Obsah prezentace

- Základní pojmy v teorii o grafech
- Úlohy a prohledávání grafů
- Hledání nejkratších cest

Základní pojmy

Vrchol grafu: {množina V}

Je to styčná vazba v grafu, nazývá se též uzlem, prvkem nebo bodem v grafu.

Hrana grafu: {množina E}

Reprezentuje spojení jednotlivých vrcholů. Toto spojení vyjadřuje nějaký vztah mezi vrcholy.

$$\boxed{\varepsilon : E \rightarrow V^2}$$

Základní pojmy

Orientovaný a neorientovaný graf

V orientovaném grafu jsou vždy orientované hrany, tj. hrany s definovaným počátečním a koncovým vrcholem.

V neorientovaných grafech se lze pohybovat přes hrany oběma směry.

$$G = (V, E, \varepsilon)$$

Hrany i vrcholy jsou v četných aplikacích ohodnoceny → **Ohodnocený orientovaný (neorientovaný) graf**

Základní pojmy

Sled:

Orientovaný

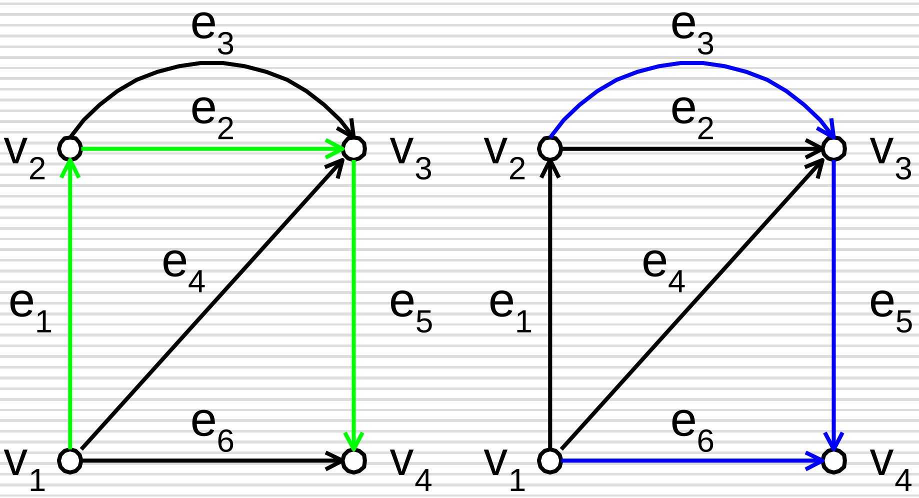
$v_1, e_1, v_2, e_2, v_3, e_5, v_4$

Neorientovaný

$v_2, e_3, v_3, e_5, v_4, e_6, v_1$

Není sledem

$v_1, e_2, v_3, e_5, v_2, e_1, v_4$



Posloupnost vrcholů a hran jak jdou za sebou.

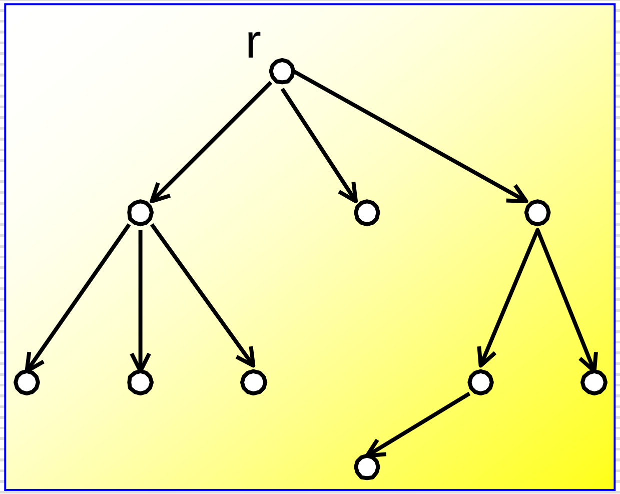
Základní pojmy

Tah

Sled, kde se neopakují hrany

Cesta

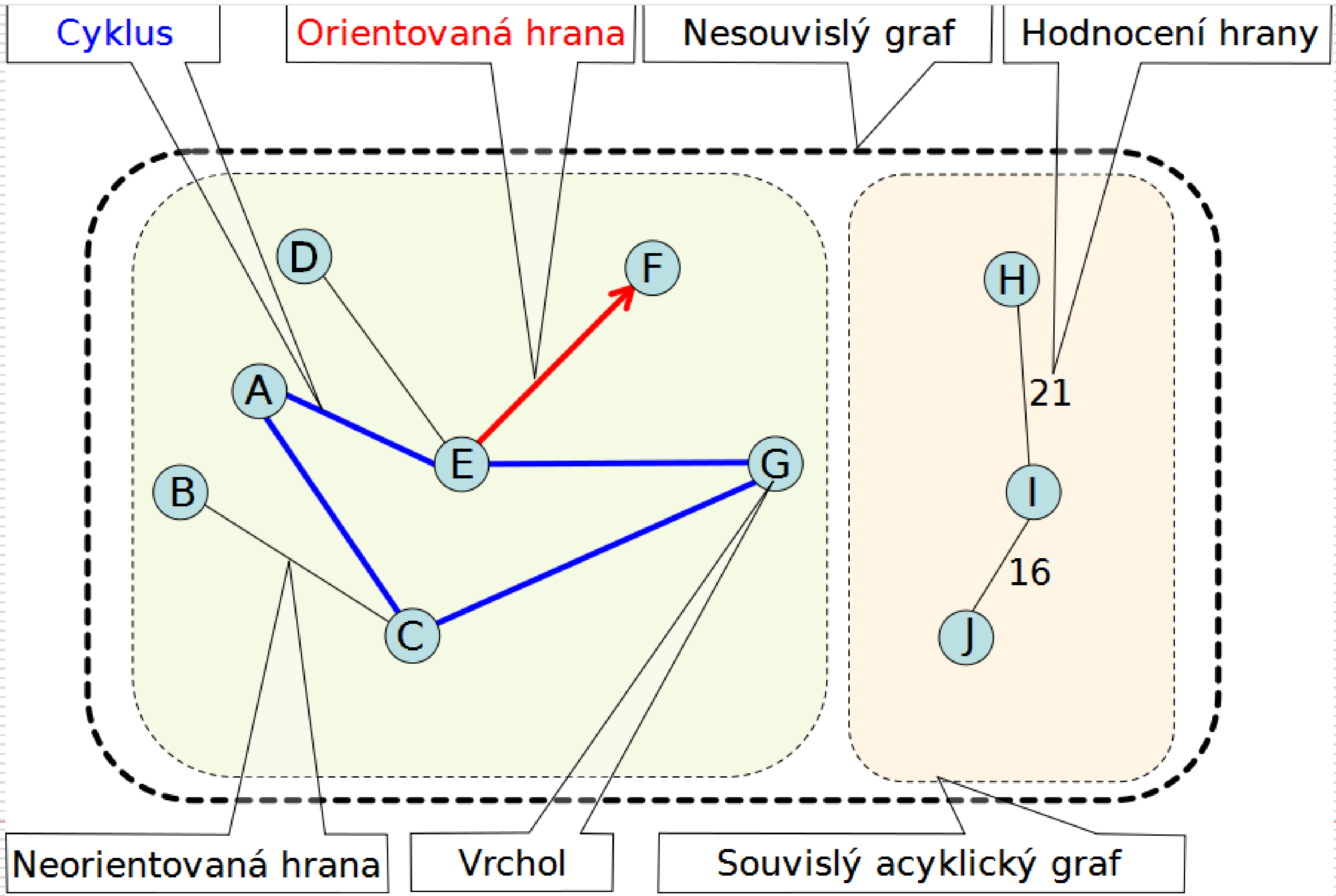
Sled, kde se neopakují vrcholy



Kořenový strom

- orientovaný graf, kde existuje vrchol r (kořen), ze kterého jsou všechny vrcholy dostupné a nevede do něj žádná hrana.

Ukázka grafu – základní pojmy



Speciální pojmy

- Hamiltonovská cesta:

  Cesta, která projde všemi vrcholy a každým pouze jedenkrát (turista).

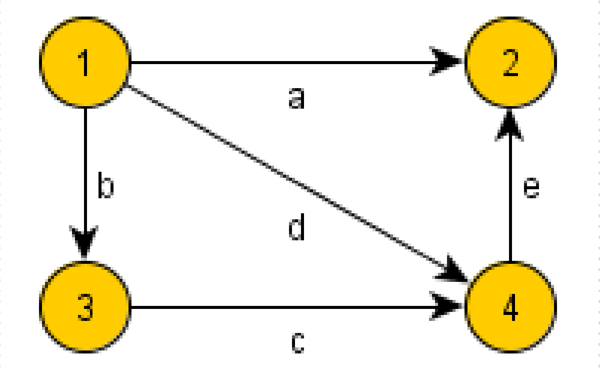
- Eulerův tah:

  Tah, který projde všemi hranami a každou pouze jedenkrát (sedm mostů v Königsbergu).

Popis grafu

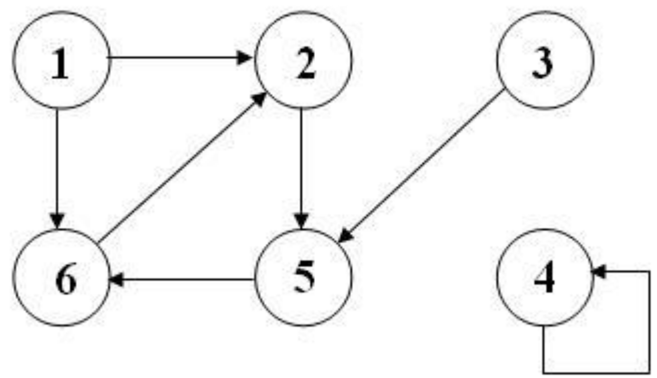
- Incidenční maticí – orientace hran (+1, -1)
- Matice sousednosti – počet hran mezi sousedy
- Spojové seznamy – seznamy následníků
- Matice délek – délka hrany mezi vrcholy (i,j)
- Matice vzdáleností

Incidenční matice

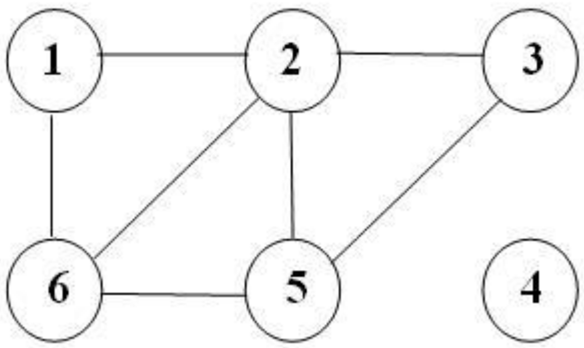


$$\begin{array}{c|ccccc} & a & b & c & d & e \\ \hline 1 & 1 & 1 & 0 & 1 & 0 \\ 2 & -1 & 0 & 0 & 0 & -1 \\ 3 & 0 & -1 & 1 & 0 & 0 \\ 4 & 0 & 0 & -1 & -1 & 1 \end{array}$$

Matice sousednosti

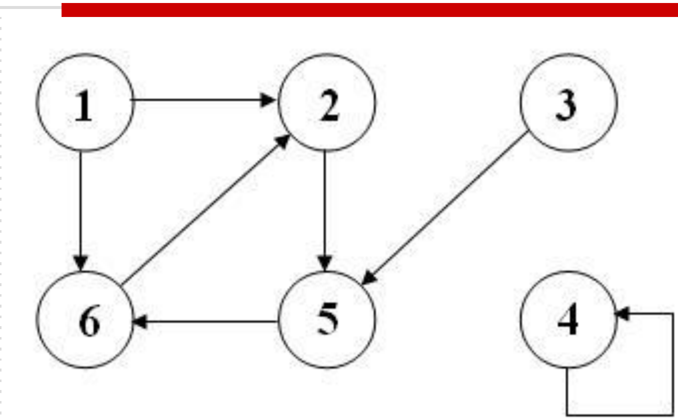


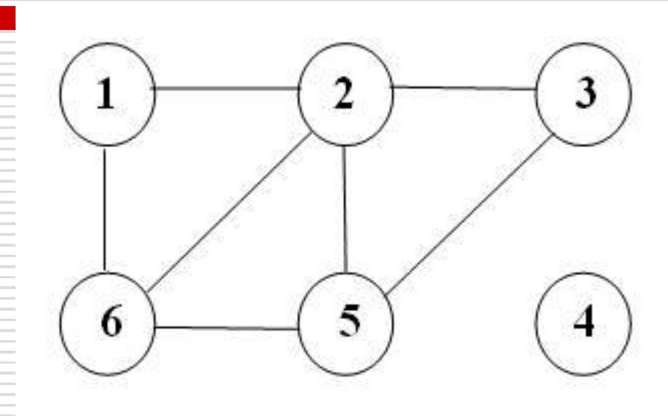
	1	2	3	4	5	6
1	0	1	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	0
4	0	0	0	1	0	0
5	0	0	0	0	0	1
6	0	1	0	0	0	0

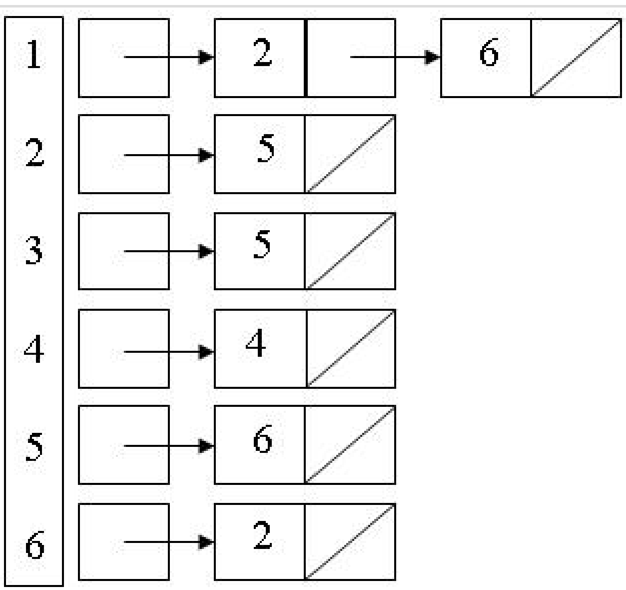


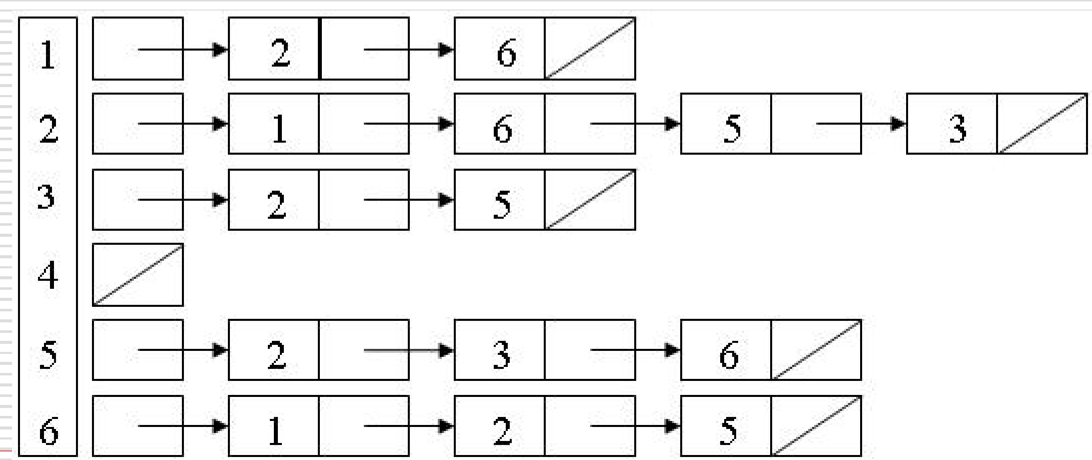
	1	2	3	4	5	6
1	0	1	0	0	0	1
2	1	0	1	0	1	1
3	0	1	0	0	1	0
4	0	0	0	0	0	0
5	0	1	1	0	0	1
6	1	1	0	0	1	0

Seznam následníků







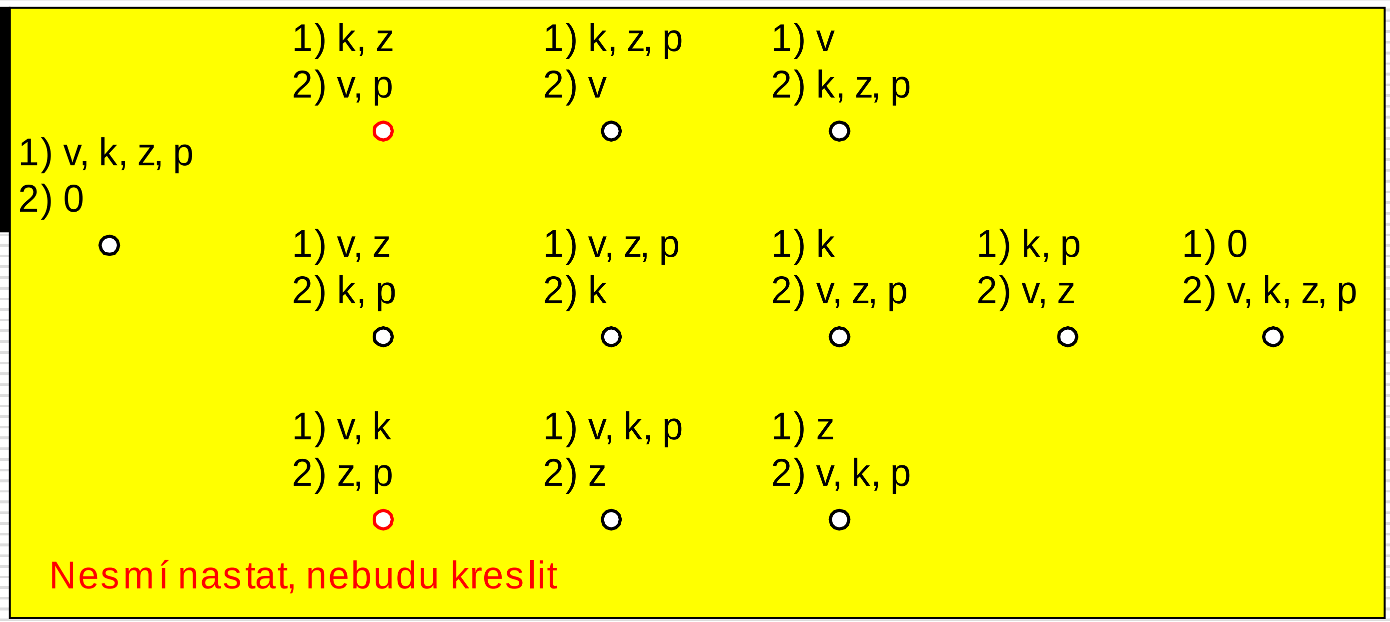


Úlohy s grafy

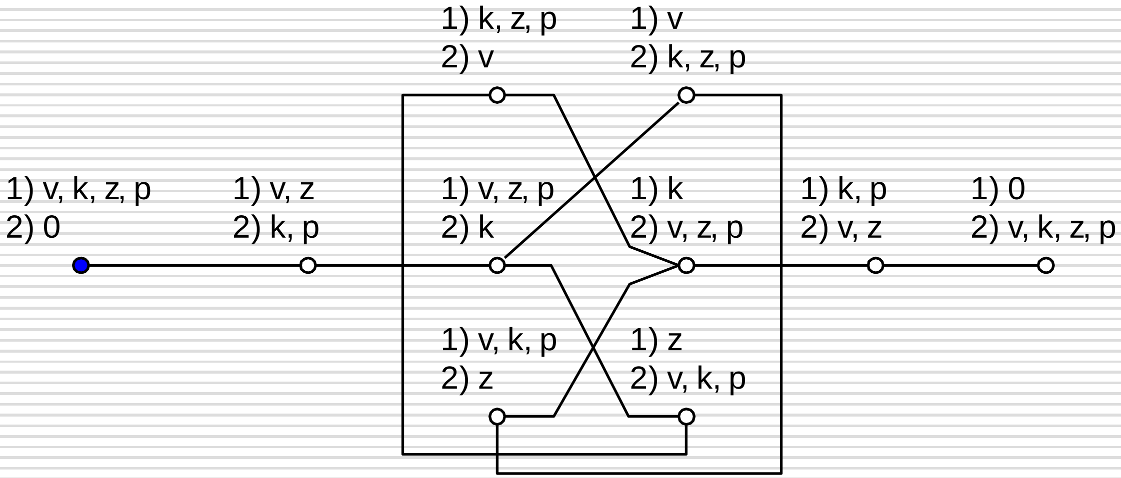
Grafické znázornění úlohy je názorné a v jednoduchých případech lze odhalit řešení i bez použití jakéhokoliv algoritmu.

Úloha pro převozníka:

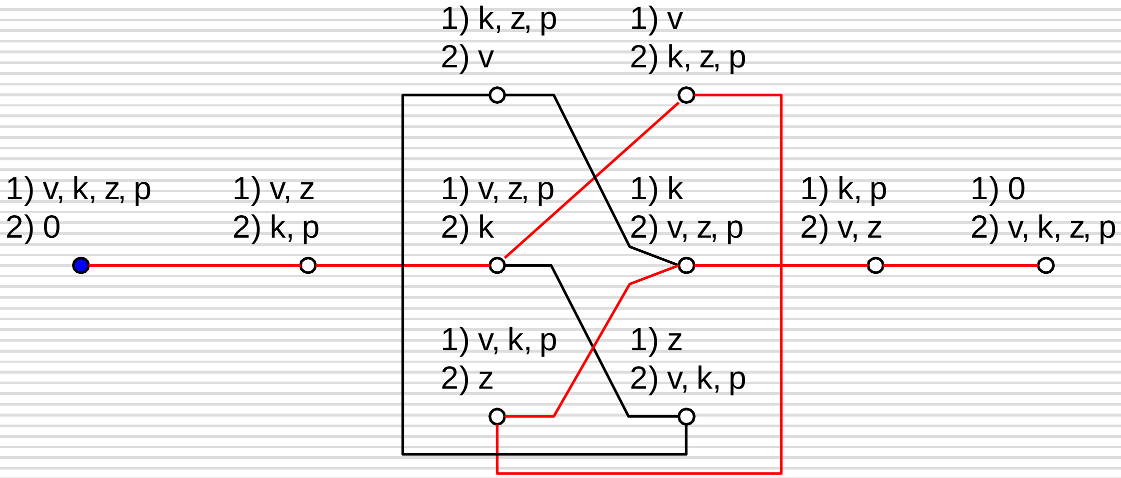
Vlk, koza, zelí



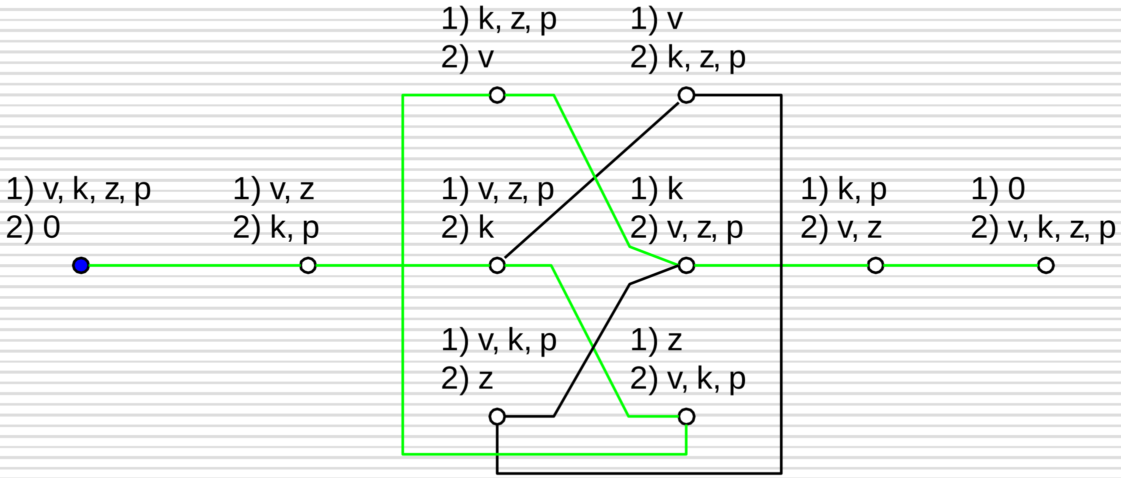
Vlk, koza, zelí



Vlk, koza, zelí



Vlk, koza, zelí



Prohledávání grafů

Úkolem prohledávání je hledání cesty z daného výchozího vrcholu do jednotlivých vrcholů grafu. To může pomoci i při vytváření grafu pro danou úlohu.

Tři způsoby prohledávání:

1) Značkování vrcholů
2) Prohledávání do šířky
3) Prohledávání do hloubky

Značkování vrcholů

Vrcholům přiřazujeme značky, pokud vrchol značku má, pak do něj vede cesta z daného výchozího vrcholu => vyjadřuje možnost

Výsledek lze převézt na kořenový strom, pokud zaznamenáme každou použitou hranu a její počáteční a koncový vrchol.

U neorientovaných grafů má tato metoda malý význam (pouze u velmi složitých, kde není zřejmé propojení jednotlivých vrcholů částí grafu).

Algoritmus značkování vrcholů

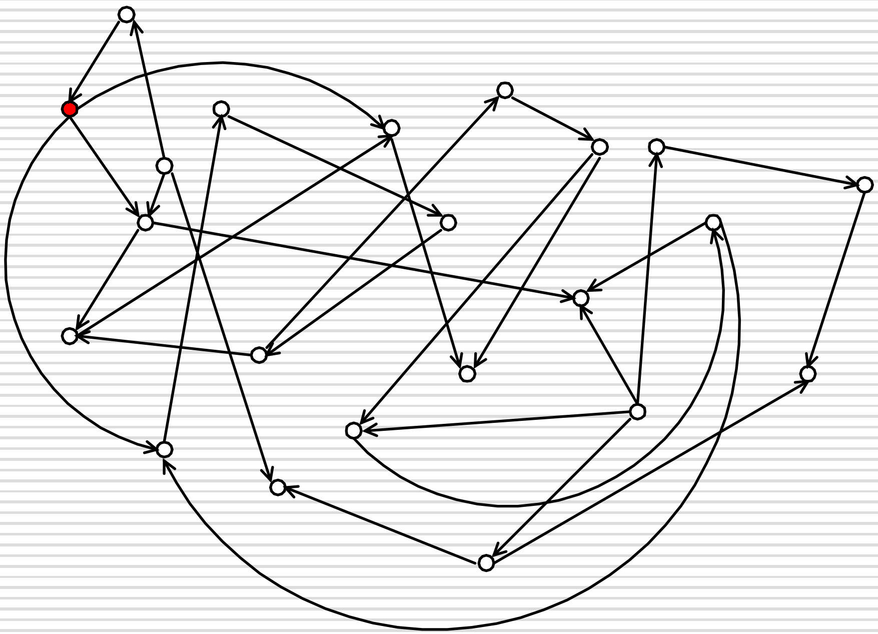
Popis

metoda pro prohledávání grafu od výchozího vrcholu *v*, při provádění algoritmu se udělují značky vrcholům, do kterých vede cesta z výchozího vrcholu

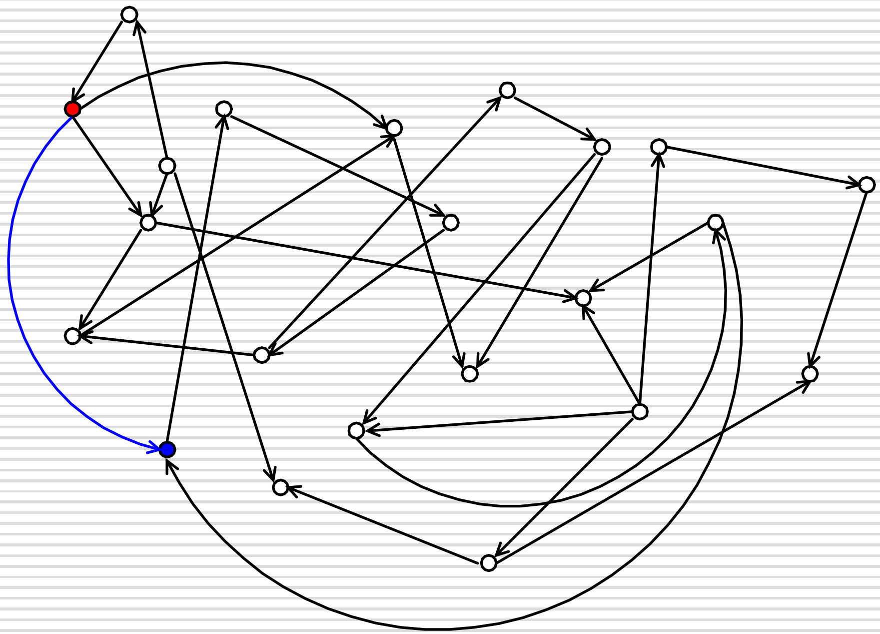
Algoritmus

1. všechny vrcholy v grafu se nastaví jako neoznačkované a označkuje se vrchol *v*
2. z pracovního vrcholu se vybere hrana *h* taková, že její počátek je ve vrcholu se značkou a její konec je ve vrcholu bez značky
3. je-li nalezena takováto hrana, pak se pokračuje krokem 4, jinak algoritmus končí a graf je prohledán
4. koncový vrchol hrany *h* (bez značky) se označkuje, pokračuje se krokem 2

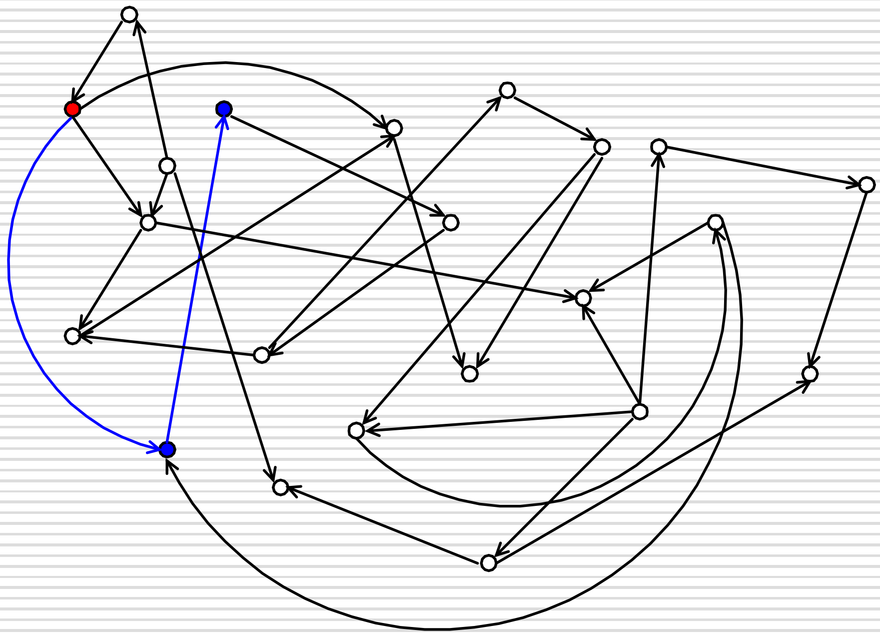
Značkování vrcholů



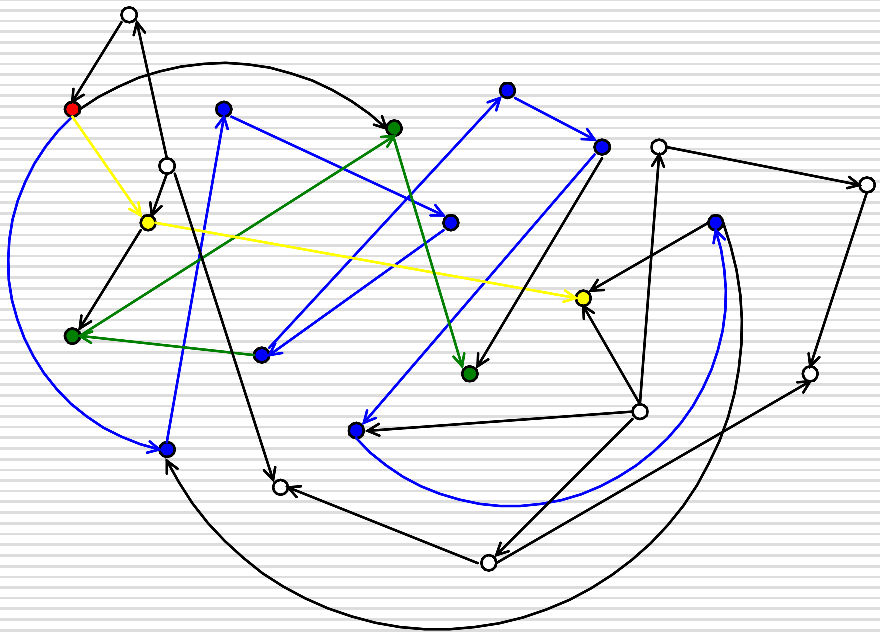
Značkování vrcholů



Značkování vrcholů



Značkování vrcholů



Značkování vrcholů

Vlastnosti:

- Odpovídá na otázku: „Je možné?‟
- Jednoduchý algoritmus
- Malé časové nároky: max V(G) – 1
- Nelze jej použít při hledání cest s určitými vlastnostmi (nejkratší, nejdelší).

Prohledávání grafu do šířky

Algoritmus lze přirovnat ke štěpné reakci. Probíhá tak, že počátečnímu vrcholu označíme všechny následníky, pak označíme následníky následníků atd.

Metoda vede k nalezení nejkratší cesty, za předpokladu, že hrany mají stejnou hodnotu => nejmenší počet tahů.

Algoritmus prohledávání do šířky

Popis

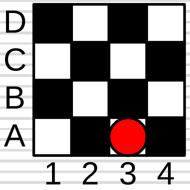
Tato metoda slouží k prohledávání grafu a to tím způsobem, že se označkují všichni následníci výchozího vrcholu a dále se označkují všichni následníci těchto následníků atd.. Tato metoda se využívá pro nalezení nejkratších cest = cest s nejmenším počtem hran (v případě hran stejných délek).

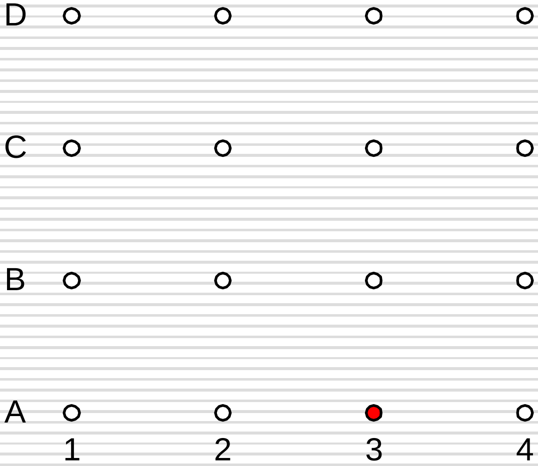
Algoritmus

- máme orientovaný graf *G* s výchozím vrcholem *v* a hledáme všechny vrcholy *i*, do kterých vede z vrcholu *v* orientovaná cesta, máme zjistit nejkratší cestu a její vzdálenost pro každý vrchol
- každý vrchol bude při prohledávání označkován a budou mu přiděleny hodnoty : vrcholu, ze kterého jsme do něj přišli a vzdálenost od výchozího vrcholu, dále bude zapotřebí seznam typu fronta, do kterého budeme ukládat vrcholy, které byly označkovány, ale ještě nebyli prohledáváni jejich následníci
- při popisu algoritmu označme funkci vzdálenosti jako *V(i)*, funkci předchozího vrcholu jako *P(i)* a frontu nazveme jako *fronta*

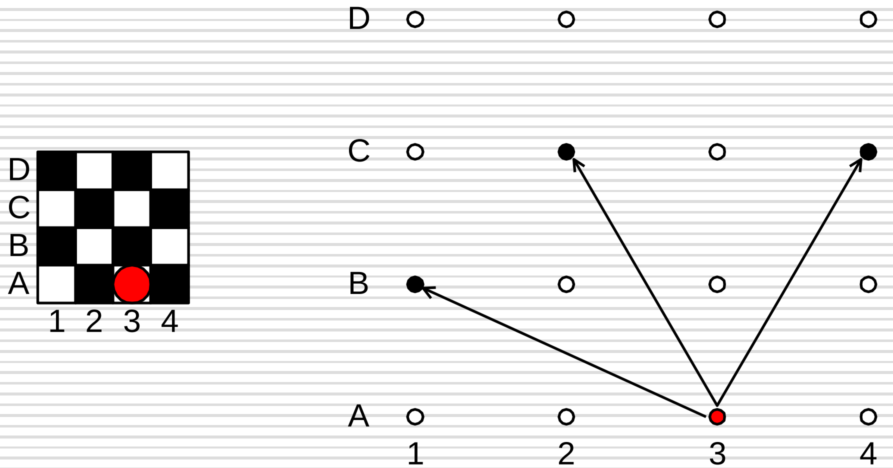
1. přidělíme značku vrcholu *v* a všechny ostatní vrcholy zůstanou beze značky, vrcholu *v* přidělíme vzdálenost *V(v) = 0* a přidáme vrchol *v* do *fronty*
2. zjistíme, je-li ve *frontě* nějaký vrchol, jestliže není algoritmus končí
3. z *fronty* odebereme vrchol a budeme s ním pracovat pod názvem *i*
4. proměnnou *j* budeme postupně plnit neoznačkovanými vrcholy takovými, kde mezi vrcholem *i* a tímto vrcholem je hrana. Vrchol *j* označkujeme a přidělíme mu předchůdce *P(j) = i*, dále nastavíme vzdálenost od výchozího vrcholu *V(j) = V(i) + velikost hrany (i -> j)* a přidáme vrchol *j* do fronty a vrátíme se ke kroku 2

Prohledávání do šířky

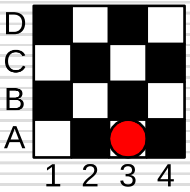


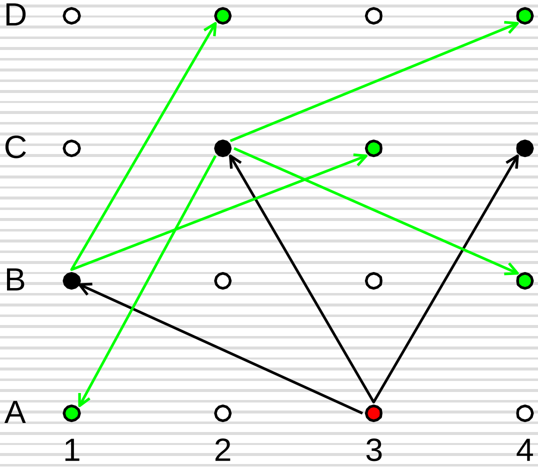


Prohledávání do šířky

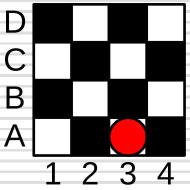


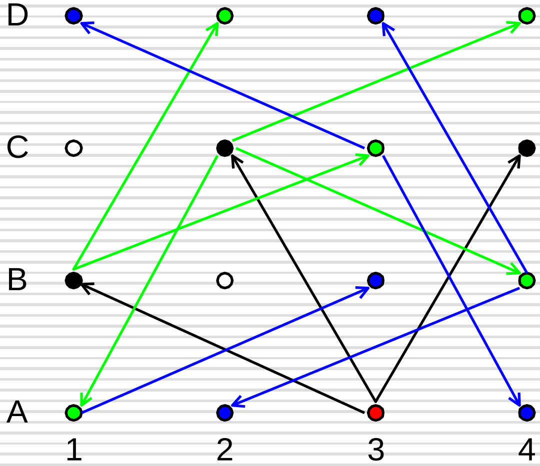
Prohledávání do šířky



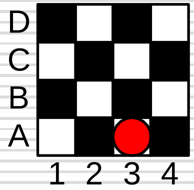


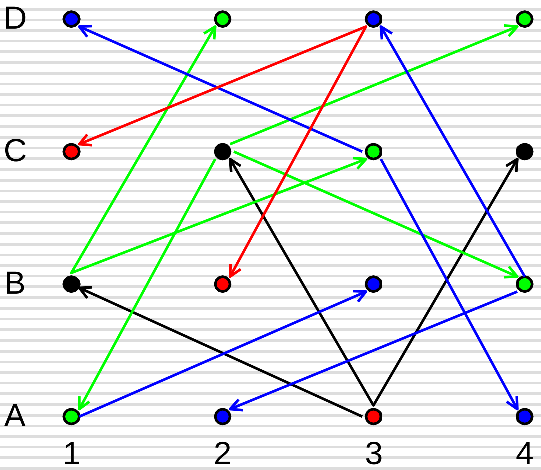
Prohledávání do šířky





Prohledávání do šířky





Prohledávání grafu do šířky

Vlastnosti:

- Podobné časové nároky jako v případě značkování vrcholů: max V(G) – 1
- Získáme kořenový strom s nejmenším počtem úrovní

Prohledávání do hloubky

Podobá se průzkumu neznámých tras. Je základem pro další metody. Jdeme do hloubky grafu kam až můžeme a pak se vracíme a hledáme odbočky, kterými lze dále pokračovat.

Algoritmus je časově náročnější než oba předešlé, avšak jeho úpravou a zaznamenáváním jednotlivých tras jej lze využít pro hledání cest splňujících nějakou speciální podmínku. Například cesta začínající v r a obsahující všechny vrcholy =Hamiltonovská cesta

Algoritmus prohledávání do hloubky

Popis

Prohledávání do hloubky najde v grafu všechny vrcholy do nichž vede orientovaná cesta, s tím že se algoritmus vrací po nalezené cestě zpět

Algoritmus

- máme orientovaný graf G s výchozím vrcholem *v*
- aktuální vrchol, ve kterém se nacházíme budeme označovat *i* a seznam *cesta* bude obsahovat cestu k aktuálnímu vrcholu

1. jako aktuální vrchol se nastaví vrchol výchozí *(i = v)*, vyprázdní se seznam *cesta* a označkuje se vrchol *v* a ostatní zůstanou beze značek
2. jako *h* označíme hranu začínající v *i*, která ještě nebyla použita, jestliže taková hrana existuje pokračujeme krokem 3, jinak krokem 5
3. koncový vrchol hrany *h* označme jako *j*, je-li vrchol *j* již označkován pokračujeme krokem 2, jinak krokem 4
4. do seznamu *cesta* přidáme hranu *h*, označkujeme vrchol *j* a nastavíme výchozí vrchol na *j (i = j)*, pokračujeme krokem 2
5. jestliže je seznam *cesta* prázdný algoritmus končí, jinak odebereme ze seznamu hranu *h*, její počáteční vrchol označíme jako *i* a pokračujeme krokem 2

Eulerův tah

- Uzavřený:
  - $\Rightarrow$ Vrátíme se do stejného vrcholu

- Otevřený:
  - $\Rightarrow$ Skončíme v jiném vrcholu

Eulerův tah

Uzavřený:

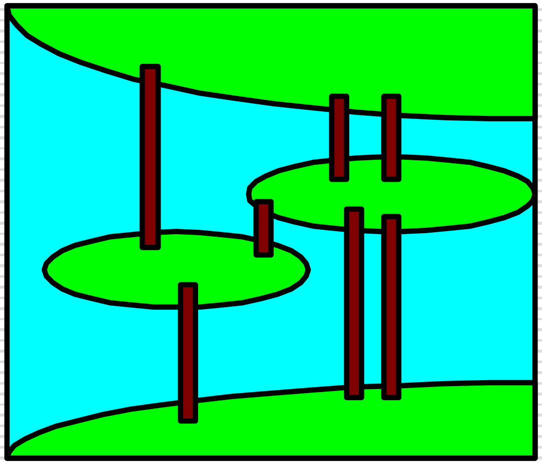
- Neorientovaný – vrcholy mají sudý stupeň
- Orientovaný – počet vstupních hran je stejný jako počet výstupních

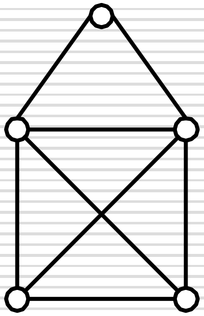
Otevřený:

- Neorientovaný – obsahuje právě dva vrcholy s lichým stupněm
- Orientovaný – stejně jako neor. + do jednoho vrcholu musí hrana vcházet, z druhého vycházet.

Eulerův tah

Sedm mostů v Königsbergu





Musí se začít ve spodních vrcholech!

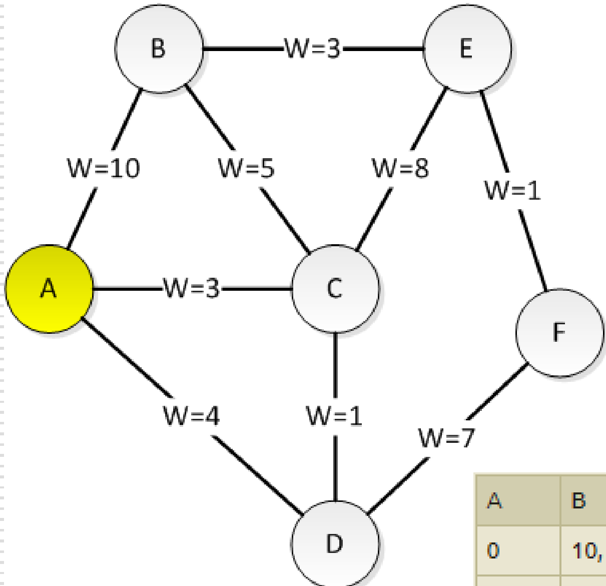
Nejkratší cesty

Úloha: najít nejkratší cestu

- z daného výchozího vrcholu do daného cílového vrcholu
- z daného výchozího vrcholu do každého vrcholu
- z každého vrcholu do daného cílového vrcholu
- mezi všemi uspořádanými dvojicemi

Lze vynechat smyčky a rovnoběžné hrany

Dijkstrův alg.



Slouží k vyhledání nejkratší cesty z počátečního uzlu do všech ostatních uzlů ohodnoceného grafu

A	B	C	D	E	F	množina X
0	10, A	3, A	4, A	max	max	A
–	8, C	–	4, A	11, C	max	A, C
–	8, C	–	–	11, C	11, D	A, C, D
–	–	–	–	11, C	11, D	A, C, D, B
–	–	–	–	–	11, D	A, C, D, B, E
–	–	–	–	–	–	A, C, D, B, E

Výsledek

A	B	C	D	E	F
0	8, C	3, A	4, A	11, C	11, D

MATICE DÉLEK

Charakterizuje délky hran mezi jednotlivými vrcholy. Definice:

$a_{ii} = 0$,

a_{ij} = délka hrany mezi vrcholy i a j;

Pokud zde hrana nevede, pak značím nekonečno.

MATICE VZDÁLENOSTÍ

Charakterizuje vzdálenosti jednotlivých vrcholů. Definice:

$u_{ii} = 0$,

$u_{ij} =$ vzdálenost mezi vrcholy i a j;

Pokud do vrcholu j nevede cesta z i pak je u_{ij} rovno nekonečnu.

MATICE VZDÁLENOSTÍ

Matice vzdáleností je přímo maticí nejkratších cest. Pokud nás nezajímá kudy cesta vede, lze použít k jejímu výpočtu následující způsoby:

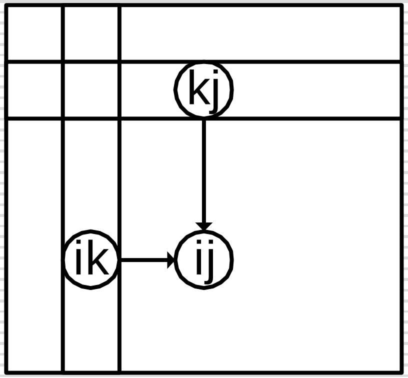
1. Upravené násobení matic
2. Floydův algoritmus

MATICE VZDÁLENOSTÍ

Floydův algoritmus:

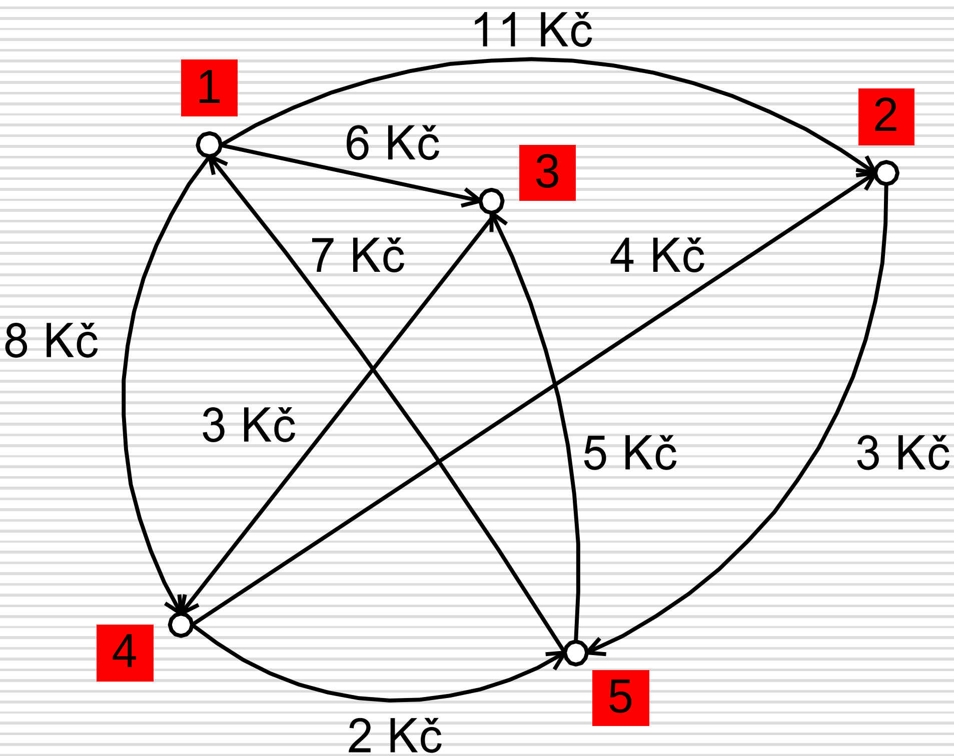
Podmínka:

$$U(i,j) > U(i,k) + U(k,j) \Rightarrow U(i,j) = U(i,k) + U(k,j)$$



MATICE VZDÁLENOSTÍ

Příklad: Minimální náklady na dopravu zboží



	1	2	3	4	5
1	0	11	6	8	∞
2	∞	0	∞	∞	3
3	∞	∞	0	3	∞
4	∞	4	∞	0	2
5	7	∞	5	∞	0

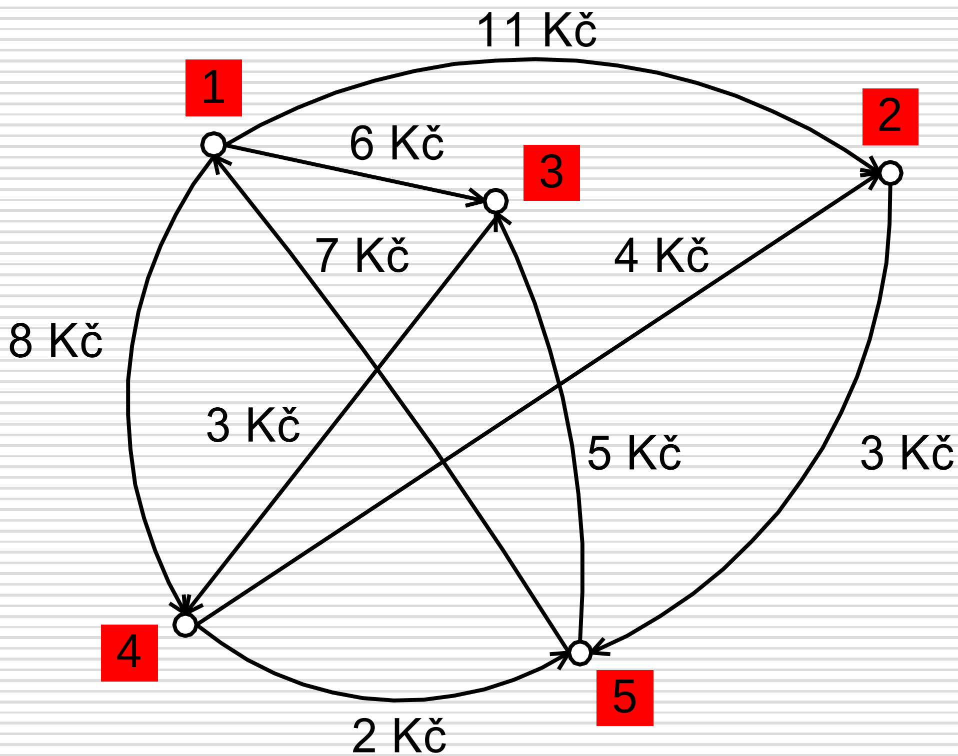
	1	2	3	4	5
1	0	11	6	8	∞
2	∞	0	∞	∞	3
3	∞	∞	0	3	∞
4	∞	4	∞	0	2
5	7	∞	5	∞	0

	1	2	3	4	5
1	0	11	6	8	∞
2	∞	0	∞	∞	3
3	∞	∞	0	3	∞
4	∞	4	∞	0	2
5	7	18	5	15	0

	1	2	3	4	5
1	0	11	6	8	14
2	∞	0	∞	∞	3
3	∞	∞	0	3	∞
4	∞	4	∞	0	2
5	7	18	5	15	0

MATICE VZDÁLENOSTÍ

Příklad: Minimální náklady na dopravu zboží



	1	2	3	4	5
1	0	11	6	8	14
2	∞	0	∞	∞	3
3	∞	∞	0	3	∞
4	∞	4	∞	0	2
5	7	18	5	8	0

	1	2	3	4	5
1	0	11	6	8	10
2	∞	0	∞	∞	3
3	∞	7	0	3	5
4	∞	4	∞	0	2
5	7	18	5	8	0

	1	2	3	4	5
1	0	11	6	8	10
2	10	0	8	11	3
3	12	7	0	3	5
4	9	4	7	0	2
5	7	18	5	8	0

Součty
35 Kč
32 Kč
27 Kč
22 Kč
38 Kč

Závěrem

- Použití grafů je názornou pomůckou při řešení složitých problémů.
- Složitá řešení se zpravidla již neobejdou bez použití výpočetní techniky.
- Byl to jen letmý úvod, grafy se dále zabývají toky, hledání cyklů, nejlevnějších tahů, úlohami o párování…