



$$C = \{v_3, v_5, v_6, v_3\}$$

- Graf, který neobsahuje žádný cyklus se nazývá acyklický.
- Cyklus se může nacházet i v neorientovaném grafu

# Souvislost grafu

- Každý neprázdný graf G se nazývá souvislý, pokud existuje cesta mezi kteroukoliv dvojicí vrcholů.
- U orientovaných grafů se při vyhodnocování souvislosti orientace ignoruje.

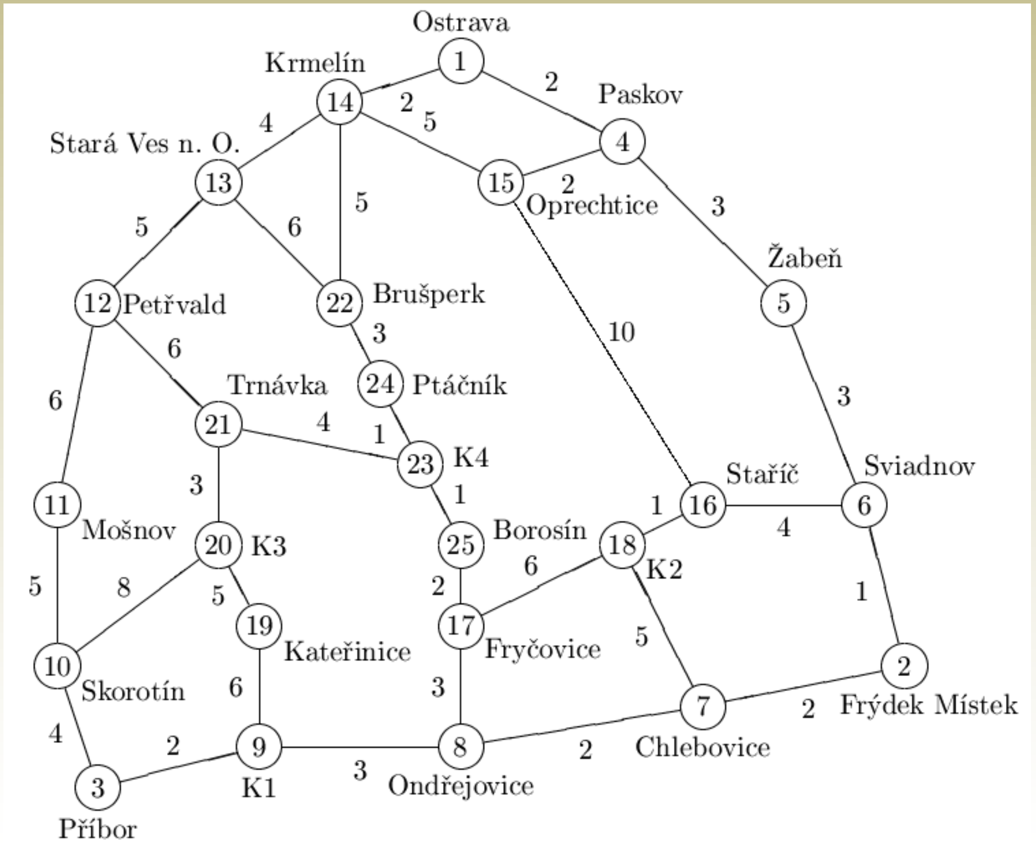



Nesouvislý graf

# Minimální kostra

Výchozí graf se upraví tak, aby splňoval následující požadavky:

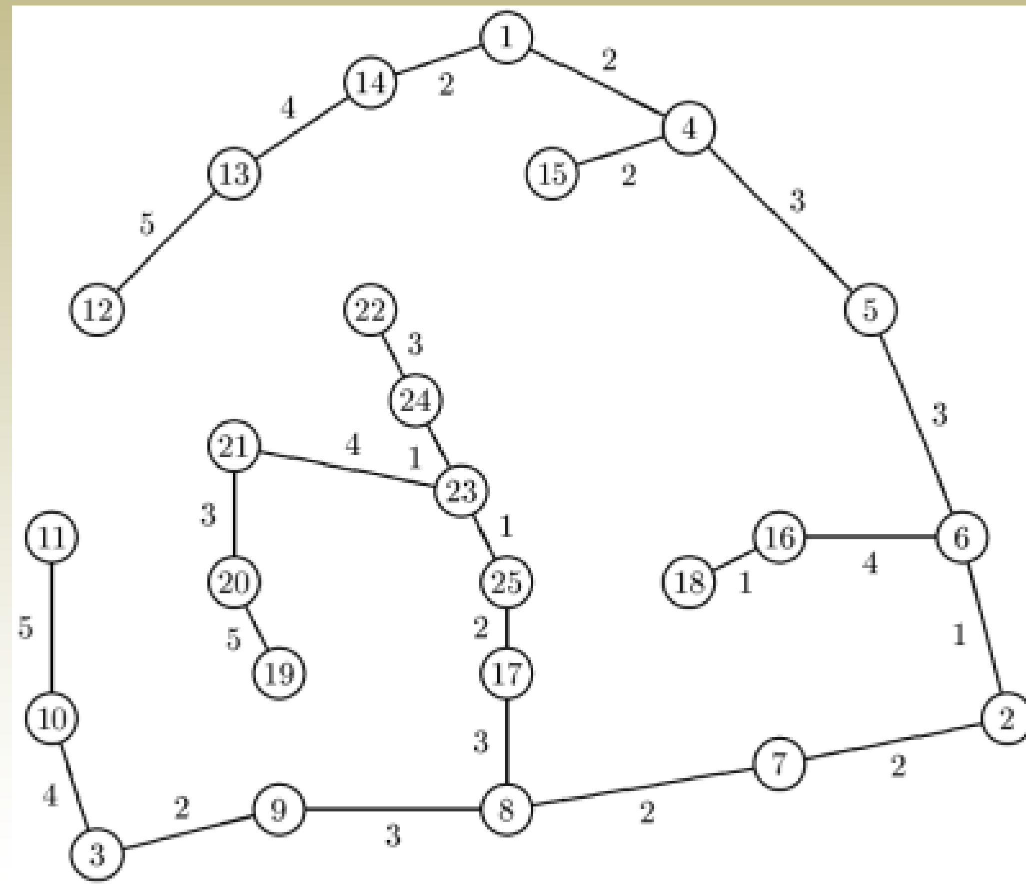
- stejný počet vrcholů,
- souvislý,
- acyklický,
- s minimálním ohodnocením.




http://homel.vsb.cz/~kov16/talks/algoritmy/

# Minimální kostra

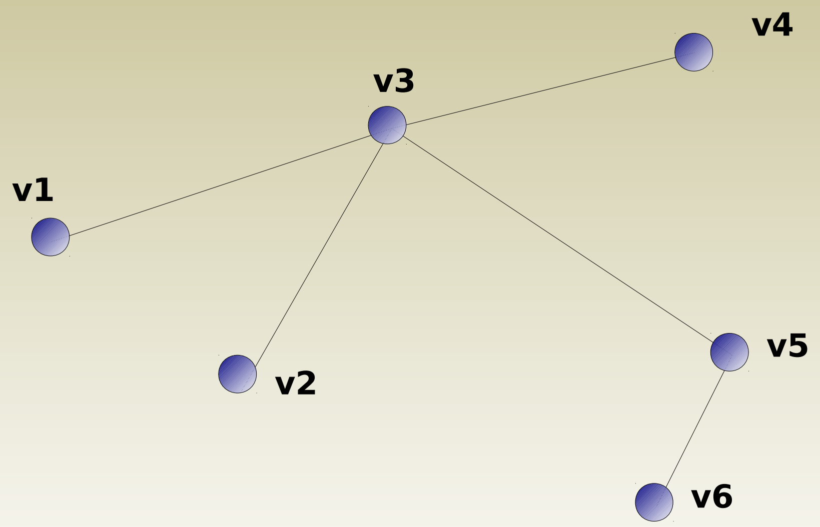
Kostra u souvislého grafu vždy existuje a má n-1 hran.




http://homel.vsb.cz/~kov16/talks/algoritmy/

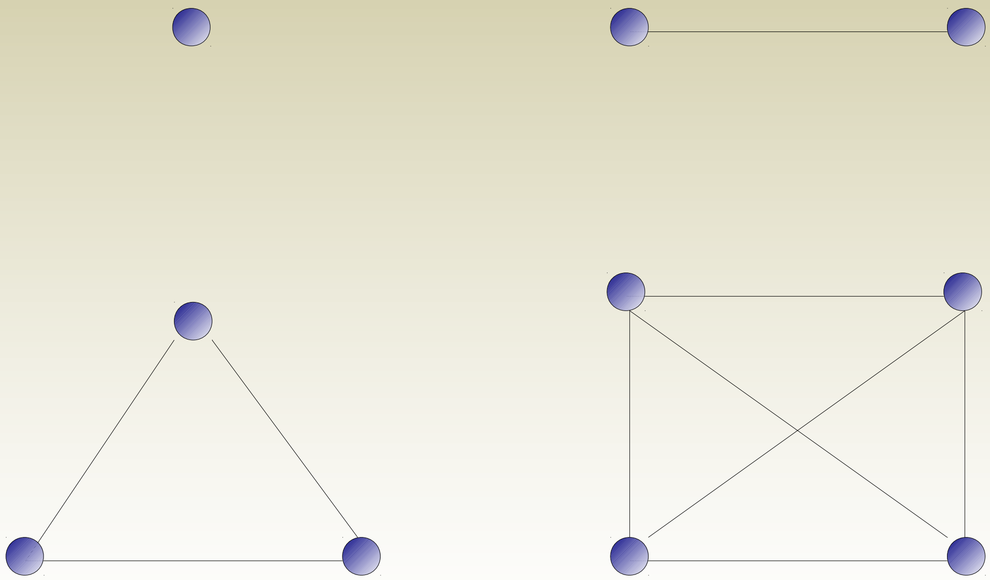
# Stromy

- Spojitý a acyklický graf se nazývá strom.



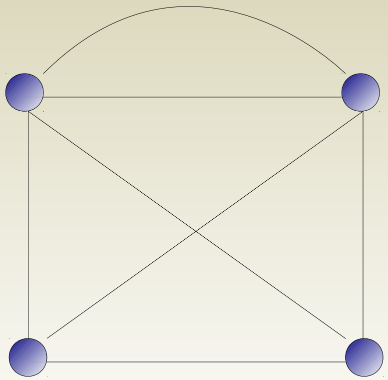
# Úplný graf

- Úplný graf je takový graf, ve kterém je každý vrchol propojen hranou se všemi ostatními vrcholy v grafu



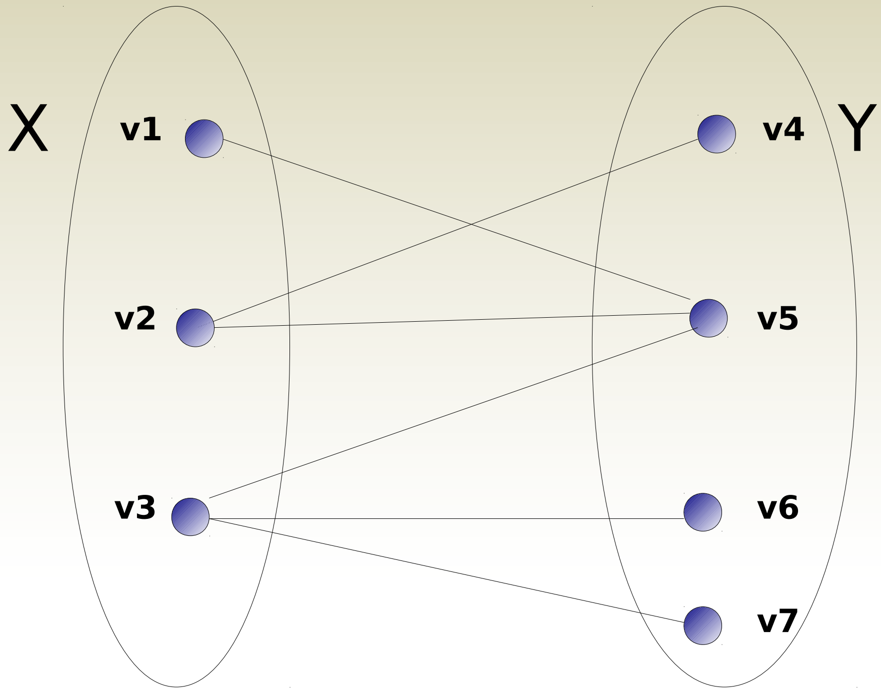
# Multigraf

- Multigraf je takový graf, který obsahuje alespoň jednu násobnou hranu (dva vrcholy spojené dvěma nebo více hranami)



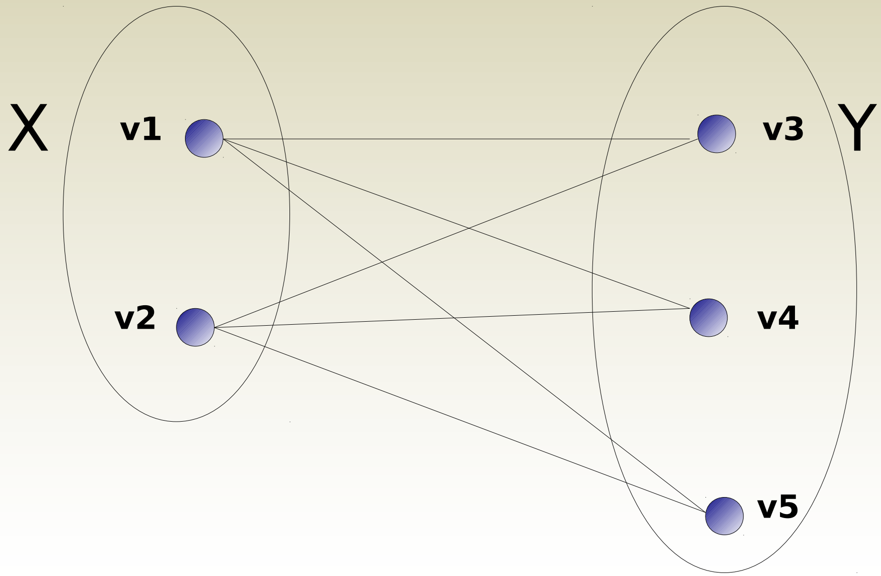
# Bipartitní graf

- Bipartitní graf je takový graf, ve kterém se množina vrcholů V skládá ze dvou disjunktních podmnožin X, Y zvaných parity. Množina hran E takového grafu obsahuje pouze hrany, jejichž jeden koncový vrchol leží v množině X a druhý koncový vrchol v množině Y. Žádné dva vrcholy z parity X (resp. Y) nejsou spojeny hranou.



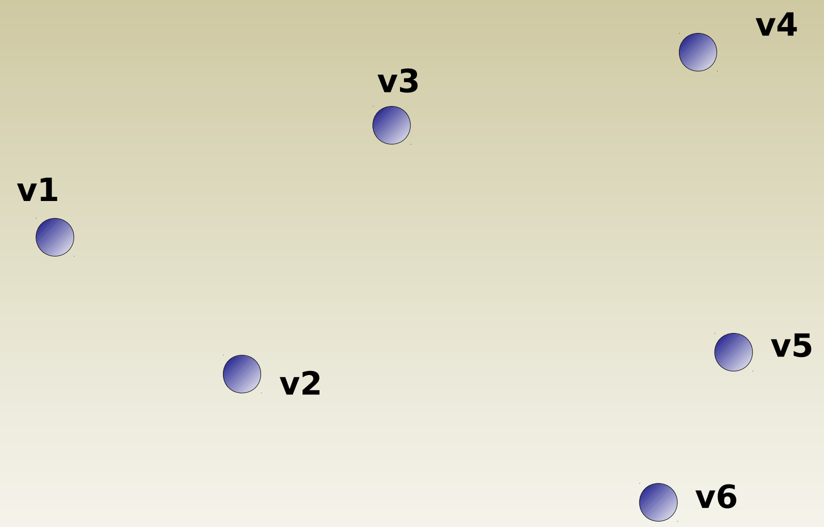
# Úplný bipartitní graf

- Úplný bipartitní graf je bipartitní graf, ve kterém je každá dvojice vrcholů $x \in X$ a $y \in Y$ spojena právě jednou hranou.



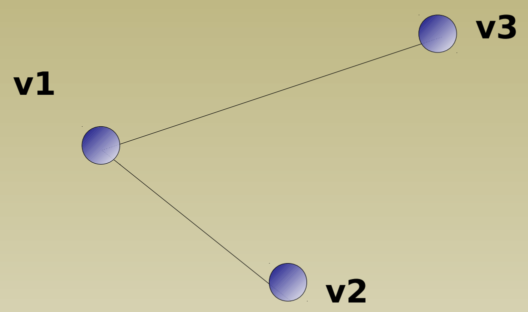
# Disktrétní graf

- Diskrétní graf je takový graf, který nemá žádnou hranu (||G|| = 0)



# Reprezentace grafu

- Obrázek (nakreslení grafu)




- Matice sousednosti (popř. matice incidence)

$$A = \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{matrix}$$

- Seznam vrcholů a jejich sousedů

$v_1$ -> $v_3$

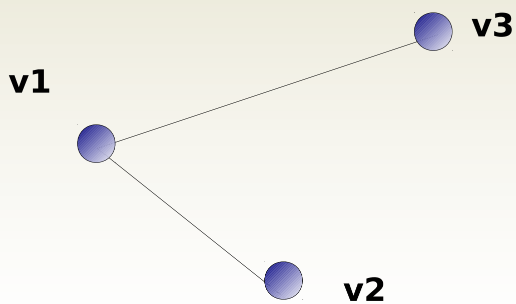
$v_1$ -> $v_2$

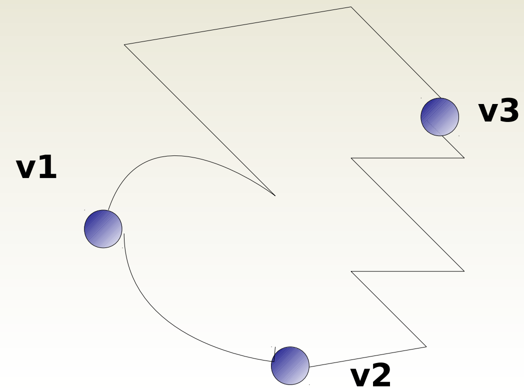
- Seznam hran

$G = \{(v_1,v_3), (v_1,v_2)\}$

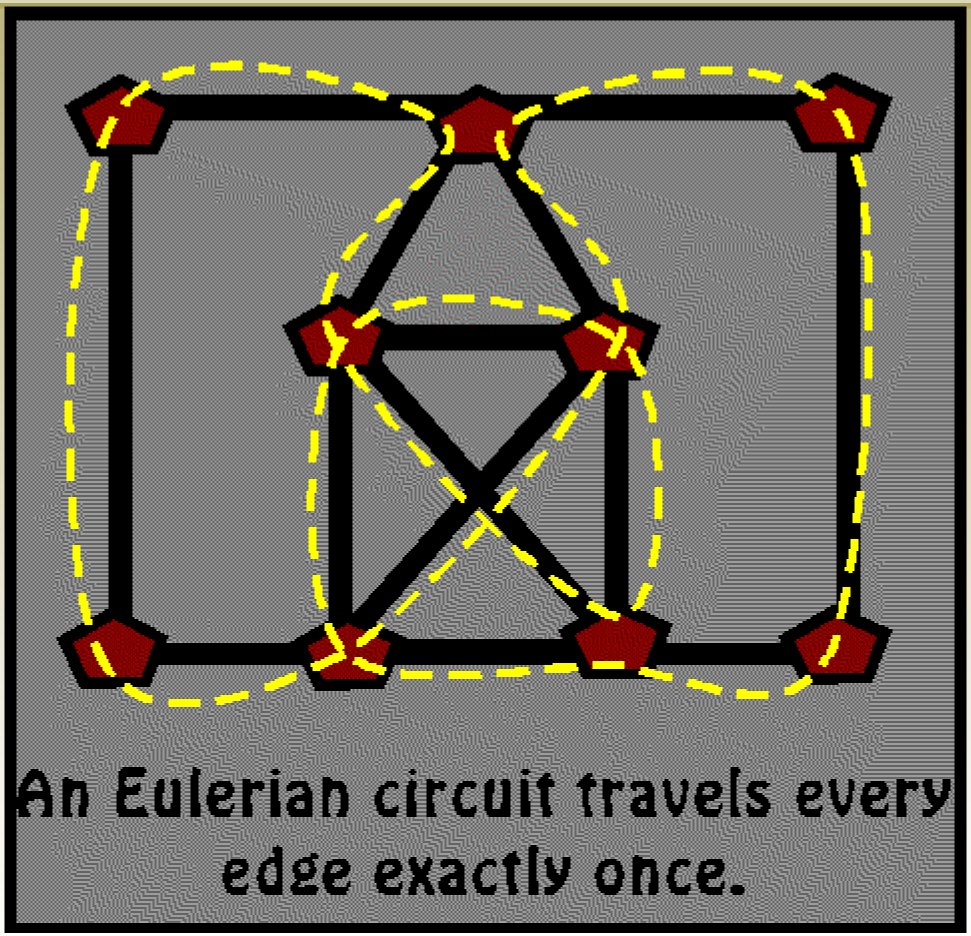
# Nakreslení grafu

- Vrcholy grafu jsou nakresleny jako body (kroužky), hrany jako čáry.
- Orientovanou hranu znázorňujeme šipkou od počátečního vrcholu ke koncovému.
- Existuje mnoho různých nakreslení stejného grafu.
- Výhodou je přehlednost.
- Nakreslení je nevýhodné v případě rozsáhlého grafu, nebo pokud vyžadujeme automatické zpracování grafu.



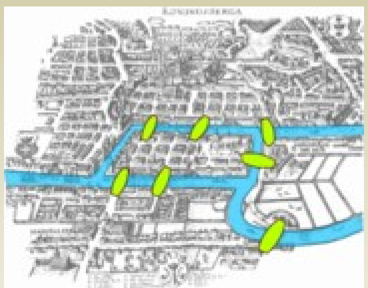
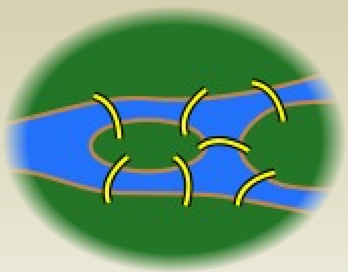
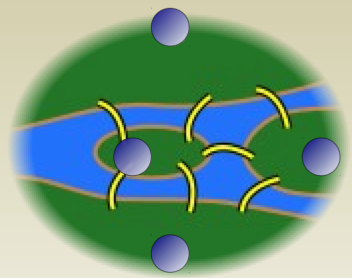
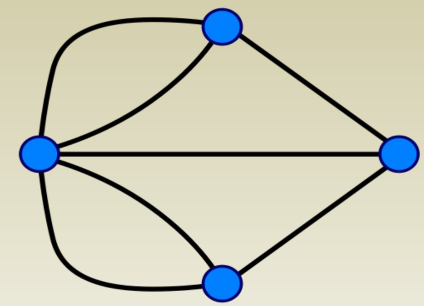
# Eulerovský tah




http://www.c3.lanl.gov/

# Eulerovský tah

Sedm mostů města Královce

http://cs.wikipedia.org/wiki/Sedm_most%C5%AF_m%C4%9Bsta_Kr%C3%A1lovce

# Matice sousednosti

Postup pro vytvoření:

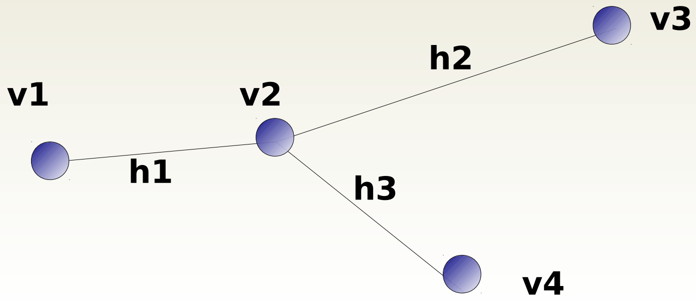
- Zvolíme libovolné (ale pevné) pořadí vrcholů ($v_1$ ... $v_n$), můžeme grafu přiřadit matici incidence takto:

  **$A_{i,j}$ =** **1**, pokud **$v_i$** je incidentní s **$v_j$**

  **0** v opačném případě.

  V případě ohodnocených hran může být **$A_{i,j}$ = n**, kde n je hodnota hrany.

- Matice sousednosti neorientovaného grafu bude vždy symetrická (u orientovaného grafu nemusí být).




| | v1 | v2 | v3 | v4 |
|---|---|---|---|---|
| v1 | **0** | **1** | **0** | **0** |
| v2 | **1** | **0** | **1** | **1** |
| v3 | **0** | **1** | **0** | **0** |
| v4 | **0** | **1** | **0** | **0** |

# Matice incidence

- Tato reprezentace je vhodná pro automatické zpracování počítačem.
- Nevýhodou je nepřehlednost pro člověka.

Postup pro vytvoření:

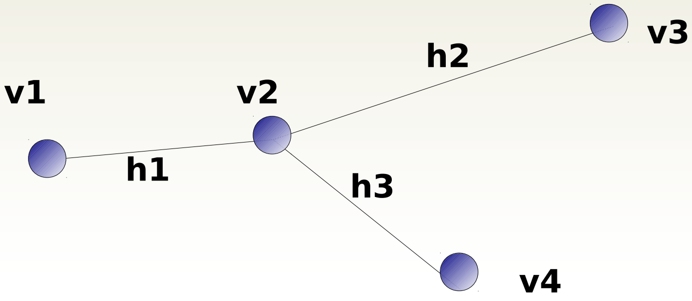
- Zvolíme libovolné (ale pevné) pořadí vrcholů ($v_1$ ... $v_n$) a hran ($h_1$ ... $h_m$), můžeme grafu přiřadit matici incidence takto:

  $\mathbf{A_{i,j} = 1}$, pokud $\mathbf{v_i}$ je incidentní s $\mathbf{h}_j$

  = **0** v opačném případě.

  V případě ohodnocených hran může být $\mathbf{A_{i,j} = n}$, kde n je hodnota hrany.

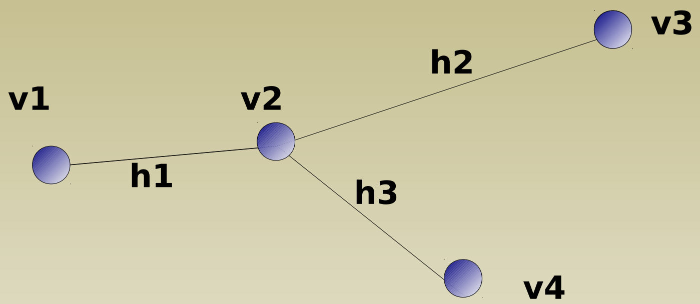
  V případě orientovaného grafu je $\mathbf{A_{i,j} = n}$ pokud je $\mathbf{v_i}$ počáteční vrchol hrany a $\mathbf{A_{i,j} = -n}$, pokud je $\mathbf{v_i}$ koncový vrchol hrany.



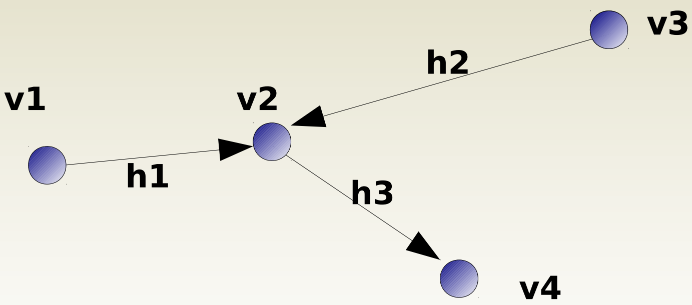

| | h1 | h2 | h3 |
|---|---|---|---|
| v1 | **1** | **0** | **0** |
| v2 | **1** | **1** | **1** |
| v3 | **0** | **1** | **0** |
| v4 | **0** | **0** | **1** |

# Seznam sousedů

- Graf je dán seznamem vrcholů a jejich sousedů



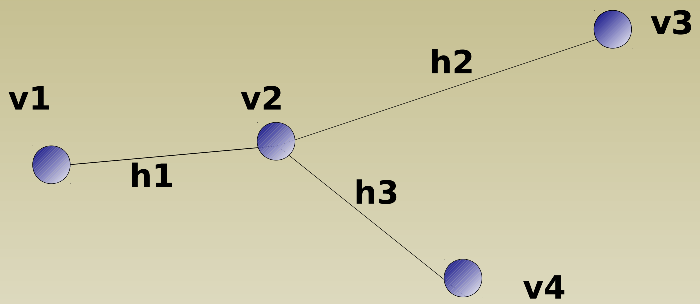

**v1 → v2**
**v2 → v1, v3, v4**
**v3 → v2**
**v4 → v2**



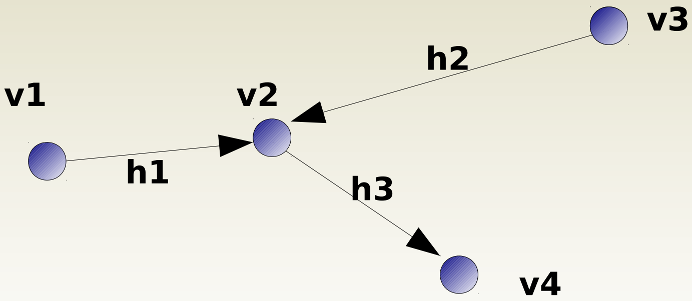

**v1 → v2**
**v2 → v4**
**v3 → v2**
**v4 →**

# Seznam hran

- Graf je dán seznamem hran




**G = {(v1, v2), (v2,v3), (v2,v4)}**




**G = {(v1, v2), (v3,v2), (v2,v4)}**

# Prohledávání grafu

- Prohledávání grafu je systematický postup, kterým můžeme řešit například hledání nejkratší, nejdelší, nejlevnější cesty z jednoho vrcholu do druhého, popřípadě zjišťovat dostupnost vrcholů v grafu apod.

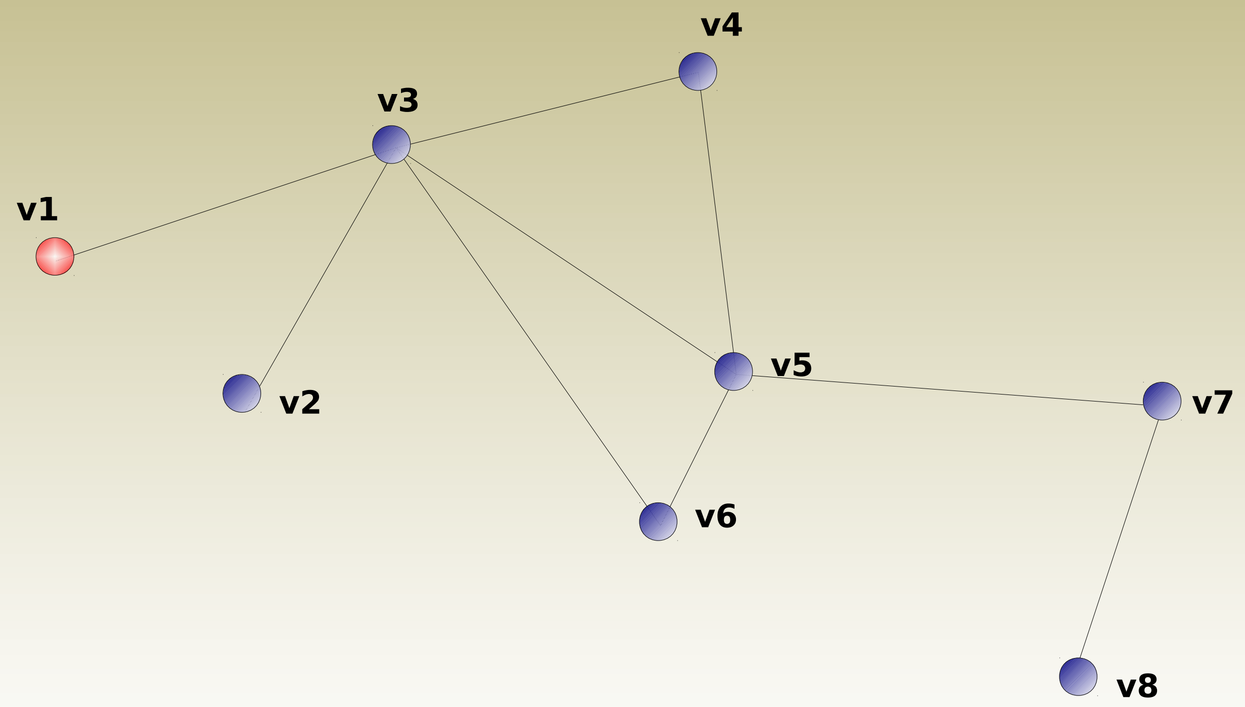
# Prohledávání grafu

- Prohledávání grafu do hloubky
- Prohledávání grafu do šířky

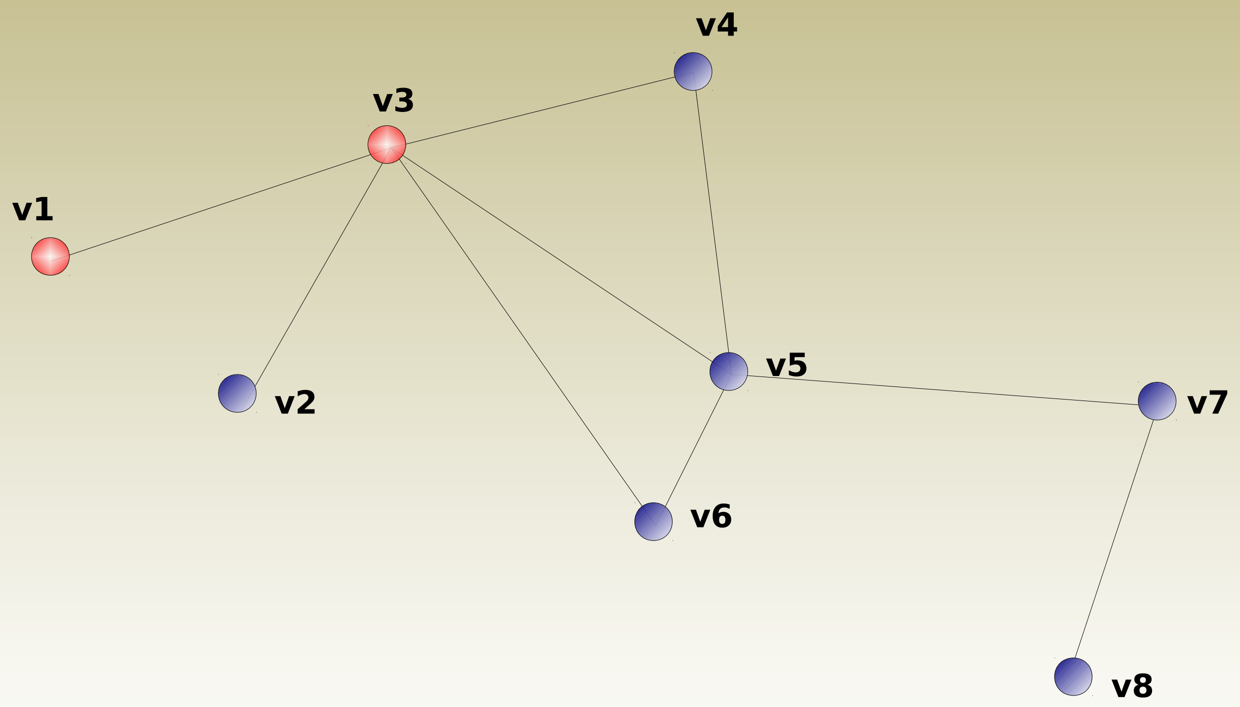
# Prohledávání grafu do hloubky

- Tento algoritmus si lze představit jako průchod bludištěm, kdy procházím z místnosti do místnosti po chodbách a v dosažené místnosti si vybíráme chodbu, po které jsme ještě nešli. Pokud taková chodba neexistuje, musíme se z této místnosti vrátit do předešlé místnosti.
- Algoritmus končí, pokud jsme prošli všechny uzly grafu nebo pokud jsme našli cílový vrchol.

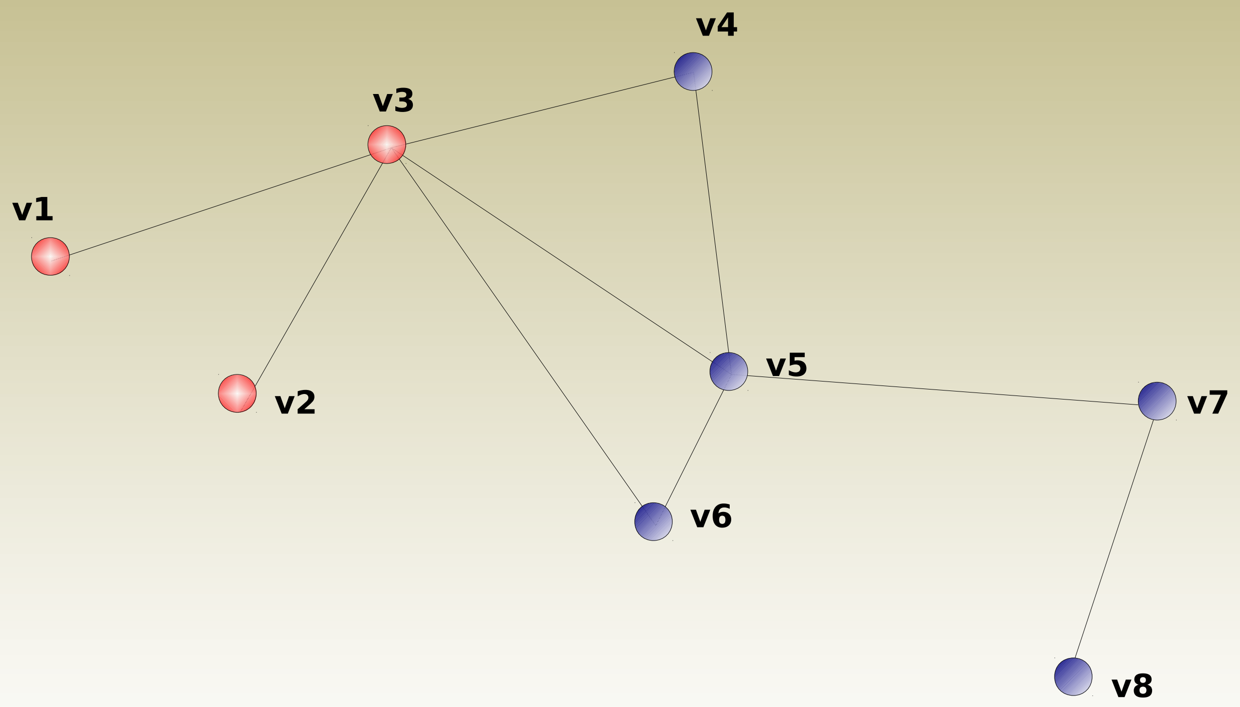
# Prohledávání do hloubky ($v_1$ -> $v_8$)



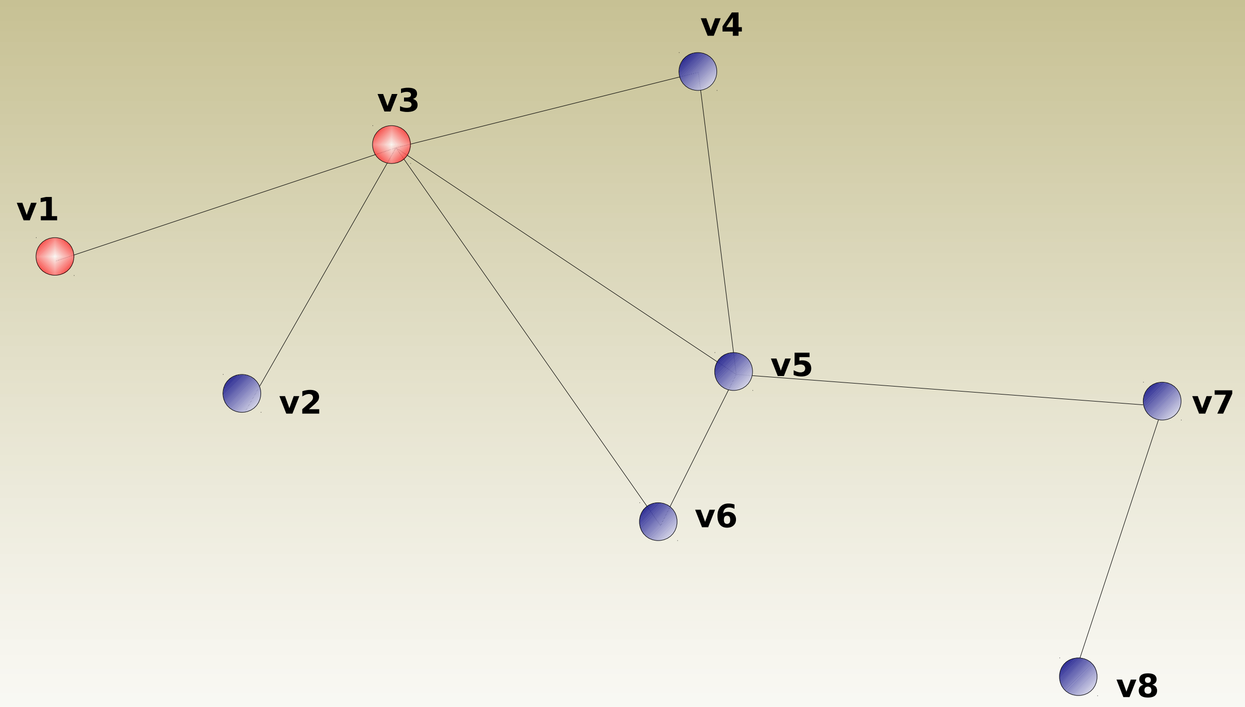
# Prohledávání do hloubky ($v_1$ -> $v_8$)



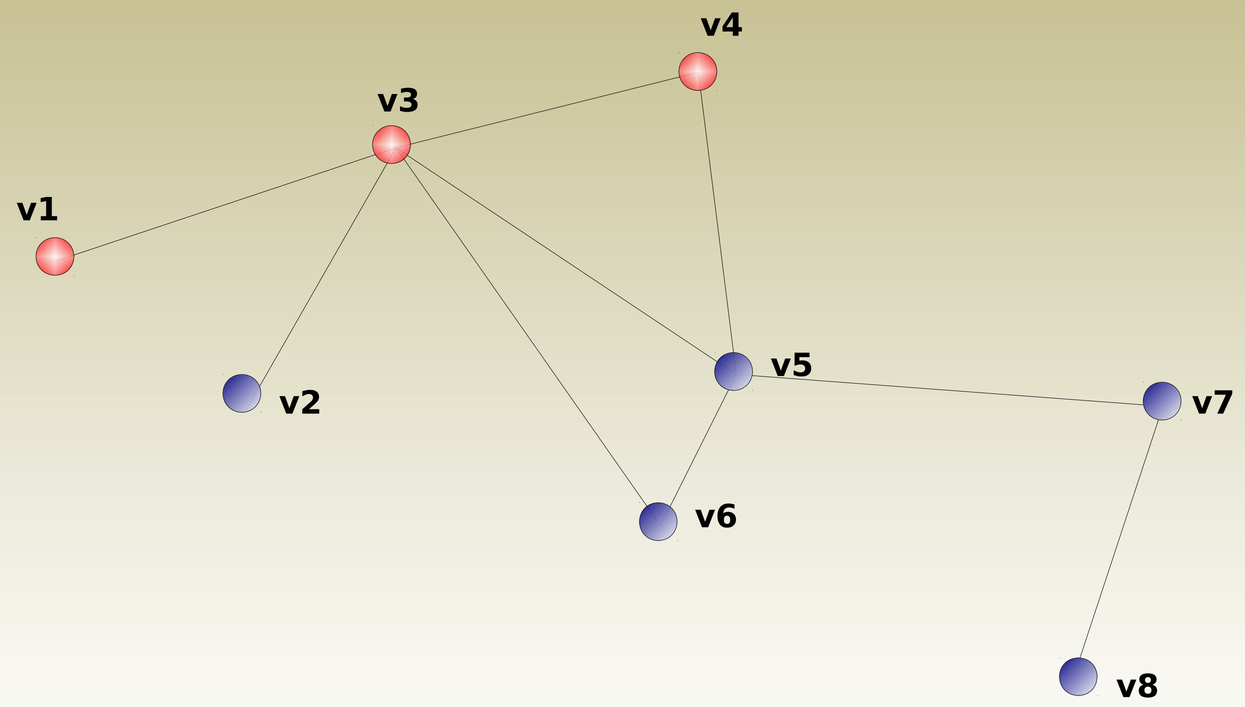
# Prohledávání do hloubky ($v_1$ -> $v_8$)



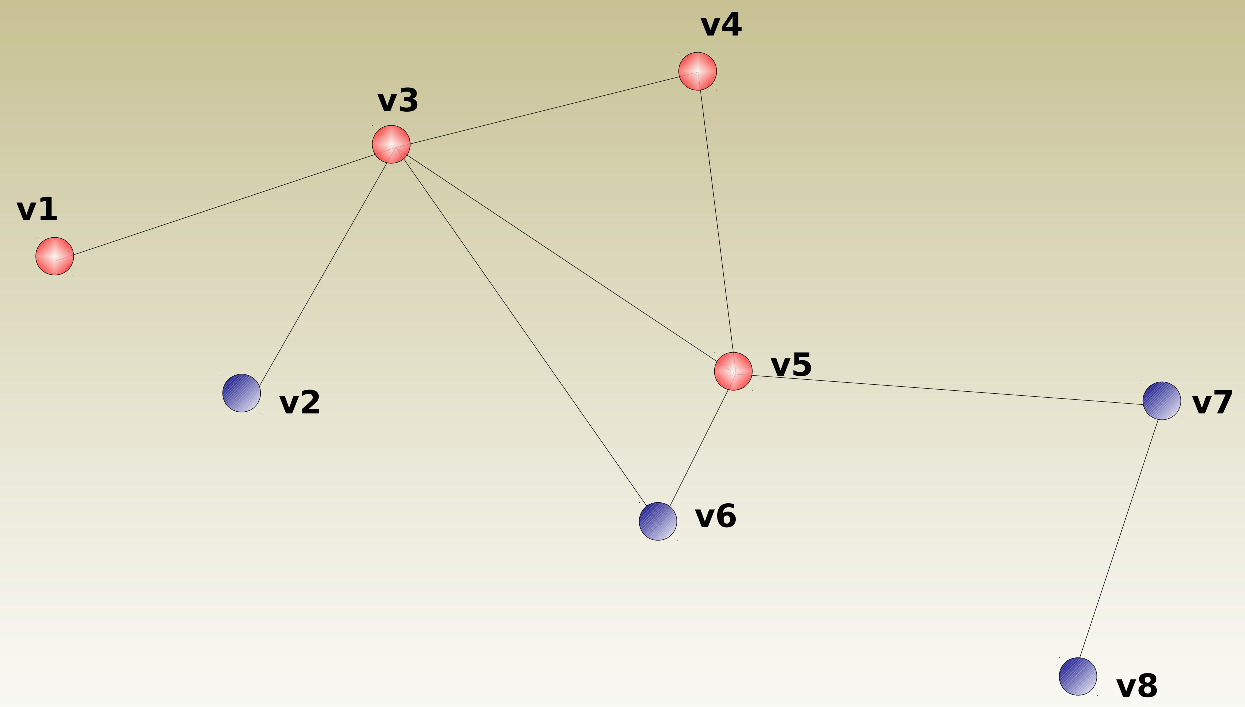
# Prohledávání do hloubky ($v_1$ -> $v_8$)



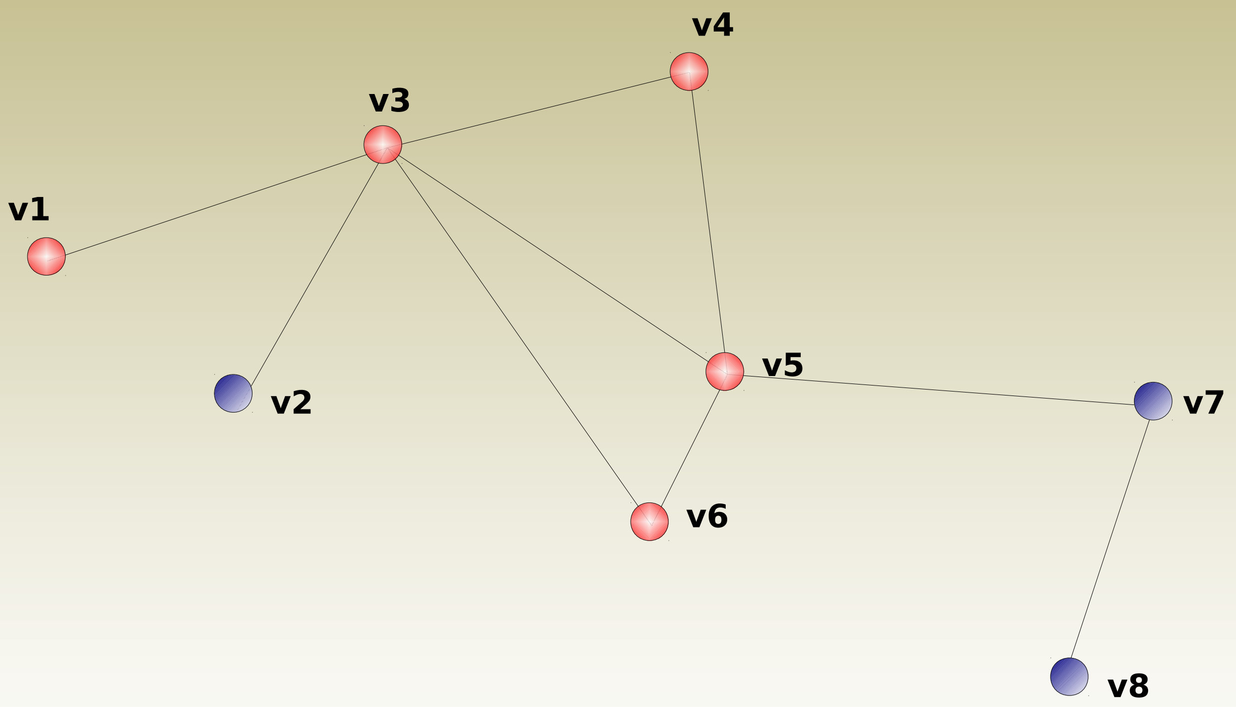
# Prohledávání do hloubky ($v_1$ -> $v_8$)



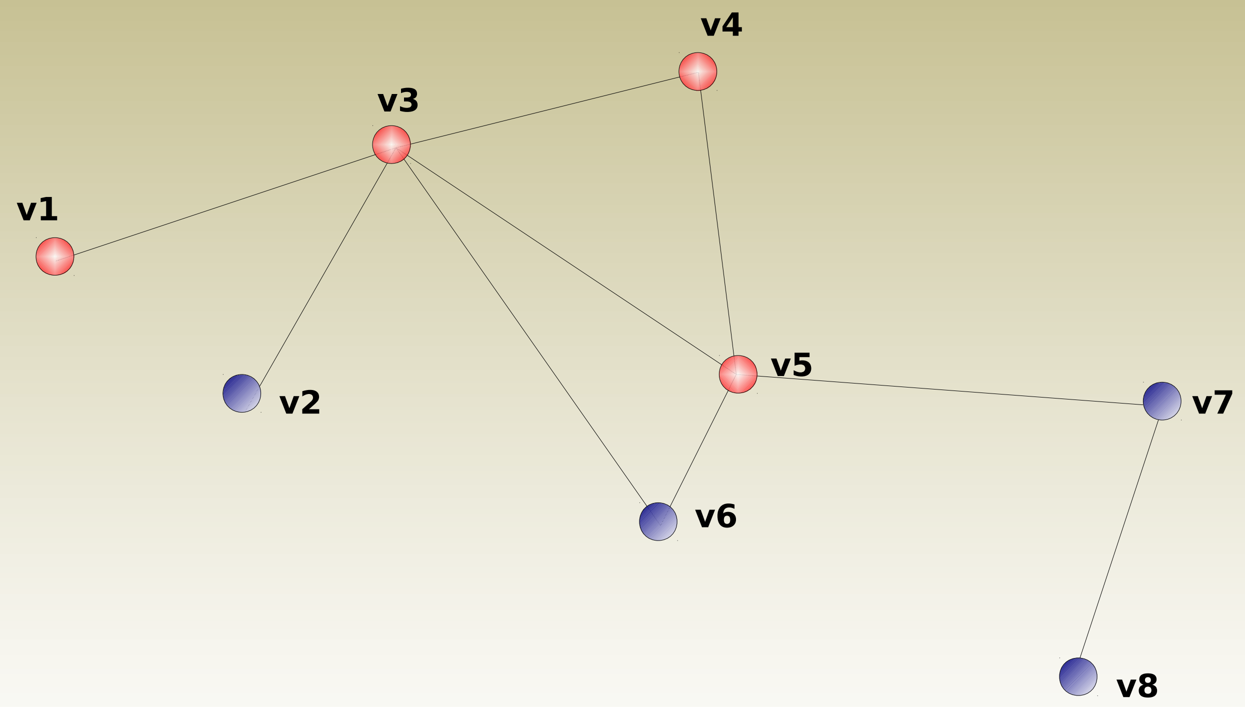
# Prohledávání do hloubky ($v_1$ -> $v_8$)



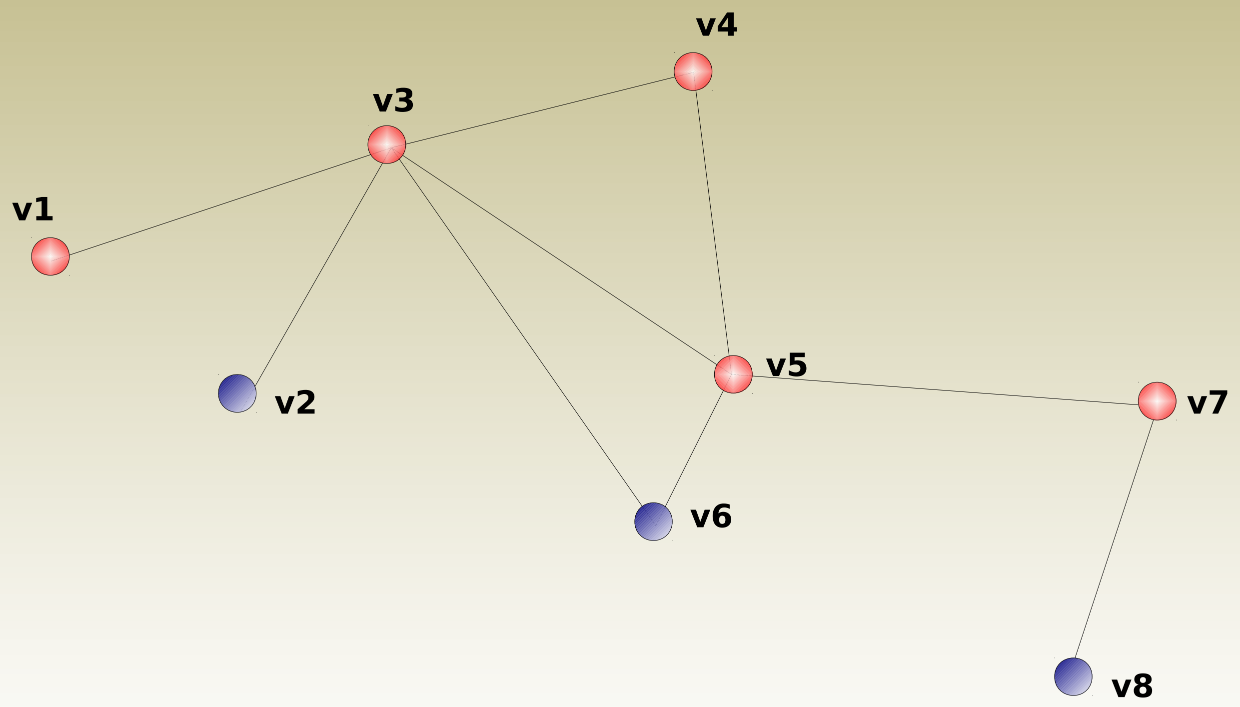
# Prohledávání do hloubky ($v_1$ -> $v_8$)



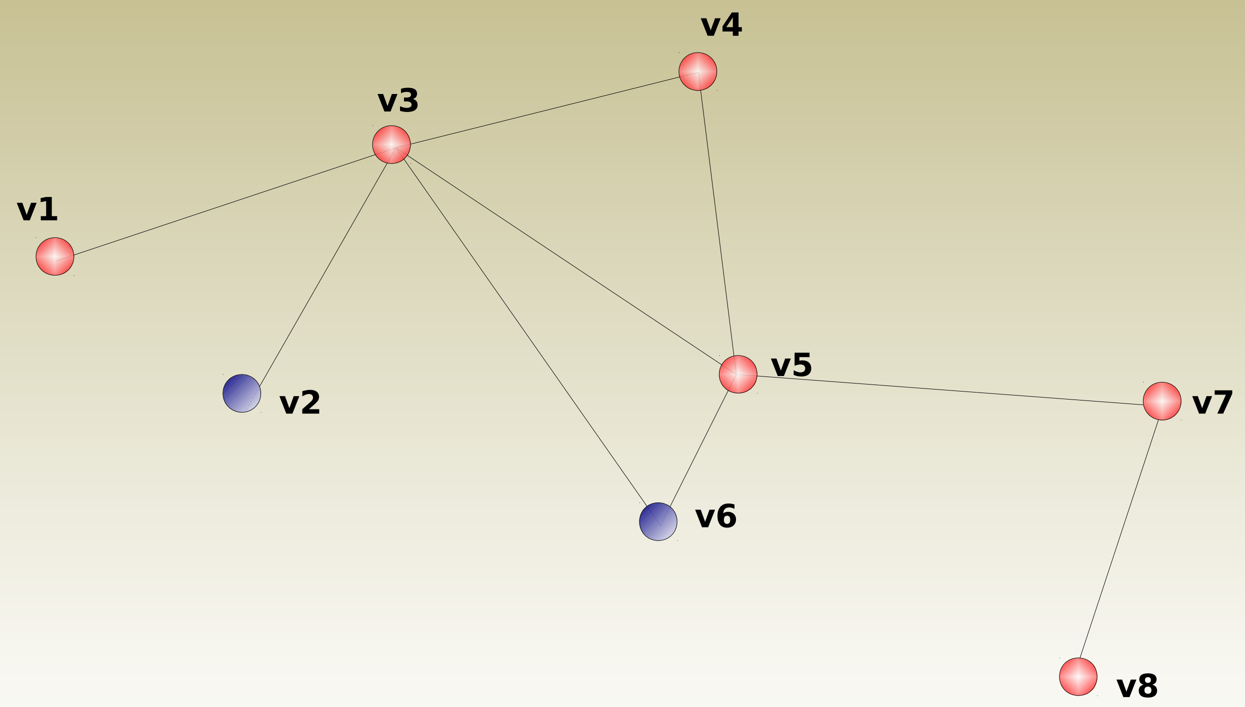
# Prohledávání do hloubky ($v_1$ -> $v_8$)



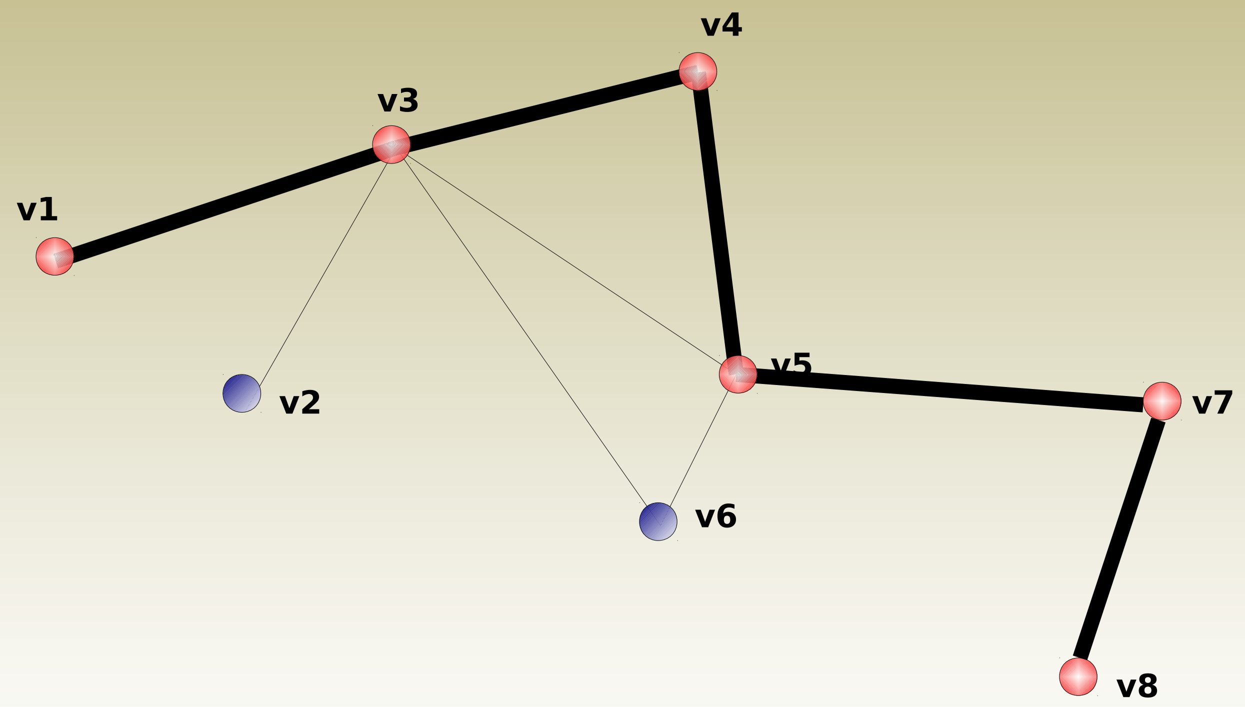
# Prohledávání do hloubky ($v_1$ -> $v_8$)



# Prohledávání do hloubky ($v_1$ -> $v_8$)



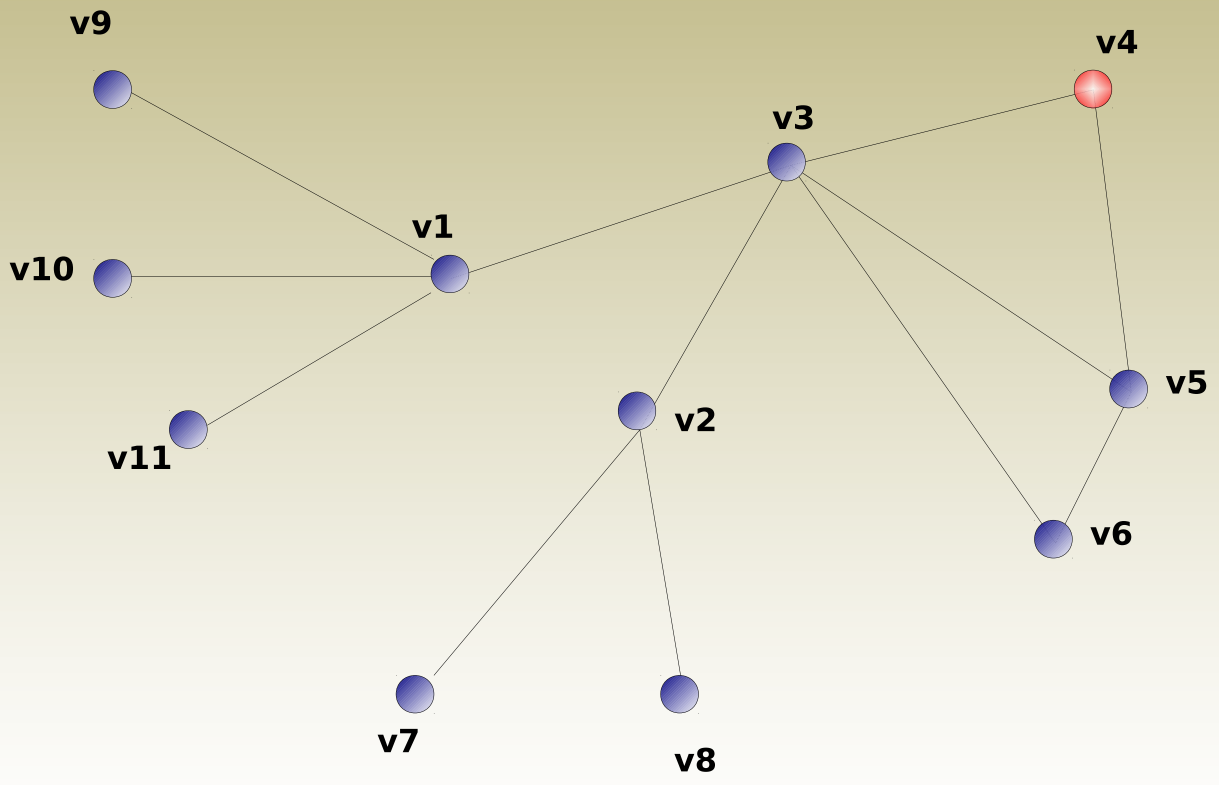
# Prohledávání do hloubky ($v_1$ -> $v_8$)



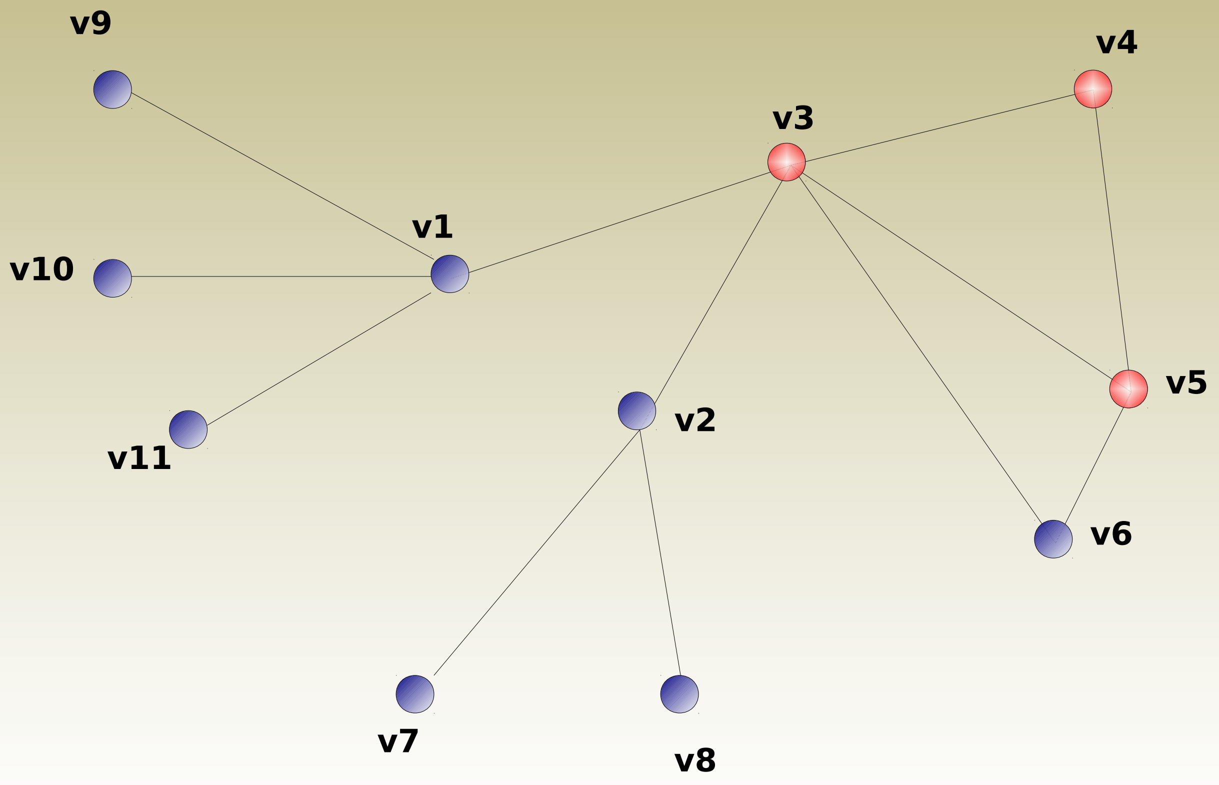
# Prohledávání grafu do šířky

- Odlišnost tohoto algoritmu spočívá v tom, že se z dosaženého vrcholu „rozhlédneme“ najednou do všech sousedních vrcholů

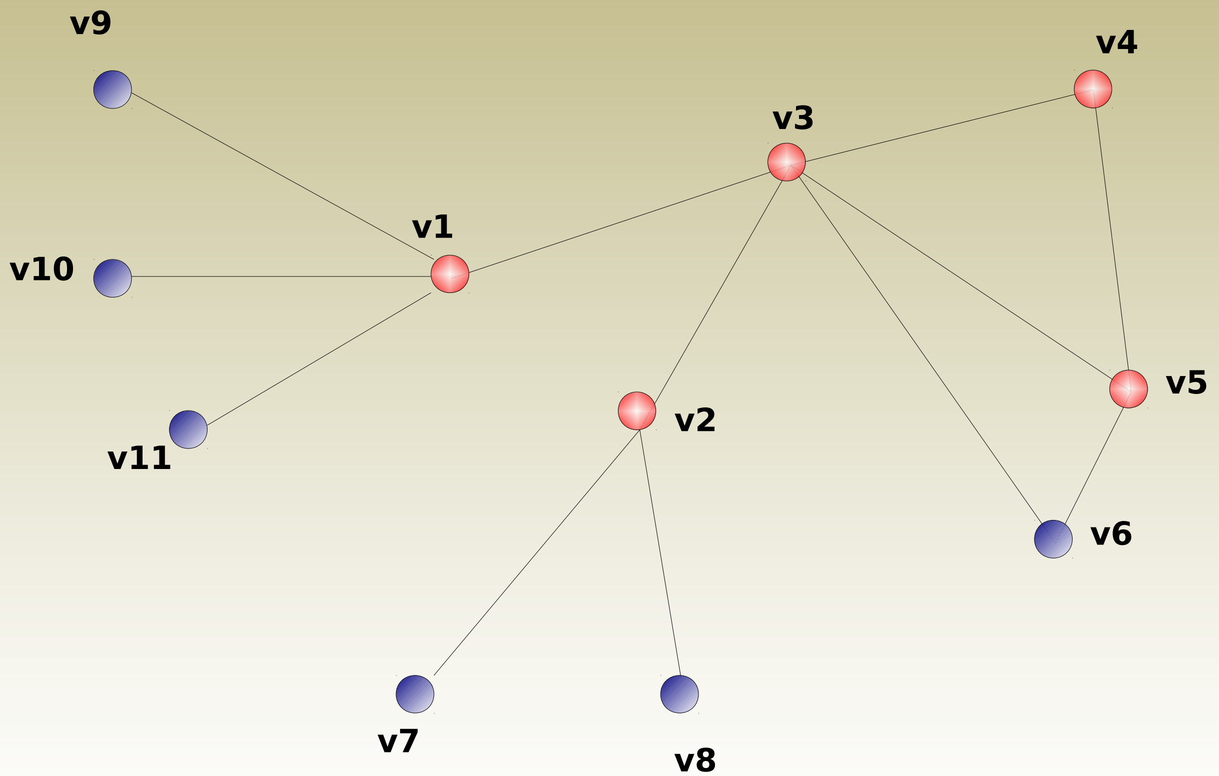
# Prohledávání do šířky ($v_4$ -> $v_8$)



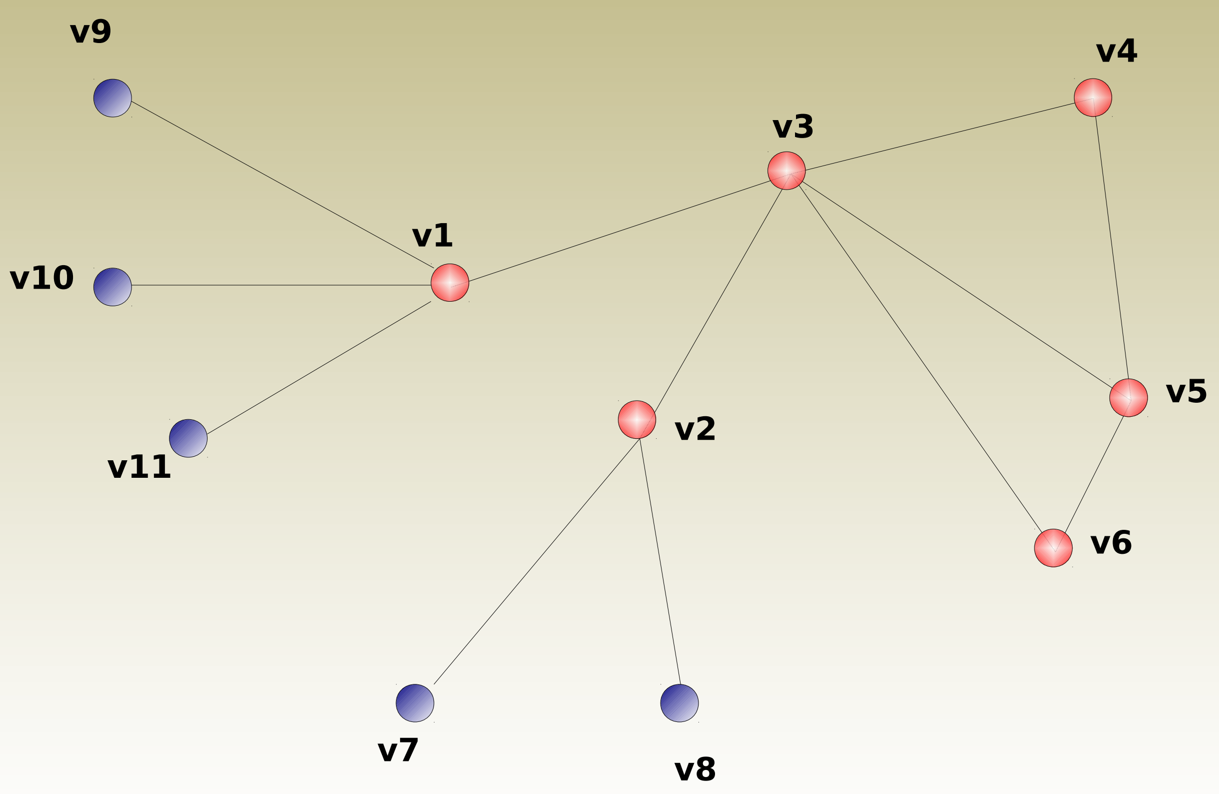
# Prohledávání do šířky ($v_4$ -> $v_8$)



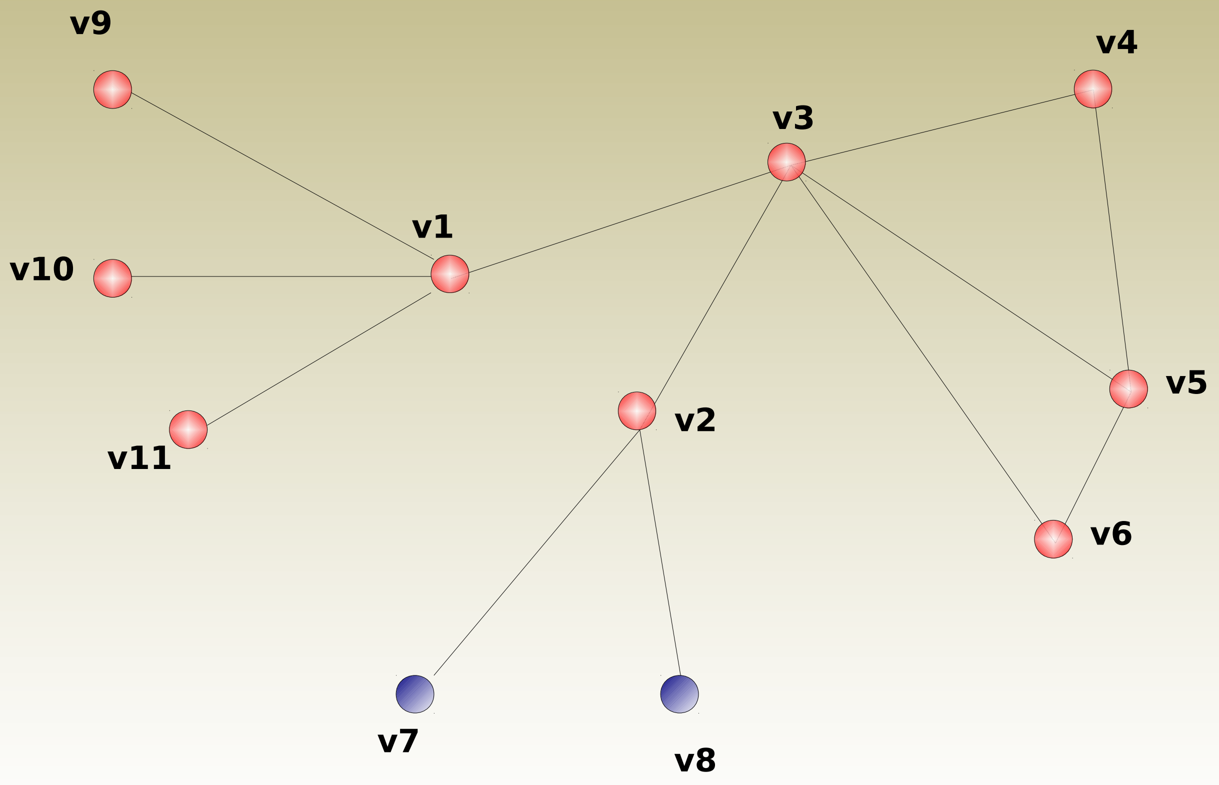
# Prohledávání do šířky ($v_4$ -> $v_8$)



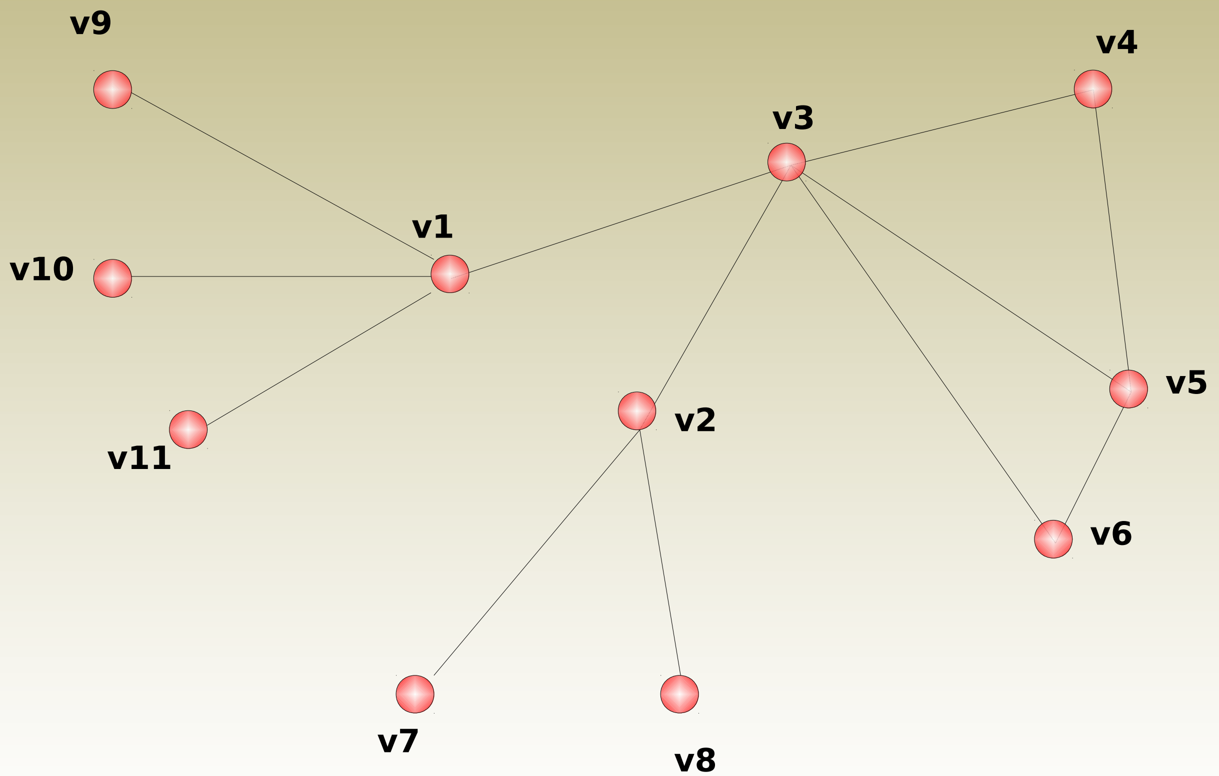
# Prohledávání do šířky ($v_4$ -> $v_8$)



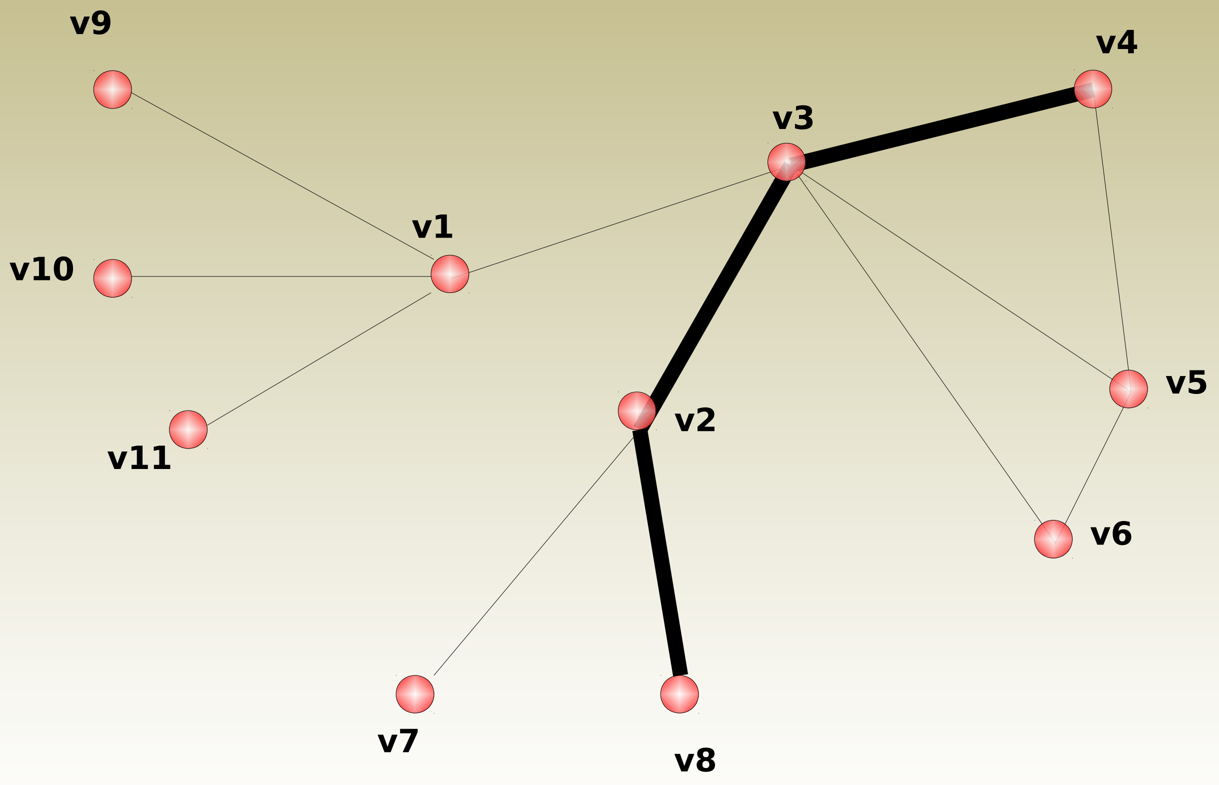
# Prohledávání do šířky ($v_4$ -> $v_8$)



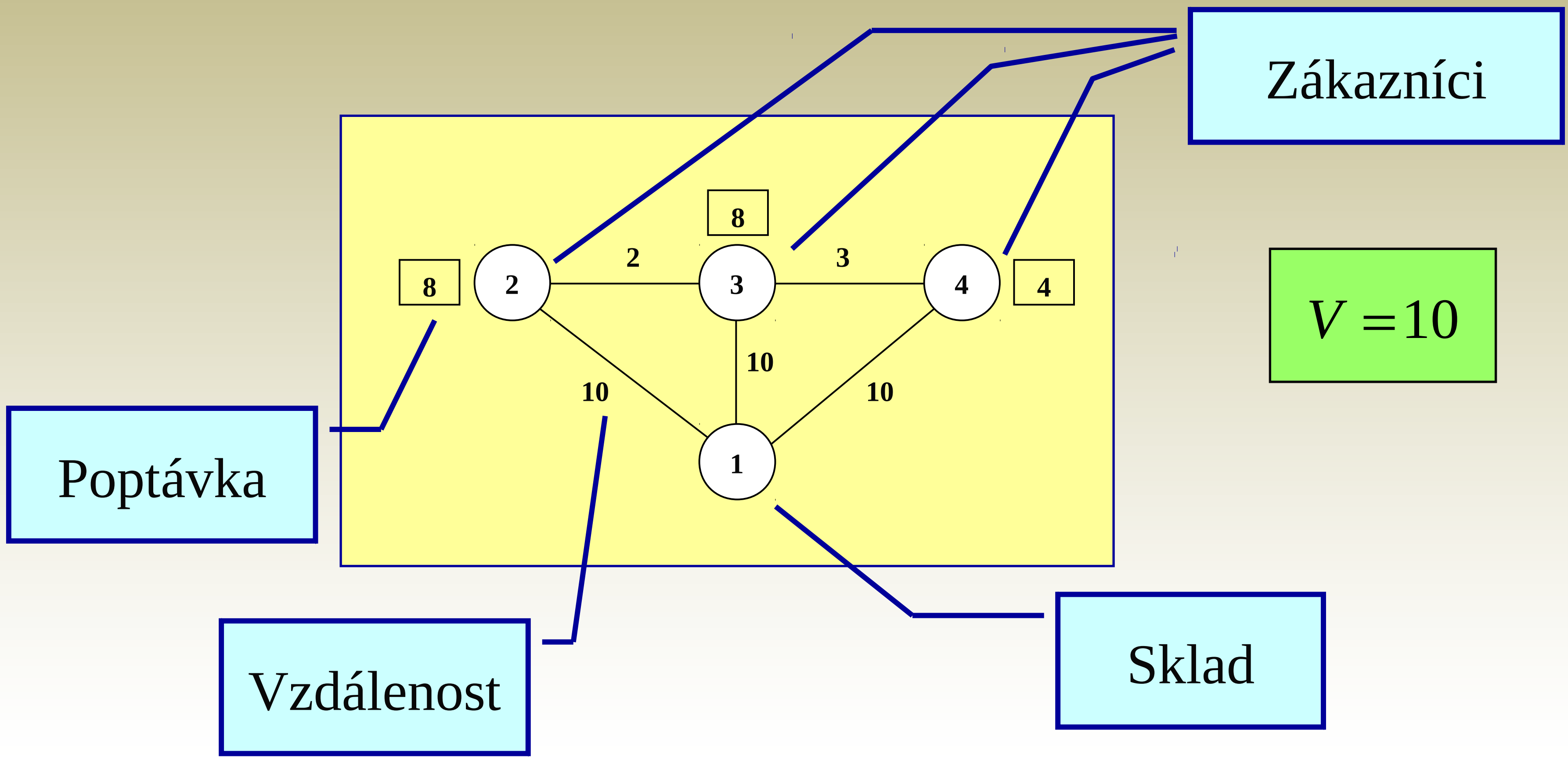
# Prohledávání do šířky ($v_4$ -> $v_8$)



# Prohledávání do šířky ($v_4$ -> $v_8$)



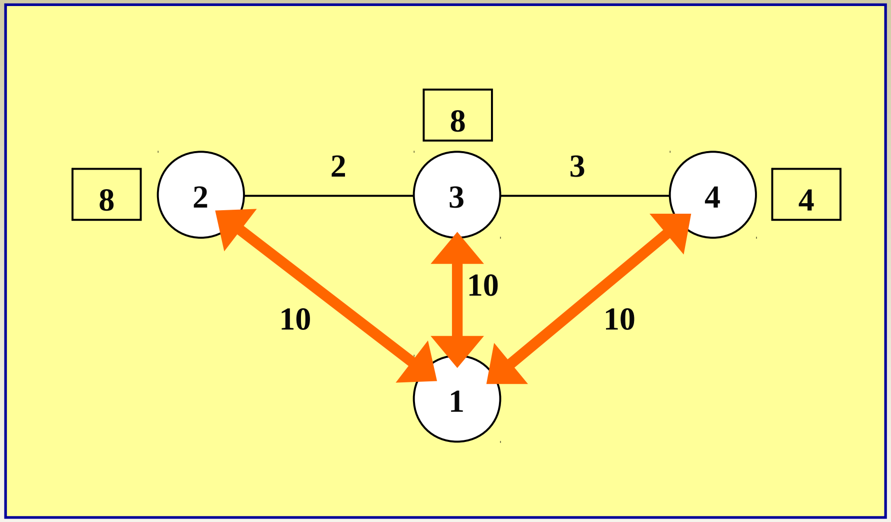
# Rozvozní problém




Převzato z: nb.vse.cz/~fabry/4EK314-prezentace.ppt

# Rozvozní problém

Řešení s **nedělenou** dodávkou



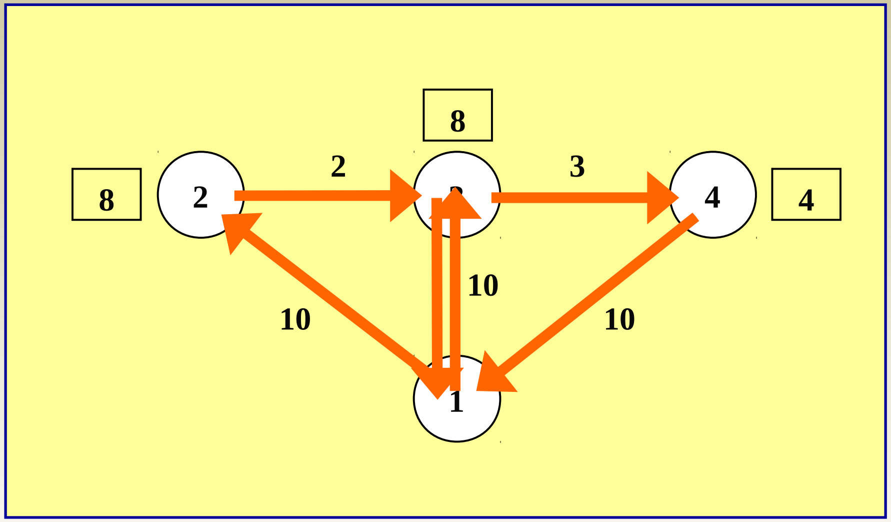

Trasy: 1-2-1; 1-3-1; 1-4-1;

Ujetá vzdálenost: (10+10)+(10+10)+(10+10) = 60

Převzato z: nb.vse.cz/~fabry/4EK314-prezentace.ppt

# Rozvozní problém

Řešení s **dělenou** dodávkou



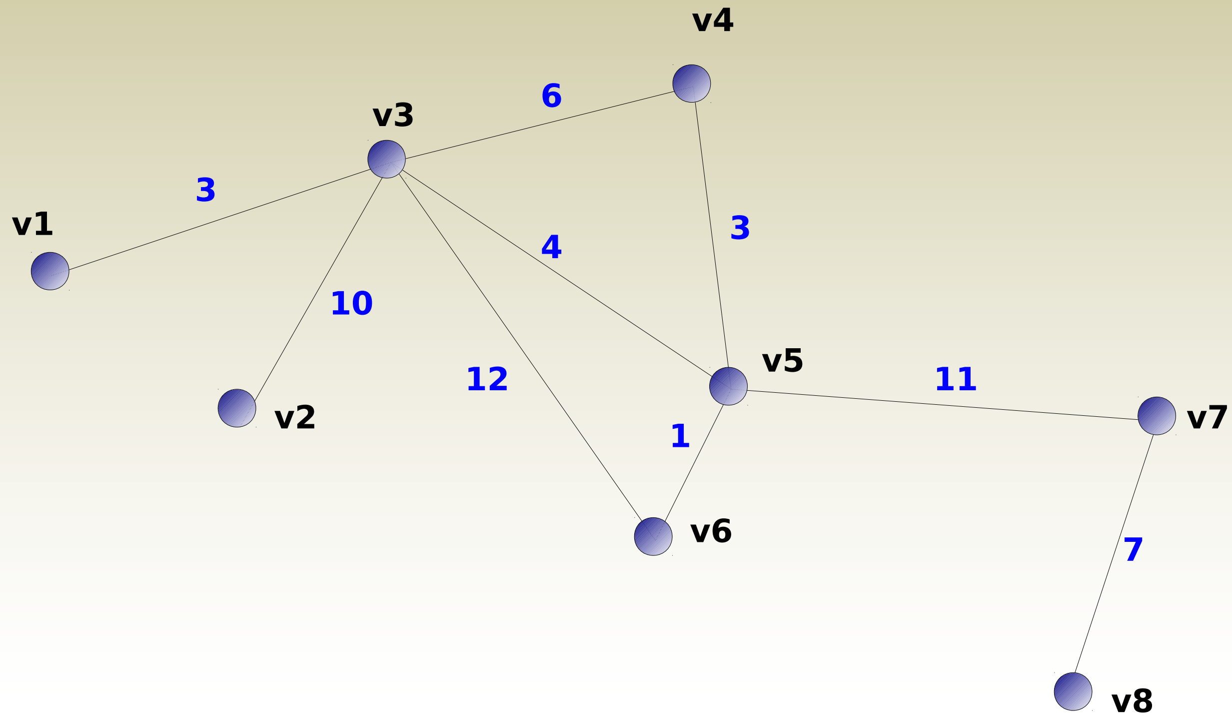

Trasy: 1-2-3-1; 1-3-4-1;

Ujetá vzdálenost: (10+2+10)+(10+3+10) =45

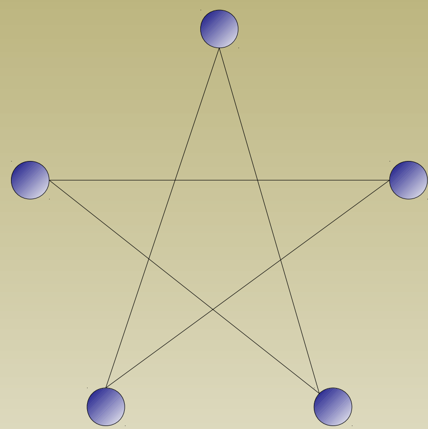
Převzato z: nb.vse.cz/~fabry/4EK314-prezentace.ppt

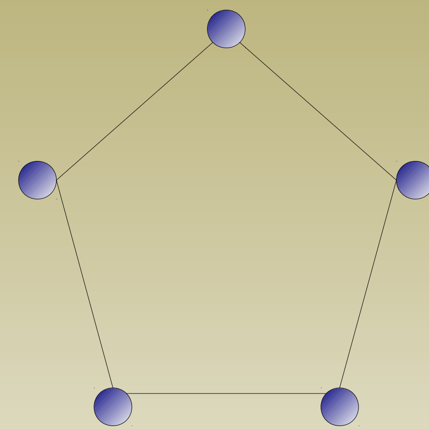
# Problém obchodního cestujícího

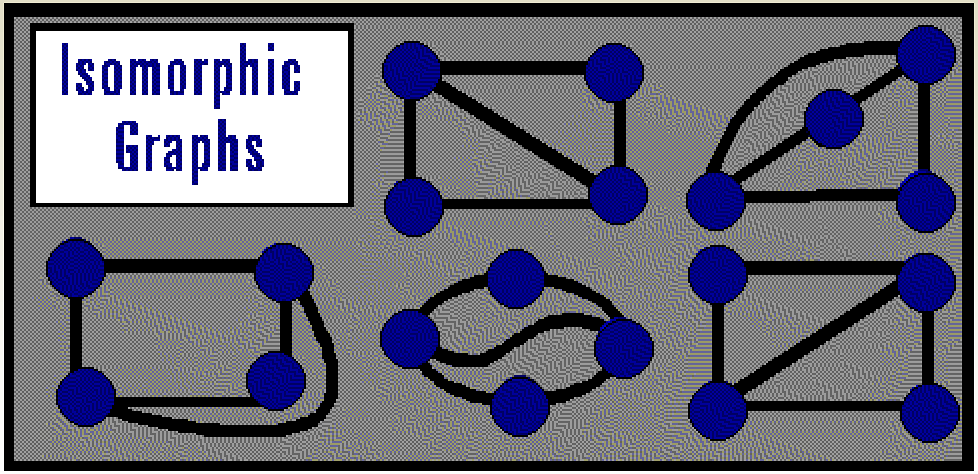
- Existuje n měst, mezi nimi silnice o známých délkách. Úkolem je najít nejkratší možnou trasu, procházející všemi městy a vracející se nazpět do výchozího města.



# Isomorfismus grafů



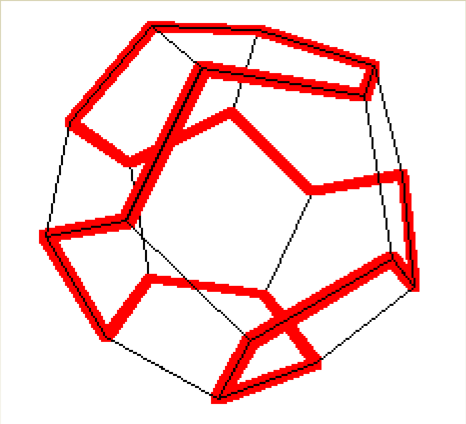


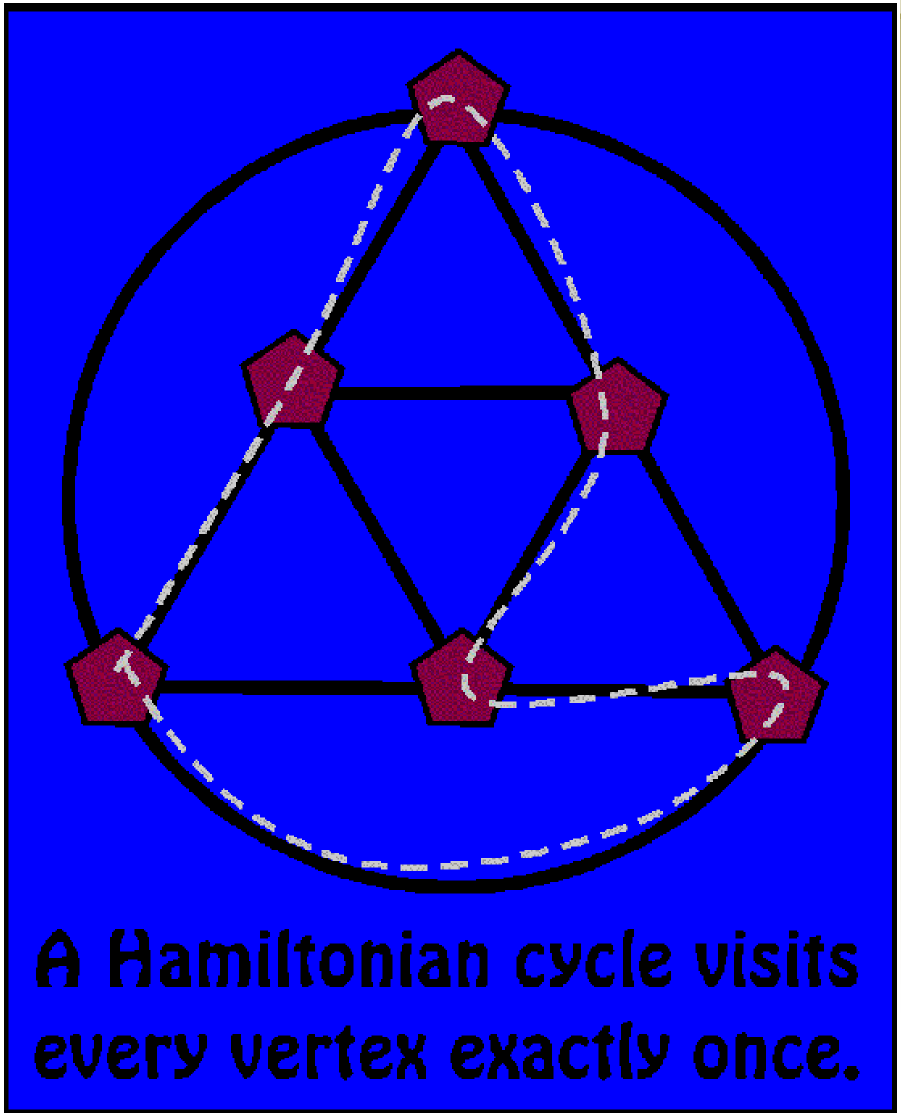



http://www.c3.lanl.gov/mega-math/workbk/graph/griso.gif

# Hamiltonův cyklus

The Icosian Game
(icosian = dvacítkový)






http://teorie-grafu.elfineer.cz/

http://www.c3.lanl.gov/

# Zdroje


- http://www.c3.lanl.gov/
- nb.vse.cz/~fabry/
- Eliška Ochodková: Grafové algoritmy, 2003
- Reinhard Diestel: Graph Theory, Electronic edition, 2000
- http://homel.vsb.cz/~kov16/talks/algoritmy/
- http://teorie-grafu.elfineer.cz/
- http://cs.wikipedia.org/wiki/Sedm_most%C5%AF_m%C4%9Bsta_Kr%C3%A1lovce