

8. ZÁKLADY TEÓRIE GRAFOV

Cieľ prednášky: objasniť základné pojmy teórie grafov, vedieť formulovať typické úlohy riešené prostredníctvom grafov (minimálna cesta, min/max kostra grafu, úloha obchodného cestujúceho, maximálny tok v sieťovom grafe) a spoznať podstatu ďalších úloh, ako napr. úloha čínskeho poštára, úloha o farbení grafu a pod.

Teória grafov (ale aj oblasť tzv. **sieťovej analýzy**) sú obvykle uvažované ako samostatné oblasti deterministicky zameraných postupov operačnej analýzy. Problémy, ktoré v rámci nich riešime už nie sú formulované pomocou ich matematických modelov, na aké sme boli zvyknutí pri matematickom programovaní (aj keď pre niektoré z nich by bolo takéto modely definovať možné).

Ukážeme si, ako je možné niektoré rozhodovacie a optimalizačné problémy riešiť na **modeli rozhodovacieho problému, vyjadrenom vo forme grafu**.

Graf vo všeobecnosti predstavuje určitú grafickú (vizuálnu) formu modelu rozhodovacieho alebo optimalizačného problému, ktorý skúmame.

Výhodou použitia grafov je hlavne prehľadná štruktúra problému, umožňujúca jeho grafické vyjadrenie, ktoré je pre rozhodovateľa väčšinou zrozumiteľnejšie, ako jeho formulácia v podobe matematického modelu. Využitie grafov je vhodné pre reprezentáciu reálneho systému, najčastejšie vo forme určitej „infraštruktúrnej siete“.

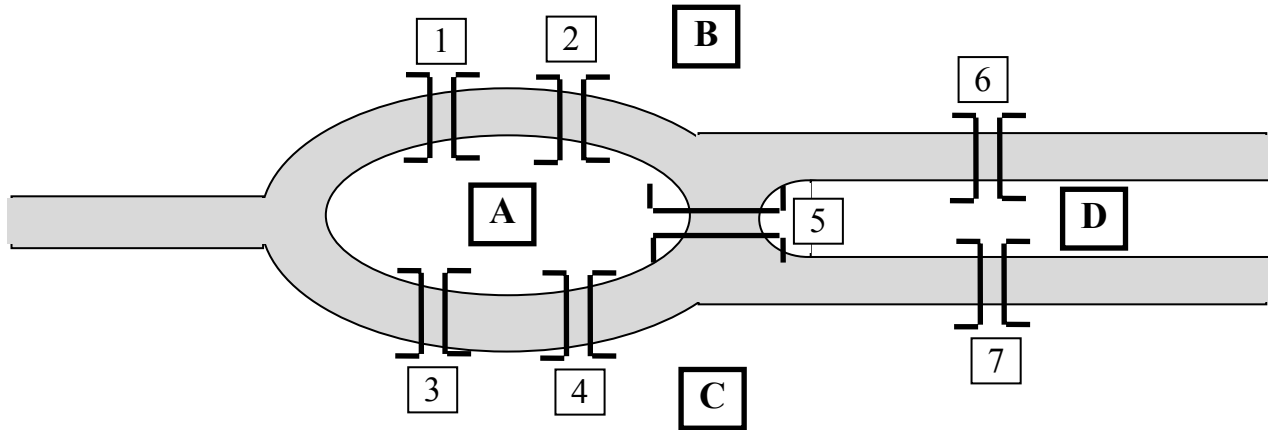
Grafické zobrazenie rôznych rozhodovacích problémov je obvykle veľmi názorné, či už ide o problematiku zobrazenia terénu a infraštruktúry umiestnenej v ňom, vyjadrenia štruktúry zložitých chemických organických zlúčenín alebo grafické zobrazenie elektrických obvodov. Všetky uvedené, aj keď principiálne odlišné, problémy môžeme riešiť v rámci oblasti nazývanej **teória grafov**.

Historické medzníky:

- 1736 - Euler, prvá zdokumentovaná úloha o grafoch, problém „o 7 královeckých mostoch“.
- 1747 - Kirchhoff, využitie grafov k zobrazeniu elektrických obvodov.
- 1857 - Cayley, využitie grafických štruktúr k znázorneniu organických chemických zlúčenín.
- 1859 - Hamilton, formulácia úloh typu cesta v grafe a definovanie tzv. úlohy „obchodného cestujúceho, problém minimálnych okružných jázd a ďalšie.
- 1926 - Borůvka, algoritmus pre určenie minimálnej kostry (minimálneho stromu) v grafe.

Pre ilustráciu uveďme vyššie zmienenú úlohu o 7 mostoch. Zadanie úlohy: rieka v meste (Kráľovec, Königsberg – Kaliningrad) (obr.1), rozdelila územie mesta na štyri časti (ozn. **A**, **B**, **C**, **D**), ktoré boli navzájom pospájané siedmymi mostmi (ozn. **1**, **2**, ..., **7**). Otázka znela, či je možné prejsť všetky mosty tak, aby sa cez každý most prešlo iba raz.

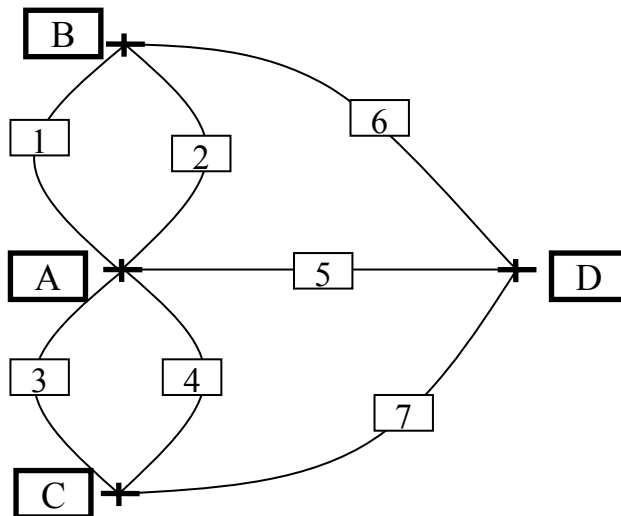
Ide o úlohu, ktoré radíme skôr medzi hlavolamy (tzv. *jednot'ažka*).



Obr.1a Sedem kráľoveckých mostov

Podstatné je, že Euler pri jej riešení ako prvý k jej znázorneniu použil **graf** (obr.1b), v ktorom definoval niektoré základné pojmy, ako vrcholy (uzly) grafu a hrany grafu.

Vidíme, že jednotlivé časti pevniny sú znázornené ako body A, B, C, D (vrcholy grafu) a cesty cez jednotlivé mosty označené čiarami (hrany grafu).



Obr.1b Graf úlohy „o siedmych královeckých mostoch“.

Poznámka: Pre úplnosť uvedme, že zadaná úloha nemá riešenie. To je možné dokázať vopred tak, že každému vrcholu priradíme jeho tzv. stupeň vrcholu rovný počtu z neho vystupujúcich hrán. Riešiteľné sú iba úlohy, ktoré majú všetky vrcholy rádu (t.j. počet vystupujúcich uzlov) párneho alebo práve dva vrcholy rádu nepárneho a všetky ostatné sú rádu párneho. Pre tento prípad to neplatí, pretože úloha má všetky vrcholy stupňa nepárneho (B, C, D stupňa tretieho, vrchol A stupňa piateho).

V teórii grafov najčastejšie používame dva základné pojmy a to:

- **vrchol (uzly) grafu** – všetky vrcholy v grafe tvoria množinu vrcholov V a
- **hrana grafu** – všetky hrany v grafe tvoria množinu hrán H .

K úplnému zadefinovaniu grafu ešte chýba vyjadrenie (predpis), ktoré vrcholy sú navzájom spojené a akými hranami.

Jednému z takýchto vyjadrení hovoríme **incidencia**, ktorá je vyjadrená buď tzv. **incidenčnou tabuľkou** (Tab.1), v ktorej pre jednotlivé hrany uvádzame, ktoré vrcholy tieto hrany spájajú alebo častejšie tzv. **incidenčnou maticou**, kde sú uvedené súvislosti vyjadrené jednoduchšie.

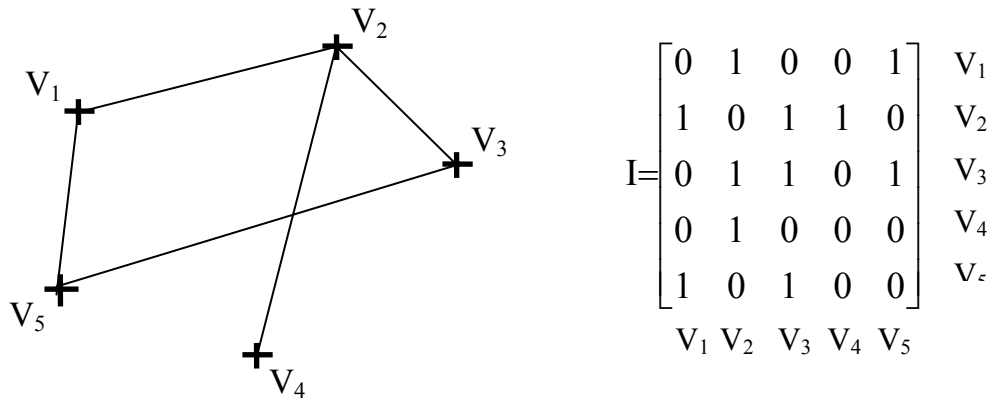
Tab.1 : Incidenčná tabuľka grafu z obr.1b.

Hrana	1	2	3	4	5	6	7
Vrcholy	A	B	A	C	A	B	C
	B	A	C	A	D	D	D

Takto zadáný graf môžeme zadefinovať ako usporiadanú trojicu $G \equiv (V, H, I)$, kde $V \equiv \{A, B, C, D\}$ je množina vrcholov, $H \equiv \{1, 2, 3, 4, 5, 6, 7\}$ je množina hrán a I je tzv. **incidencia**. Vrcholy označujeme symbolmi V_j (s jedným indexom $j=1,2,\dots,n$ - t.j počet vrcholov), hrany symbolmi h_{ij} (spája vrcholy V_i a V_j) a vzťahy medzi vrcholmi a hranami zapisujeme do **incidenčnej matice**.

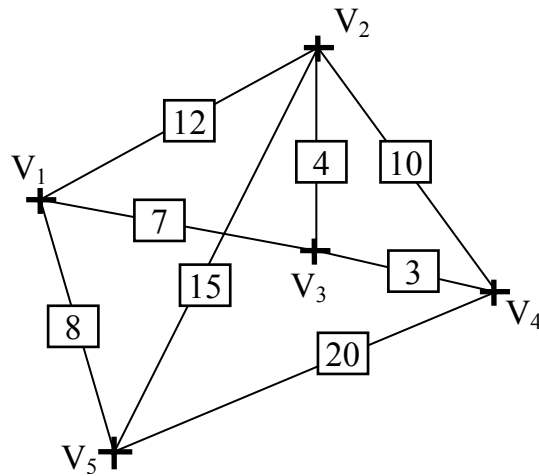
Najjednoduchšie môžeme vyjadriť incidenciu tzv. **neohodnoteného grafu**. Pokiaľ hrana medzi vrcholmi existuje - priradíme jej hodnotu 1, pokiaľ neexistuje priradíme 0. Príklad neohodnoteného grafu s jeho incidenčnou maticou je na Obr.2a.

Neohodnotené grafy často využívame, napr. v elektrotechnike – „1“ v incidenčnej matici znamená, že medzi 2 bodmi grafu (obvodu) existuje hrana (spoj).



Obr.2a Neohodnotený graf a jeho incidenčná matica

V praxi sa však častejšie vyskytujú tzv. **grafy ohodnotené**.



$$I = \begin{bmatrix} 0 & 12 & 7 & N & 8 \\ 12 & 0 & 4 & 10 & 15 \\ 7 & 4 & 0 & 3 & N \\ N & 10 & 3 & 0 & 20 \\ 8 & 15 & N & 20 & 0 \end{bmatrix} \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{matrix}$$

$V_1 \quad V_2 \quad V_3 \quad V_4 \quad V_5$

Obr.2b Ohodnotený graf a jeho incidenčná matica.

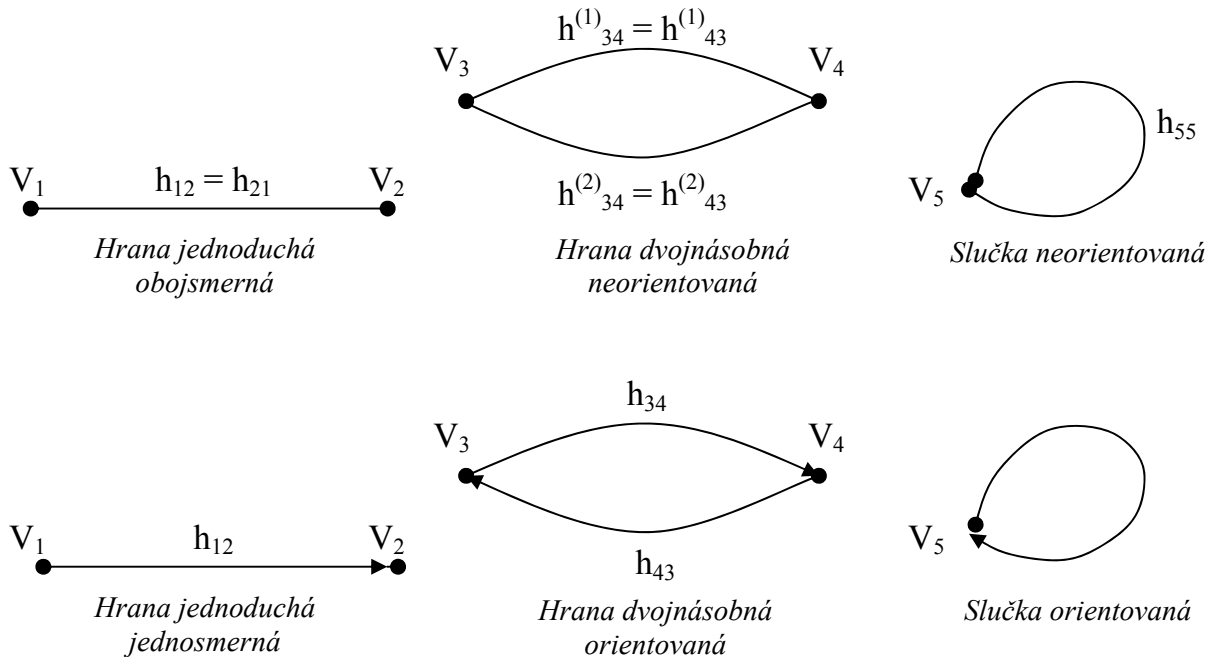
Hranám priradzujeme hodnoty s konkrétnym významom, napr. *vzdialenosť, náklady, stúpanie, priepustnosť, intenzita dopravy, spotreba PHM, čas jazdy* apod.

Incidenčná matica ohodnoteného grafu (obr.2b) svojimi prvkami vyjadruje priamo hodnoty priradené jednotlivým hranám. Pokiaľ hrana medzi vrcholmi neexistuje, do matice buď nepíšeme nič alebo (pre niektoré algoritmy) do incidence vkladáme veľmi veľkú hodnotu (symbolicky N).

Grafy na *Obr.2a*, aj *Obr.2b* uvažujeme ako **graf neorientovaný**. Je charakteristický tým, že pre hodnoty hrán platí: $h_{ij} = h_{ji}$, pre všetky i, j . Znamená to, že jeho incidenčná matica je symetrická okolo hlavnej diagonály. Takýto neorientovaný graf nazývame aj **multigraf**.

Graf, v ktorom platí: $h_{ij} \neq h_{ji}$, pre všetky i, j nazývame graf **orientovaný** (často aj ako **orgraf**) a jeho incidenčná matica bude určite nesymetrická.

Hrany podľa Obr.3 môžu byť: **jednoduché** (jednosmerné alebo obojsmerné), **násobné** (obojsmerné aj jednosmerné) alebo vytvárajú tzv. **cyklus (slučku)**. Slučka je zvláštny typ hrany, ktorý vychádza aj končí v tom istom vrchole.



Obr.3 Základné typy hrán v grafoch

8.1. Základná terminológia teórie grafov

Základnou štruktúrou v rámci teórie grafov je tzv. **graf**.

Grafom $G = (V, H)$ budeme rozumieť usporiadanú dvojicu množín V a H , kde:

$V = \{u_1, u_2, \dots, u_n\}$ je konečná, neprázdna množina tzv. **vrcholov grafu** G ,

$H = \{h_{ij}\}$ je množina tzv. **hrán grafu** G , pričom $h_{ij} = (u_i, u_j)$, kde $u_i, u_j \in V$.

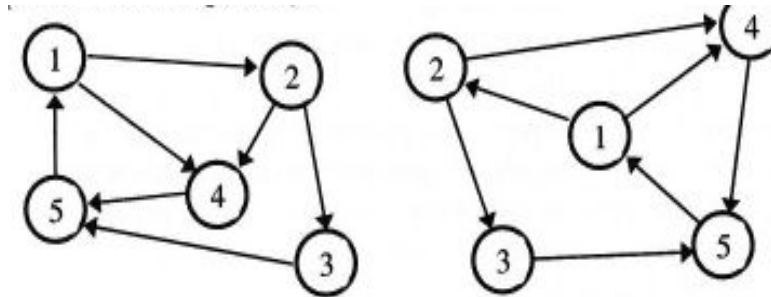
Pod pojmom **graf** preto rozumieme útvar, skladajúci sa z určitého množstva bodov, ktoré nazývame **vrcholy grafu** a množiny spojnic medzi vrcholmi, ktoré nazývame **hrany grafu**.

*Poznámka: V niektorých aplikáciách sa častejšie využívajú alternatívne pojmy **uzol** alebo **úsek**.*

Pri zakresľovaní grafu obvykle používame nasledujúci spôsob znázornenia:

- **vrcholy** – znázorňujeme ako kolieska, do ktorých obvykle vpisujeme index vrcholu,
- **hrany** - znázorňujeme priamymi čiarami alebo jednoduchými oblúkmi,
- **orhrany** - znázorňujeme pomocou šípok v zmysle orientácie hrany.

Graf je možné zakresliť rôznymi spôsobmi (*Obr. 4*), obvykle však volíme čo najprehľadnejšie zakreslenie, t.j. také zobrazenie, pri ktorom sa pokiaľ možno hrany nepretínajú.



Obr.4 Príklady možnosti dvojakého znázornenia toho istého orientovaného grafu

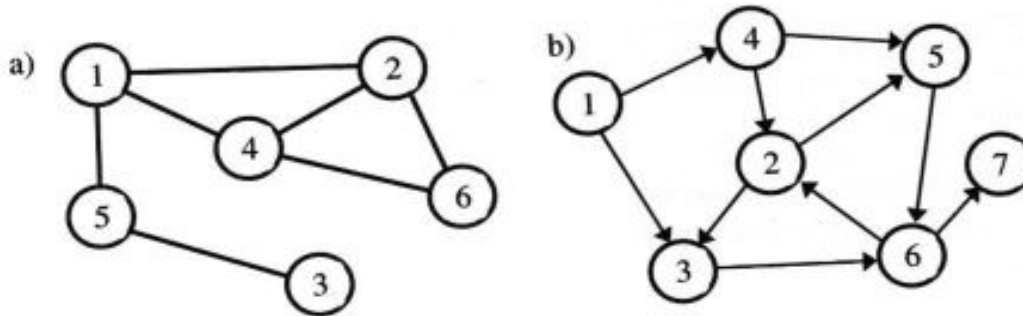
Platí: Graf je definovaný iba svojimi vrcholmi a hranami, nie spôsobom zakreslenia.

Každá hrana h_{ij} spája práve dva vrcholy grafu u_i a u_j a môžeme ju preto definovať pomocou dvojice vrcholov $h_{ij} = (u_i, u_j)$, ktoré spája. Často sa používa slovné spojenie, že hrana h_{ij} je **incidentná** s vrcholmi u_i a u_j . Susedné vrcholy sú potom každé dva vrcholy, spojené hranou.

V prípade, že hrany grafu $(u_i, u_j) \in H$ sú definované ako:

- **neusporiadané dvojice vrcholov** t.j. $(u_i, u_j) = (u_j, u_i)$ – t.j. nezáleží na tom, v akom poradí sú vrcholy, ktoré hranu definujú, uvedené – hovoríme o tzv. **neorientovanom grafe** = **MULTIGRAF** (napr. graf podľa Obr.5a).

- **usporiadané dvojice vrcholov** t.j. $(u_i, u_j) \neq (u_j, u_i)$ – t.j. záleží na tom, v akom poradí sú vrcholy, ktoré hranu definujú, uvedené – hovoríme o tzv. **orientovanom grafe** = **ORGRAF**. Orientované hrany v grafe znázorňujeme pomocou šípok (napr. Obr.5b). Ako prvý sa v usporiadanej dvojici (u_i, u_j) definujúcej hranu h_{ij} uvádza počiatočný vrchol a ako druhý potom koncový vrchol orientovanej hrany (orhrany) h_{ij} .



Obr.5 Príklad znázornenia: a) neorientovaný graf, b) orientovaný graf

Vo všeobecnosti je možný aj iný prístup. Keď vrcholy grafu reprezentujú činnosti a hrany grafu slúžia k znázorneniu väzieb medzi jednotlivými činnosťami. Doteraz uvažované prípady predstavovali **hranovú reprezentáciu jednotlivých činností**, v takomto prípade by sa jednalo o tzv. **vrcholovú reprezentáciu jednotlivých činností**.

Ak je každej hrane grafu priradená číselná hodnota ide o tzv. **hranovo ohodnotený graf**.

Takéto hodnoty ohodnotenia hrany môžu v závislosti na riešenom probléme vyjadrovať napr.

- **vzdialenosť** – dĺžková miera, napr. metre či kilometre v dopravnej infraštruktúre a pod.,
- **čas** – doba jazdy dopravného prostriedku po trase, ktorú daná hrana reprezentuje, doba trvania určitej činnosti reprezentovanej danou hranou a pod.,
- **náklady** - ekonomické vyjadrenie, napr. náklady pri realizácii činnosti reprezentovanej hranou,
- **kapacitu (priepustnosť)** – výkonnostné parametre hrany, obvykle priepustnosť danej hrany (napr. kapacita liniek v počítačovej sieti, počet aut, ktoré prejdú po istom úseku komunikácie za jednotku času, kapacita schodiska v evakuačnom modeli veľkej budovy a pod.).

Samozrejme, podľa riešenej aplikácie je možné použiť aj iné veličiny pre ohodnotenie hrán. Pokiaľ je to potrebné, každý neohodnotený graf môžeme považovať za graf ohodnotený a to tak, že všetkým hranám priradíme rovnakú hodnotu – obvykle hodnotu jedna.

V niektorých, skôr ale špecifických, aplikáciách je vhodnejšie priradiť ohodnotenie vrcholom grafu a vtedy hovoríme o **vrcholovo ohodnotenom grafe**.

8.2. Príklady reálnych systémov

Mnoho reálnych systémov môžeme znázorniť formou grafov. Graf môže znázorňovať napr. *distribučnú alebo dopravnú sieť*. Uzly v grafe môžu byť interpretované napr. ako *distribučné centrá*, hrany grafu ako *spojnice medzi nimi*. Takáto grafická reprezentácia je názorná a zrozumiteľná aj pre menej skúsených riešiteľov.

Príklady reálnych systémov modelovaných pomocou:

8.2.1. Neorientované grafy:

A1. dopravná sieť (cestná, železničná, letecká, počítačová infraštruktúra a pod.) kde:

- *vrchol (uzol)* - reprezentuje napr. križovatku, dopravňu, letisko, ale aj jednotlivú výhybku, počítačový server, terminál a pod.
- *hrana (úsek)* - reprezentuje cestu, úsek trate či jednotlivú koľaj, letecký spoj, kábel apod. (tie, ktoré sú vždy použiteľné v oboch smeroch)

A2. rádiokomunikačná sieť, kde:

- *vrchol* – reprezentuje jednotlivé rádiové či televízne vysielacie, uzly pre prenos signálu siete mobilných operátorov a pod.
- *hrana* – reprezentuje pomyselné spojnice medzi jednotlivými vysielacími či prenosovými uzlami (obvykle susednými).

8.2.2. Orientované grafy:

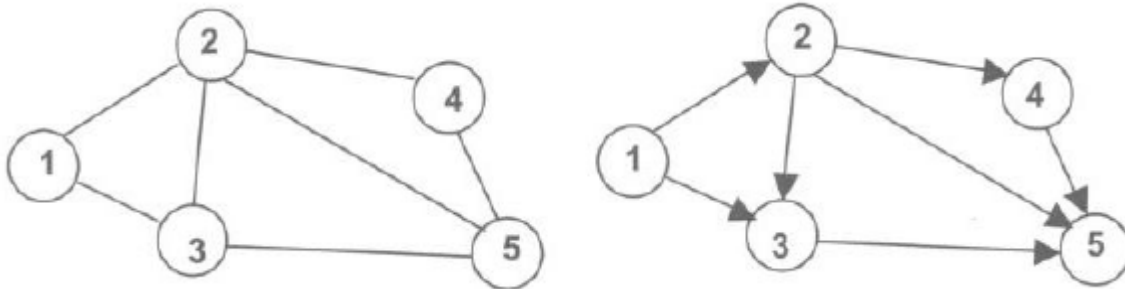
B1. dopravná sieť s jednosmernými komunikáciami

B2. siete modelujúce procesy prebiehajúce v čase (napr. sieťová analýza), kde:

- *hrana* - predstavuje jednotlivú modelovanú činnosť (východiskový vrchol hrany reprezentuje začiatok danej činnosti a koncový vrchol jej ukončenie),
- *vrchol* - reprezentuje začiatok a koniec jednotlivých činností.

8.3 Vymedzenie ďalších základných pojmov z teórie grafov

- **Graf** – útvar, ktorý je možné v rovine znázorniť pomocou bodov a spojnic medzi nimi. Takéto body budeme označovať ako **uzly grafu**, ozn. $u_1, u_2, \dots, u_n$, kde n je počet uzlov (vrcholov) grafu. Spojnice medzi jednotlivými uzlami nazývame **hrany grafu** a spojnicu (hranu) medzi uzlom u_i a u_j budeme označovať ako **hrana h_{ij}** .



Obr.6 Neorientovaný a orientovaný graf

Majme dva jednoduché grafy (Obr.6), obsahujúce 5 uzlov (u_1, u_2, u_3, u_4, u_5), ktoré sú navzájom prepojené 7 hranami ($h_{12}, h_{13}, h_{23}, h_{24}, h_{25}, h_{35}, h_{45}$). V prvom z nich je zmysel pohybu (tok, pohyb) uvažovaný po všetkých hranách v oboch smeroch, tzn. existuje možnosť spojenia napr. z uzla u_1 do u_2 , ale rovnako aj z uzla u_2 do u_1 .

Platí:

- **Neorientovaná hrana** - hrana umožňujúca obojsmerný pohyb medzi dvojicou uzlov, ktoré spája.
- **Neorientovaný graf** - graf, v ktorom sú iba neorientované hrany.

V druhom grafe (Obr.6 vpravo) je každej hrane jednoznačne priradený povolený smer pohybu, znázornený šípkami. Pohyb v ňom je teda uvažovaný iba v smere určenom šípkou, napr. iba z uzla u_1 do uzla u_3 , ale v smere opačnom už nie.

Platí:

- **Orientovaná hrana** - hrana umožňujúca iba zadaný smer pohybu medzi dvojicou uzlov, ktoré spája.
- **Orientovaný graf** - graf obsahujúci orientované hrany.

Cieľom riešenia väčšiny problémov v teórii grafov je určenie tzv. **sledu hrán s požadovanými atribútmi**.

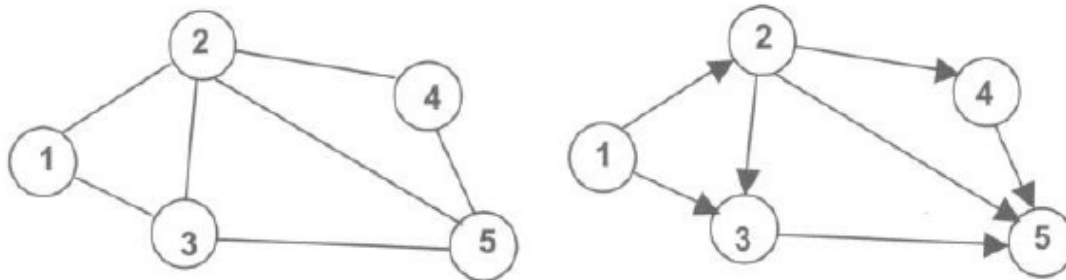
- **Sled v grafe** – postupnosť vrcholov a hrán začínajúca končiaca vo vrchole.

Najčastejšie sa jedná o nasledovné sledy hrán:

- **Cesta v grafe** – cesta medzi uzlom u_1 a u_j je postupnosť (sled) navzájom na seba nadväzujúcich hrán, ktorá začína v uzle u_i a končí v uzle u_j .

- **Orientovaná cesta v grafe** – cesta v orientovanom grafe, rešpektujúca povolenú orientáciu hrán.
- **Neorientovaná cesta v grafe** – cesta v orientovanom grafe, neuvažujúca s orientáciou hrán.

Napríklad, v neorientovanom grafe na *Obr.6* existuje medzi uzlom u_1 a u_5 niekoľko ciest, napr. $u_1-u_2-u_4-u_5$ alebo $u_1-u_3-u_2-u_5$. Ale v orientovanom grafe (*Obr.6* vpravo) je prvá cesta ($u_1-u_2-u_4-u_5$) súčasne **cestou orientovanou**, naopak druhá cesta ($u_1-u_3-u_2-u_5$) v tom istom grafe predstavuje **cestu neorientovanú** (t.j. pohyb z uzla u_3 do uzla u_2 nie je vzhľadom k orientácii hrany povolený).



Obr.6 Neorientovaný a orientovaný graf

Z hľadiska homogenity grafického vyjadrenia problému rozoznávame:

- **Súvislý graf** – graf, v ktorom medzi ľubovoľnou dvojicou uzlov existuje neorientovaná cesta.
- **Cyklus** – špeciálny typ cesty, ktorá začína a končí v rovnakom uzle (napr. v neorientovanom grafe podľa *Obr.6* vľavo je cyklom napr. cesta $u_1-u_2-u_3-u_1$).

V reálnych úlohách grafy predstavujú vždy reprezentáciu určitého reálneho systému. Každý prvok grafu, t.j. každý uzol a každá hrana, majú určitú reálnu interpretáciu. Napríklad: celý graf môže predstavovať **sieť distribúcie tovaru** – potom **uzly budú reálne miesta** (sklady, odberatelia, dodávatelia) a **hrany predstavujú cestné spojenie alebo iné väzby medzi týmito miestami**.

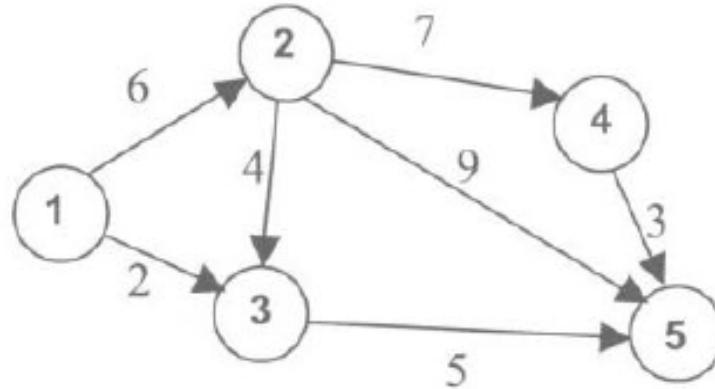
Každej hrane, ako aj každému uzlu môžeme priradiť jednu (príp. aj viac) charakteristiku, ktorú nazývame **ohodnotenie hrany / uzla**. Pri ohodnotení hrany sa u grafov najčastejšie jedná o dĺžku spojenia medzi jednotlivými prvkami distribučnej siete (napr. *m, km*).

Ale uzly a hrany môžu mať aj iné vyjadrenie. Napríklad, v oblasti **riadenia projektov** – tzv. **sieťová analýza**, hrany obvykle reprezentujú reálne činnosti a uzly predstavujú okamžiky začatia alebo ukončenia jednotlivých činností. Pokiaľ hrany grafu predstavujú činnosti (tzv. **sieťový graf**), ohodnotením najčastejšie býva: doba trvania činnosti, náklady na realizáciu činnosti alebo aj zdroje pre realizáciu činnosti (napr. nasadené pracovné kapacity na jednotlivé činnosti v rámci projektu).

V rámci všeobecnej teórie grafov sa preto využívajú aj špeciálne typy grafov, ako sú **sieťový graf** (sieť) alebo **strom**.

- **Sieťový graf** – graf, ktorý je súvislý, orientovaný, nezáporne hranovo (uzlovo) ohodnotený, acyklický (neobsahujúci cyklus) a obsahujúci dva špeciálne uzly – vstup a výstup.

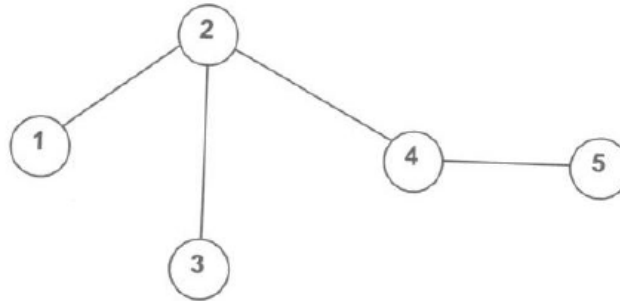
Vstupný uzol je uzol, z ktorého hrany iba vystupujú (žiadna do neho nevstupuje). Analogicky, výstupný uzol je taký uzol, z ktorého už žiadna hrana nevystupuje (hrany do neho iba vstupujú). Napríklad, v sieti podľa Obr. 7 je vstupným uzlom zrejme uzol u_1 a výstupným uzlom uzol u_5 .



Obr. 7 Sieťový graf

Ohodnotenie **cesty v sieti** býva najčastejšie realizované ako súčet ohodnotení hrán, ktoré cestu v sieti tvoria. Napr. ohodnotenie cesty $u_1-u_2-u_4-u_5$ v sieti podľa Obr. 7 sa rovná $6+7+3=16$

- **Strom** – je súvislý, neorientovaný graf, ktorý neobsahuje žiadny cyklus. Príklad stromu je na Obr. 8. Je zrejmé, že platí: **Medzi každými dvomi uzlami stromu existuje práve jedna cesta**



Obr.8 *Strom*

V každom prípade – či už sa jedná o graf alebo sieť – rozoznávame buď grafy:

- **hranovo ohodnotené** – graf v ktorom sú všetky hrany ohodnotené.
- **uzlovo ohodnotené** – graf v ktorom sú ohodnotené všetky uzly.

V teórii grafov sa však zaoberáme najčastejšie hranovo ohodnotenými grafmi, kde pre ohodnotenie hrany h_{ij} bude používaný symbol y_{ij} .

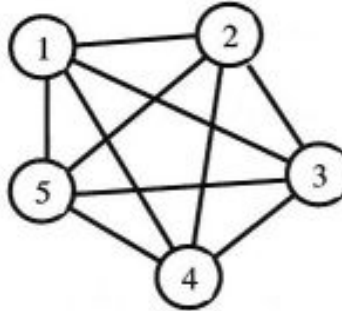
Prehľad najvýznamnejších základných pojmov v teórii grafov:

1. **Graf** – útvar skladajúci sa z konečnej neprázdnej množiny vrcholov a množiny hrán medzi týmito vrcholmi.
2. **Neorientovaný graf** – graf, kde hrany nemajú definovaný počiatok ani koniec, t.j. je možné sa po týchto hranách pohybovať v oboch smeroch.
3. **Orientovaný graf** - graf, kde každá hrana má definovaný počiatočný a koncový vrchol, t.j. umožňujú pohyb iba v jednom smere.
4. **Podgraf grafu** – graf tvorený niektorými alebo aj všetkými vrcholmi pôvodného grafu a len niektorými z hrán grafu pôvodného.
5. **Multigraf** – je špeciálny druh grafu v ktorom sú vrcholy u_i a u_j spojené nie jednou, ale viacerými paralelnými hranami.
6. **Acyklický graf** – orientovaný graf, ktorý neobsahuje cyklus. V každom konečnom acyklickom grafe existuje aspoň jeden vrchol vstupný a jeden vrchol výstupný. Platí tiež, že orientovaný graf s k vrcholmi je acyklický práve vtedy, keď je možné všetky jeho vrcholy očíslovať číslami $1, 2, 3, \dots, k$ tak, aby čísla vrcholov každej orhrany h_{ij} spĺňali vzťah $i < j$.
7. **Stupeň vrcholu** – počet hrán incidentných s daným vrcholom, t.j. počet hrán, ktoré vedú do zadaného vrcholu.

- 8. Sled v grafe** – postupnosť vrcholov a hrán začínajúca končiaca vo vrchole.
- 9. Ťah v grafe** – sled, v ktorom sa neopakuje žiadna hrana.
- 10. Cesta v grafe** – sled, v ktorom sa neopakuje žiadny vrchol.
- 11. Hamiltonovská cesta** – cesta v grafe, ktorá obsahuje všetky vrcholy grafu, ale nemusí obsahovať všetky hrany.
- 12. Eulerov sled** – sled obsahujúci všetky hrany grafu.
- 13. Eulerov ťah** – je taký Eulerov sled, v ktorom sa neopakuje žiadna hrana.
- 14. Súvislý graf** – graf, kde z každého vrcholu existuje cesta do každého ďalšieho vrcholu grafu.
- 15. Nesúvislý graf** – je graf, ktorý nie je súvislý. Graf sa javí ako „rozpadnutý“ na viac častí.
- 16. Úplný graf** – graf, v ktorom existuje hrana medzi každou dvojicou vrcholov grafu (*Obr.9a*).

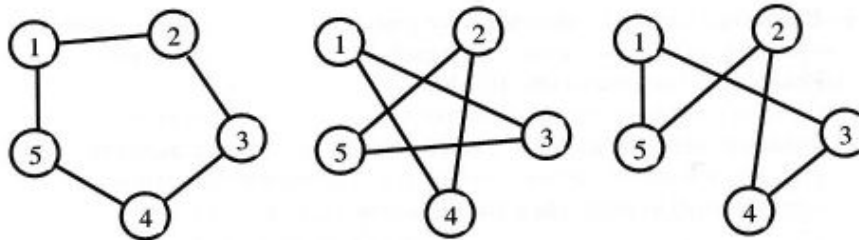
Počet hrán úplného neorientovaného grafu je $p = \binom{n}{2}$, kde n je počet vrcholov.

- 17. Kostra grafu** – podgraf, ktorý obsahuje všetky vrcholy pôvodného grafu a neobsahuje kružnicu, t.j. je stromom.
- 18. Vstupný vrchol** – každý vrchol orientovaného grafu, z ktorého hrany iba vystupujú.
- 19. Výstupný vrchol** – každý vrchol orientovaného grafu, do ktorého hrany iba vstupujú.



Obr.9a Úplný graf s piatimi vrcholmi

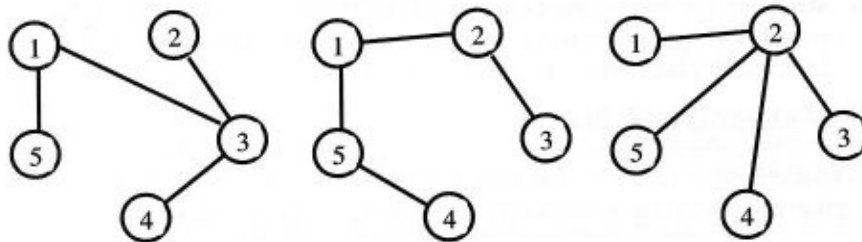
20. Kružnica – súvislý neorientovaný graf, kde z každého vrcholu vychádzajú práve dve hrany.



Obr.9b Rôzne kružnice s 5 vrcholmi (všetky sú podgrafom úplného grafu)

21. Cyklus – silne súvislý orientovaný graf, kde z každého vrcholu vychádza práve jedna orhrana a práve jedna orhrana do neho aj vstupuje.

22. Strom – je súvislý neorientovaný graf, v ktorom medzi dvoma vrcholmi existuje práve jedna cesta, t.j. neobsahuje kružnicu (Obr.9c).



Obr.9c Rôzne stromy s 5 vrcholmi (všetky sú podgrafom a aj kostrou úplného grafu 9b)

- 23. Sieť, sieťový graf** – orientovaný, ohodnotený, súvislý graf, ktorý neobsahuje cyklus a obsahuje práve jeden vstupný vrchol (z neho orhrany iba vychádzajú) a práve jeden výstupný vrchol (hrany do neho iba vstupujú).
- 24. Úloha čínskeho poštára** – nájdenie najkratšej trasy v grafe, ktorá obsahuje všetky hrany grafu.
- 25. Úloha obchodného cestujúceho** – úloha nájdenia najkratšej možnej uzatvorenej trasy v grafe, ktorá obsahuje všetky vrcholy grafu.
- 26. Úloha optimálneho umiestnenia depa** – úloha nájdenia vzdialenostne optimálneho umiestnenia miesta v grafe, ktoré môže slúžiť ako stredisko obsluhy pre ostatné vrcholy v grafe.

- 27. Úloha určenia minimálnej cesty** – úloha, v ktorej hľadáme najkratšiu cestu medzi dvomi zadanými vrcholmi v grafe.
- 28. Úloha určenia minimálnej (maximálnej) kostry grafu** – úloha, v ktorej hľadáme taký podgraf, ktorý obsahuje všetky vrcholy pôvodného grafu, je stromom a súčet ohodnotenia jeho hrán je minimálny (resp. maximálny).
- 29. Úloha určenia maximálneho toku v sieti** – úloha, v ktorej určujeme maximálny počet jednotiek, ktoré je možné pri rešpektovaní kapacít hrán v celej sieti prepraviť medzi dvomi zadanými vrcholmi.

8.4 Možnosti vyjadrenia grafov a ich vlastností, riešenie grafov

Pokiaľ nie je graf príliš veľký (najmä ak nemá príliš mnoho hrán), je asi najzrozumiteľnejším vyjadrením jeho grafické znázornenie – ako obrázok. Ak chceme pracovať s rozsiahlymi grafmi alebo ak chceme zadať graf do PC je nutné využiť iné spôsoby vyjadrenia grafov a súvislostí v nich.

Pravidlá pre zakresľovanie grafov:

- *uzly budú zakresľované ako krúžky s indexom príslušného uzla vo vnútri,*
- *hrany sú priame alebo lomené čiary spájajúce jednotlivé uzly,*
- *orientácia hrán je určená šípkou, zakreslenou u koncového uzla danej hrany,*
- *ohodnotenie hrany je vyjadrené číselnou hodnotou uvedenou pri hrane.*

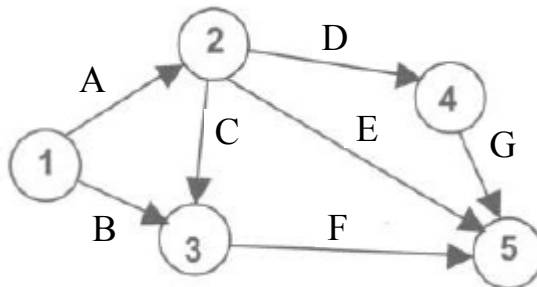
Možnosti popisu grafov a väzieb v nich

1. Zoznamy vrcholov a hrán

Množina vrcholov: popísaná jednoduchým vymenovaním (zoznamom) prvkov. Množina hrán: popísaná zoznamom trojíc tvorených menom hrany a jej počiatočným a koncovým vrcholom. Ide v podstate o úplný popis grafu podľa jeho definície. Neorientované grafy popisujeme formálne rovnakým spôsobom ako orientované, ale poradie vrcholov jednotlivých hrán volíme ľubovoľne.

Napríklad, graf podľa Obr.10 môžeme vyjadriť nasledovne:

- **Množina vrcholov** – $V = \{1, 2, 3, 4, 5\}$
- **Množina hrán** – $H = \{A, 1,2 ; B, 1,3; C, 2,3 ; D, 2,4 ; E, 2,5 ; F, 3,5 ; G, 4,5\}$



Obr.10 Orientovaný graf

2. Zoznam vrcholov a zoznamy okolia vrcholov

Ide o úspornejší variant predchádzajúceho spôsobu. Množina vrcholov je opäť popísaná zoznamom prvkov, ale hrany sú popisované po skupinách: pre každý vrchol X je vždy uvedený zoznam množiny hrán, ktoré v tomto vrchole začínajú. Každá hrana je potom popísaná iba svojim menom a koncovým vrcholom alebo počiatočný vrchol X je pre celú skupinu hrán spoločný.

- **Množina vrcholov** – $V = \{1, 2, 3, 4, 5\}$
- **Množina hrán** – $H = \{1-A,B ; 2-C,D,E ; 3-F ; 4-G\}$

3. Matica susednosti

Nech G je orientovaný graf. Ak zvolíme (ľubovoľne, ale pevne) poradie jeho vrcholov $u_1; u_2, \dots, u_n$, môžeme grafu G priradiť tzv. **maticu susednosti** $\mathbf{M}_G$ rádu n a to predpisom v tvare:

m_{ij} = počet hrán vedúcich z vrcholu i do vrcholu j .



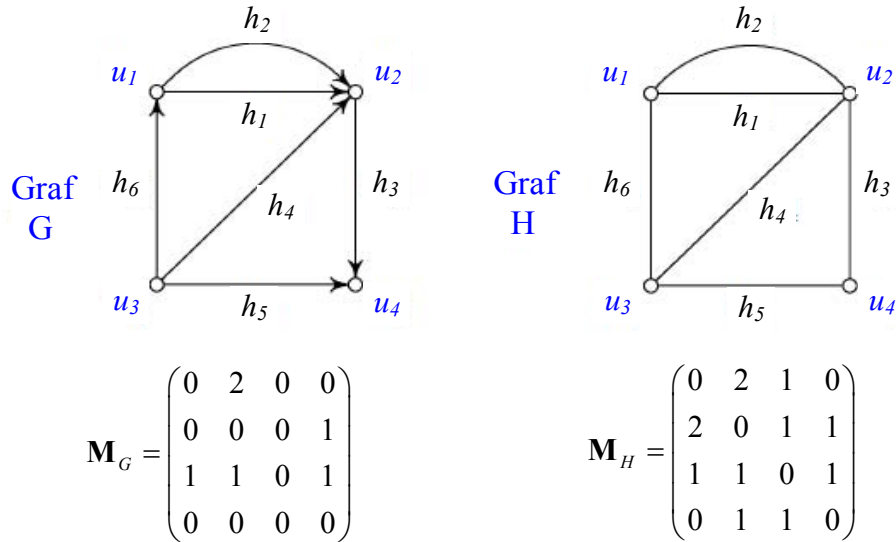
Obr.11 Orientovaný graf G a neorientovaný graf H

Pre neorientované grafy definujeme maticu susednosti $\mathbf{M}_G$ predpisom

m_{ij} = počet hrán spájajúcich vrcholy i a j .

Matica susednosti neorientovaného grafu je vždy plne symetrická: $m_{ij} = m_{ji}$ pre všetky i, j .

Pre graf G a graf H podľa Obr.11 potom dostaneme matice susednosti $\mathbf{M}_G$ a $\mathbf{M}_H$ v tvare:



Obr.11a Matice susednosti pre orientovaný graf G a neorientovaný graf H

4. Matica incidencie

Nech G je orientovaný graf bez cyklov (slučiek). Ak zvolíme (ľubovoľne, ale pevne) nielen poradie vrcholov $u_1; u_2, \dots, u_n$, ale aj poradie hrán $h_1; h_2, \dots, h_m$, môžeme grafu G priradiť tzv. **maticu incidencie B** (často aj ako **incidenčná matica**) typu (n,m) pravidlom v tvare:

$$b_{ij} = \begin{cases} 1, & \text{ak } u_i \text{ je počiatočným vrcholom hrany } h_j \\ -1, & \text{ak } u_i \text{ je koncovým vrcholom hrany } h_j \\ 0, & \text{v ostatných prípadoch} \end{cases}$$

Incidenčná matica grafu G z Obr.11a je $\mathbf{B}_G = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & -1 \\ -1 & -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 & -1 & 0 \end{pmatrix}$

Poznámka: každý stĺpec obsahuje práve jednu 1 a jednu -1 a súčet hodnôt v i -tom riadku je rovný rozdielu počtu hrán do uzla vstupujúcich a vystupujúcich, t.j. $d^+(u_i) - d^-(u_i)$.

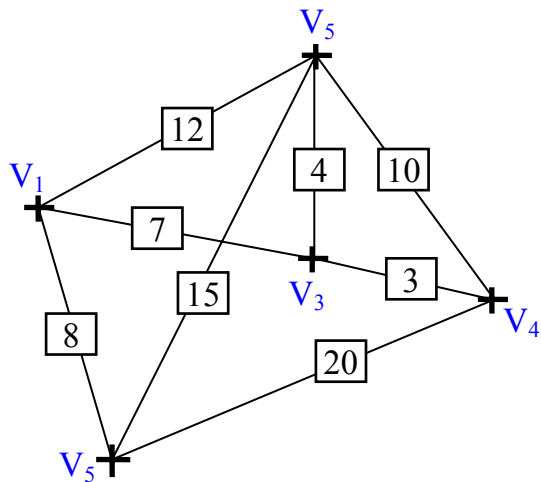
Pre neorientované grafy (bez slučiek) sa incidenčná matica definuje pravidlom

$$b_{ij} = \begin{cases} 1 & \text{ak } u_i \text{ je incidentný s hranou } h_j \\ 0 & \text{v ostatných prípadoch} \end{cases}$$

Incidenčná matica grafu H z Obr.11b je $\mathbf{B}_H = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$.

Poznámka: súčet hodnôt v i -tom riadku je rovný $d(v_i)$ a každý stĺpec obsahuje presne dve jednotky (počet uzlov tvoriacich hranu).

Najčastejšie sa však využíva zápis, kedy **incidenčná matica ohodnoteného grafu** svojimi prvkami priamo vyjadruje aj hodnoty priradené jednotlivým hranám. Napríklad graf na Obr.12.



$$\mathbf{B} = \begin{pmatrix} 0 & 12 & 7 & N & 8 \\ 12 & 0 & 4 & 10 & 15 \\ 7 & 4 & 0 & 3 & N \\ N & 10 & 3 & 0 & 20 \\ 8 & 15 & N & 20 & 0 \end{pmatrix} \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{matrix}$$

$V_1 \quad V_2 \quad V_3 \quad V_4 \quad V_5$

Obr.12 Ohodnotený graf a jeho incidenčná matica.

Poznámka: Pokiaľ hrana medzi vrcholmi neexistuje, do incidenčnej matice buď nenapišeme nič alebo (pre niektoré počítačové algoritmy) je vkladaná veľmi veľká hodnota (symbol N).

8.5 Algoritmy pre prehľadávanie grafov

Účelom tzv. **prehľadávania grafov** je zisťovanie dostupnosti, t.j. zisťovanie, do ktorých vrcholov grafu určované cesty zo zadaného východiskového vrcholu vedú a obvykle vedie aj k nájdenie takýchto ciest (MIN - najkratších resp. MAX – najdlhších).

Prehľadávať graf, zadaný graficky je možné aj „ručne“, ale ak je graf veľký a neprehľadný, je dobré poznať vhodný systematický postup, inak môžeme chybovať. Ak chceme graf prehľadávať automatizovane (PC) je systematický spôsob prehľadávania grafov úplne nevyhnutný.

Uvedieme 4 spôsoby prehľadávania (iba ich názvy a princíp spôsobu prehľadávania), ktoré sú vhodné a charakteristické hlavne pre hľadanie ciest v orientovaných grafoch.

1. Značkovanie vrcholov - vrcholom grafu priradujeme značky a ak má vrchol značku, znamená to, že do neho cesta z východiskového vrcholu r vedie.

- I. [Inicializácia] Označujeme východiskový vrchol r , ostatné vrcholy sú bez značiek.
- II. [Výber hrany] Vyberieme ľubovoľnú hranu h , ktorej počiatočný vrchol má značku a koncový vrchol nie. Ak taká hrana neexistuje, výpočet končí.
- III. [Značkovanie] Označujeme vrchol $K_u(h)$ a pokračujeme vo výpočte podľa kroku 2.

2. Zapamätávanie ciest, strom nájdených ciest – informácia o tom, že do vrcholu vedie bližšie neurčená cesta z východiskového vrcholu, obvykle nepostačuje. Chceme vedieť aj to, kadiaľ takáto cesta vedie. K tomu stačí vždy po označovaní vrcholu mu ešte navyše priradiť jednu hodnotu $Odkial'(K_u(h)) := h$, kde h je hrana vybraná v kroku 2 – tzv. **predchodca**. Takto doplnený algoritmus označuje všetky dostupné vrcholy a zároveň si zapamätá hodnotu prechádzajúceho vrcholu. Cestu do označovaného vrcholu $u \neq r$ je možné po skončení značkovacieho algoritmu ľahko nájsť spätným postupom pomocou hodnôt predchodcov.

Uvedený princíp tvorí aj teoretický základ jednej z metód, tzv. **Dijkstrovho algoritmu**.

3. Prehľadávanie grafu do hĺbky – je možné si predstaviť ako prieskum grafu cestovateľom, ktorý cestuje po hranách grafu a vracia sa dôsledne tou istou cestou, po ktorej prišiel. Princíp: postup riešenia v grafe, pri ktorom je v každom kroku z naposledy navštíveného uzla vybratá jedna hrana, ktorou postupujeme ďalej. Uzly a hrany, ktoré už boli spracované, vhodne označíme. Pokiaľ ďalšia vybraná hrana vedie do už označeného uzla, nie je v ďalšom kroku použitá. Pokiaľ z daného uzla už nevedie hrana k žiadnemu nenavštívenému uzlu, je potrebné sa o vrátiť o jeden krok späť a skúsiť inú neoznačenú hranu. Postup prehľadávania končí, ak sú už označené všetky uzly. Tento postup odpovedá hľadaniu cesty v bludisku pomocou tzv. Ariadneho nite. Je to teda dobrý návod, napr. ako sa chovať v bludisku a hľadať cestu von.

4. Prehľadávanie grafu do šírky – predstavme si to tak, ako keby graf súčasne preskúmaval veľký počet prieskumníkov, ktorí „po hladinách“ postupne „zaplavujú“ dostupnú časť grafu. Princíp: v každom kroku vyberá v navštívenom uzle všetky hrany, ktoré z neho vedú do uzlov zatiaľ neoznačených. Použité hrany aj novo navštívené uzly sú vhodne označené. Pre každý z novo označených uzlov sa potom vyberajú hrany vedúce do ešte nenavštívených uzlov. Postup prehľadávania končí, ak sú navštívené všetky uzly. Na prehľadávaní do šírky je pozoruhodné to, že vedie k nájdeniu najkratších ciest, t.j. ciest s najmenším počtom hrán.

8.6 Typické úlohy riešené v rámci teórie grafov

V rámci teórie grafov rozlišujeme tieto typické optimalizačné problémy:

1. **Optimálne spojenie miest = určenie minimálnej (maximálnej) kostry grafu.**
2. **Optimálne (najkratšie, minimálne) cesty v grafe = určenie minimálnej cesty.**
3. **Optimálne toky v grafoch = určenie maximálneho toku (priepustnosti) v sieti.**
4. **Ďalšie úlohy**, napr. úloha obchodného cestujúceho, úloha optimálneho umiestnenia depa iné.

8.6.1 Optimálne spojenie miest

Pod optimálnym spojením miest rozumieme riešenie úlohy určenia tzv. **minimálnej kostry grafu**. Ide o nájdenie takého podgrafu v pôvodnom grafe, ktorý: **bude obsahovať všetky uzly, tvoriť strom a zároveň bude mať minimálny súčet ohodnotení hrán, ktoré strom tvoria.**

Aplikácie: návrh spojenia medzi uzlami v danej lokalite tak, aby medzi každou dvojicou uzlov existovala nejaká cesta. Príklad: budovanie rozvodov telefónnych, počítačových, energetických a iných sietí v budove, areáli či meste; plán zimnej údržby komunikácií, apod. Ide o prípady, keď je potrebné zachovať dostupnosť všetkých vrcholov pri minimálnej súhrnnej dĺžke potrebných hrán. Účelom riešenia je, aby celková dĺžka spojenia (napr. potrubie, kábel a pod.) bola čo najkratšia.

Analýza problému:

Majme graf $G = (V, H)$ s ohodnotenými hranami. Úlohou je vybrať z množiny H takú podmnožinu hrán, aby:

- *z každého vrcholu existovala cesta do každého iného vrcholu,*
- *súčet ohodnotení vybraných hrán bol minimálny (resp. maximálny).*

Bolo by možné dokázať, že kostra grafu s n vrcholmi vždy obsahuje $(n-1)$ hrán.

Základný algoritmus pre hľadanie kostry grafu je jednoduchý. Sled hrán tvoriacich kostru grafu vyberáme tak, že postupne vyberáme hrany s najnižším (resp. najvyšším) ohodnotením tak, aby žiadna práve zaraďovaná hrana nevytvorila s ľubovoľnými, už vybranými, hranami kružnicu (cyklus). Po $n-1$ úspešných výberoch hrán je minimálna / maximálna kostra jednoznačne určená.

Algoritmus hľadania minimálnej kostry grafu:

1. *V celom grafe nájdeme 2 hrany s najnižším ohodnotením.*
2. *V každom ďalšom kroku vždy vyberieme ďalšiu hranu s minimálnym ohodnotením tak, aby netvorila cyklus s už skôr vybranými hranami.*
3. *Druhý krok opakujeme až do výberu celkového počtu $(n-1)$ hrán, ktoré budú tvoriť hľadanú minimálnu kostru grafu.*

Kostra (strom) súvislého grafu - taký súvislý podgraf, ktorý obsahuje všetky vrcholy grafu a neobsahuje pritom uzatvorený okruh.

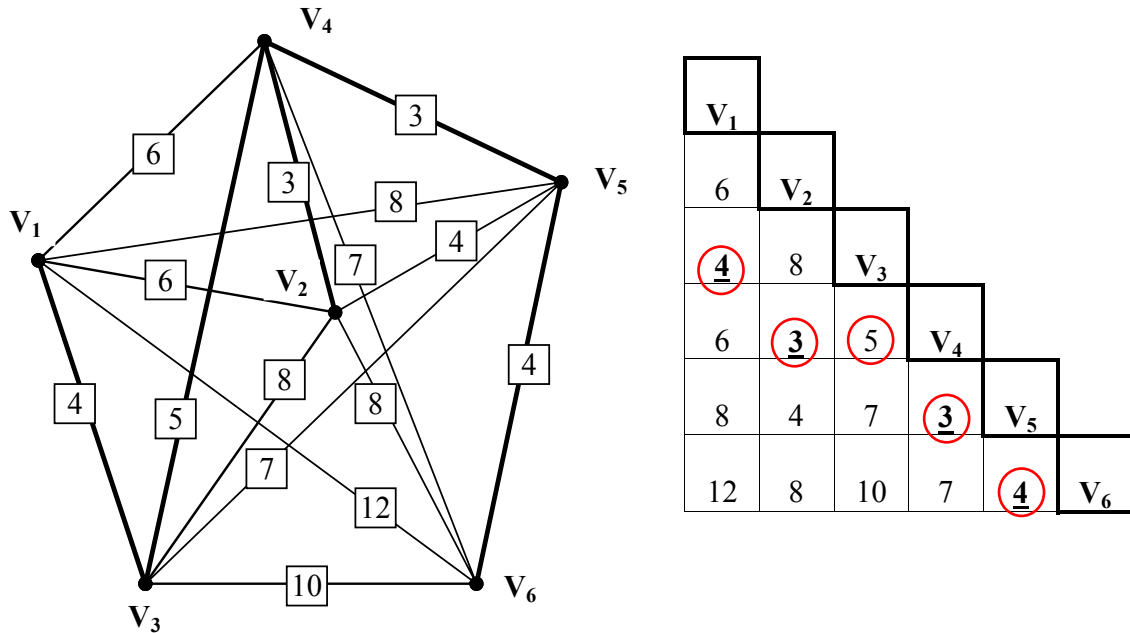
Minimálna kostra grafu – kostra grafu, v ktorej je súčet ohodnotení hrán minimálny.

Úloha: nájsť vzájomného prepojenia medzi všetkými vrcholmi tak, aby súčet ohodnotení hrán zaradených do tohto prepojenia bol minimálny. Zadaný neorientovaný súvislý graf, všeobecne s n vrcholmi a h hranami, musí však spĺňať podmienku

$$\left[(n-1) \leq h \leq \binom{n}{2} \right].$$

Príklad 1: Pre graf podľa obr.13 nájdite minimálnu kostru.

Riešenie:



Obr.13 Minimálna kostra grafu a jej incidenčná matica.

Komentár k riešeniu:

Počet vrcholov $u=6$, počet hrán $h=13$. Z hodnôt ocenenia hrán v grafe (incidenčnej matice) vyplýva, že dve hrany s minimálnym ohodnotením sú hrany $h_{24} = 3$ a $h_{45} = 3$. Stanú sa súčasťou minimálnej kostry.

Ďalšie hrany v poradí podľa veľkosti sú $h_{13} = 4$, $h_{25} = 4$ a $h_{56} = 4$, z ktorých je potrebné vyradiť hranu h_{25} , pretože s hranami h_{24} a h_{45} vytvára okruh. Do minimálnej kostry je naopak možné zaradiť hrany h_{13} a h_{56} .

Doteraz sme do minimálnej kostry zaradili 4 hrany. Nakoľko $(n-1) = 5$, je potrebné zaradiť do kostry grafu ešte jednu hranu, vyhovujúcu zadaným podmienkam. Takou môže byť hrana $h_{23} = 5$, ktorá je oproti doteraz zaradeným najbližšia väčšia a pritom s doteraz zaradenými hranami nevytvára okruh. V tomto momente je úloha vyriešená, tzn., dostávame:

Minimálnu kostru tvoria hrany (h_{13} , h_{24} , h_{34} , h_{45} a h_{56}) a jej hodnota je rovná súčtu hodnôt uvedených hrán, tzn. **MK = 4 + 3 + 5 + 3 + 4 = 19.**

Analogicky je možné určovať aj maximálnu kostru grafu, ak postup pri vyberaní hrán k zaradeniu do kostry sa obráti a hrany sa zoradia od maximálnej nestúpajúco.

Príklad na Optimálne spojenie miest, str.177.

8.6.2 Optimálne cesty v grafe

Nájdenie najkratšej cesty v grafe medzi dvomi určenými uzlami je jedna zo základných úloh teórie grafov. Úlohou je nájsť najkratšiu cestu v grafe = **určenie najkratšieho spojenia medzi zvolenou dvojicou miest**. Úlohu je možné riešiť v orientovanom aj neorientovanom grafe.

Napríklad: sieť komunikácií, kde uzly predstavujú miesta (križovatky), hrany predstavujú spojnice medzi týmito miestami a ohodnotenie hrán môže odpovedať vzdialenosti medzi miestami.

Aplikácie: jedna z najrozšírenejších aplikácií teórie grafov, napr. pre voľbu vhodnej trasy z miesta A do miesta B.

Analýza problému:

Majme graf $G=(V,H)$ s ohodnotenými hranami. V grafe sú určené dva rôzne vrcholy $u_i \in V$ a $u_j \in V$, $u_i \neq u_j$. Úlohou je nájsť minimálnu, t.j. najkratšiu, cestu z vrcholu u_i do vrcholu u_j .

Pokiaľ v danom grafe (dopravnej sieti) vyriešime úlohu určenia minimálnej cesty pre každú dvojicu vrcholov grafu, získame tzv. **maticu najkratších vzdialeností**. Príkladom môže byť tabuľka v takmer každom Autoatlase, kde je matica najkratších vzdialeností pre cestnú sieť, kde vrcholy tvoria vybrané väčšie mestá. Obvykle je znázornená iba v tzv. *trojuholníkovom tvare*, t.j. predpokladá sa jej symetrickosť a preto je zřejmé, že bola riešená na neorientovanom grafe.

Uvádza sa viac algoritmov pre optimálne cesty v grafe. Najčastejšie ide o určenie:

- ***Najkratšej vzdialenosti z daného vrcholu do vrcholu cieľového*** (obvykle získavame informácie o najkratších vzdialenostiach do všetkých ostatných vrcholov daného grafu. Inak povedané: ide o výpočet jedného riadku matice najkratších vzdialeností.
- ***Kompletnej matice vzdialeností***, t.j. údaje o najkratších vzdialenostiach z každého vrcholu do každého ďalšieho vrcholu daného grafu, t.j. medzi všetkými vrcholmi navzájom.

Najčastejšie typy úloh pre vyhľadanie najkratšej orientovanej cesty:

- **Z určeného východiskového vrcholu cez všetky ostatné a späť do vrcholu východiskového.**
- **Z určeného východiskového vrcholu do zadaného cieľového vrcholu.**
- **Z určeného východiskového vrcholu do každého ďalšieho vrcholu grafu.**
- **Z každého vrcholu grafu do zadaného cieľového vrcholu.**
- **Medzi všetkými možnými dvojicami vrcholov navzájom.**

A. Minimálna okružná cesta v grafe

Formulácia problému: prejsť z určeného vrcholu *úplného neorientovaného grafu* cez všetky jeho vrcholy a vrátiť sa do vrcholu počiatočného tak, aby prejdená dráha (t.j. súčet ohodnotení využitých hrán) bola minimálna. Takejto ceste hovoríme aj **Hamiltonovská kružnica**.

Zatiaľ nebola definovaná exaktná metóda, ktorá by zaručene poskytla optimálne riešenie. Úlohy malého rozsahu riešime ako kombinatorické problémy, napr. **metóda rozhodovacieho stromu** (preskúmať všetky možnosti ciest a vybrať cestu s minimálnou hodnotou) alebo využiť metódu „**Vetvenia a hraníc**“, ktorá zo všetkých kombinácií možných ciest vyberá iba perspektívne.

Vhodný algoritmus (nie vždy však zabezpečí optimálne riešenie, iba „dobré a prípustné“), je **metóda najbližšieho suseda**. Je vhodná aj pre riešenie rozsiahlych grafov alebo pre určenie východiskového riešenia pre metódu vetvenia a hraníc.

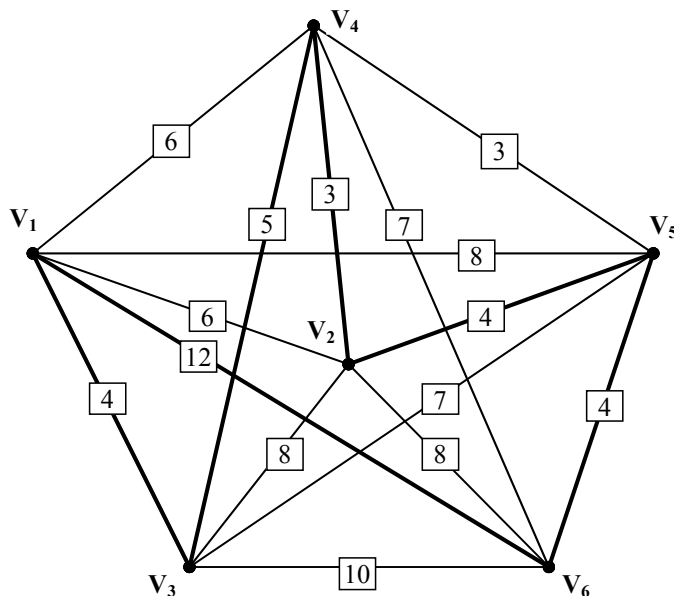
Slovné vyjadrenie praktického výpočtového postupu:

1. Zostaviť *incidenčnú maticu grafu* (ten musí byť *úplný*), inak nemožno zaistiť návrat do východiskového vrcholu (VV).
2. V riadku VV nájdeme minimálnu hodnotu hrany (najbližšieho suseda) a poznačíme si stĺpec (vrchol), v ktorom sa min. hodnota hrany nachádzala.

3. Prejdeme do riadku odpovedajúcemu určenému najbližšiemu vrcholu.
4. Pokračujeme krokom 2, pokiaľ takto neprejdeme všetky vrcholy. Musíme pritom kontrolovať, aby sme sa do žiadneho vrcholu nedostali dvakrát.
5. Po prejení všetkých vrcholov sa vrátíme (už jednou možnou cestou) do V_1 .

Príklad 2: Pre graf podľa obr.14 nájdite minimálnu okružnú cestu.

Riešenie:



Obr.14 Okružná cesta v grafe – zadanie grafu.

Za východiskový vrchol minimálnej cesty zvolíme napr. vrchol V_1 .

Tab.3 Okružná cesta v grafe – incidenčná matica úlohy.

Vrchol	1	2	3	4	5	6
1	0	6	<u>4</u>	6	8	12
2	6	0	8	3	<u>4</u>	8
3	4	8	0	<u>5</u>	7	10
4	6	<u>3</u>	5	0	3	7
5	8	4	7	3	0	<u>4</u>
6	<u>12</u>	8	10	7	4	0

Komentár k riešeniu:

Minimálna hrana v prvom riadku je $h_{13} = 4$, preto ju zaradíme do MOC a prejdeme do 3. riadku, odpovedajúcemu vrcholu V_3 . V riadku V_3 je minimálna hodnota hrany $h_{31} = 4$, ale túto hranu nie je možné využiť, pretože by sme sa vrátili do V_1 , ale ešte neboli prejdené všetky vrcholy.

Ďalšia najbližšia je preto hrana $h_{34} = 5$ (MOC pokračuje do vrcholu V_4). V riadku V_4 si okrem hodnoty v prvom stĺpci „zakážeme“ aj hodnotu v stĺpci tret'om a preto za najbližšieho suseda zvolíme V_2 , pretože $h_{24} = 3$. Takto pokračujeme ďalej a postupne do MOC zaradíme aj hrany $h_{25} = 4$, $h_{56} = 4$ a $h_{61} = 12$.

Výsledok: minimálna okružná cesta prechádza vrcholmi $V_1 - V_3 - V_4 - V_2 - V_5 - V_6 - V_1$ a jej hodnota (tvorená súčtom odpovedajúcich hrán) bude $z = 4 + 3 + 5 + 4 + 4 + 12 = 32$.

Záver: do MOC musela byť v poslednom kroku zaradená aj úplne najdlhšia hrana $h_{61} = 12$, tzn. určená MOC s veľkou pravdepodobnosťou nie je tým najlepším – optimálnym – riešením.

Úloha MOC je formálne podobná matematickému modelu priradovacej úlohy. Ak v každom riadku a stĺpci vyberieme jednu hodnotu (hranu) a budeme akceptovať aj podmienky určujúce okružnú cestu potom podstatou riešenia môže byť aj matica „redukovaných sadzieb“ (hodnôt hrán), ktorú dokážeme vytvoriť už postupom známym z priradovacej úlohy. Takto dostaneme *Tab.4*.

Tab.4 Matica redukovaných sadzieb pre okružnú cestu.

Vrchol	1	2	3	4	5	6	
1	-	2	0	2	4	(8) 7	$r = 2$
2	3	-	5	0	1	(5) 4	$r = 1$
3	0	4	-	1	3	(6) 5	$r = 1$
4	3	0	2	-	0	(4) 3	$r = 2$
5	5	1	4	0	-	(1) 0	$r = 1$
6	8	4	6	3	0	-	$r = 3$
	$r = 3$	$r = 1$	$r = 2$	$r = 1$	$r = 1$	$r = 3$	

Po riadkovej redukcii sa nepodarilo získať nulovú redukovanú sadzbu v šiestom stĺpci. V *Tab.4* je už nulová sadzba v každom riadku aj stĺpci. Ďalším krokom je nájdenie takého sledu hrán, ktorý bude vyhovovať podmienkam okružnej úlohy a zároveň bude obsahovať čo najviac hrán s nulovým redukovaným ohodnotením alebo ohodnotením najbližším vyšším.

Ak obdobne ako u Vogelovej aproximačnej metódy (VAM), určíme aj rozdiely v riadkoch a stĺpcoch (čísla *r* vedľa tabuľky) zistíme, že prednostne by sme mali využiť hrany h_{65} a h_{31} (pretože ich nevyužitie by zmenilo hodnotu účelovej funkcie „nepriaznivým smerom“ najviac). Máme teda pokryté cesty medzi vrcholmi $V_6 - V_5$ a $V_3 - V_1$, to sú pre nás tzv. ***perspektívne hrany***. Do vrcholu V_6 (ak vylúčime h_{56}) je možné sa dostať z vrcholu V_2 alebo V_4 a do vrcholu V_3 z vrcholu V_4 .

To nám už poskytuje prípustný sled vrcholov v tvare

$$V_1 - V_2 - V_6 - V_5 - V_4 - V_3 - V_1,$$

ktorý súčasne vyhovuje určeným podmienkam pre okružnú cestu. Hodnota účelovej funkcie (po dosadení hodnôt hrán z pôvodnej incidenčnej matice) je už minimálna a platí

$$z = h_{12} + h_{26} + h_{65} + h_{54} + h_{43} + h_{31} = 6 + 8 + 4 + 3 + 5 + 4 = 30$$

Poznámka: Podobnými kombináciami by sme v tomto prípade mohli získať aj ďalšie varianty minimálnej okružnej cesty (napr. $V_1 - V_3 - V_4 - V_6 - V_5 - V_2 - V_1$) s rovnakou hodnotou ÚF.

Pre úlohy menších rozmerov je možné nájsť optimálne riešenie okružnej cesty v grafe aj metódou **Vetvenia a hraníc**, aplikáciou ktorej preskúmavame viaceré (ale iba tzv. perspektívne) kombinácie sledov hrán. Za optimálne riešenie potom zvolíme sled hrán tvoriaci okružnú cestu s najnižšou hodnotou účelovej funkcie.

[Príklad na najkratšiu cestu v grafe: str.173](#)

B. Minimálna cesta v grafe

Vyhľadanie minimálnych ciest v grafoch patrí medzi najvýznamnejšie úlohy v TG. Minimálne cesty v grafe (MC) je pritom možné určovať v grafoch orientovaných, ako aj neorientovaných.

Najčastejšie sa jedná o minimálne cesty:

- *Z jedného definovaného vrcholu grafu do ďalšieho určeného vrcholu grafu.*
- *Z jedného určeného vrcholu grafu do všetkých ostatných vrcholov grafu.*
- *Medzi všetkými vrcholmi grafu navzájom.*

V literatúre je uvádzaných viac algoritmov s rôznou efektívnosťou a s rôznymi obmedzeniami. Od základnej verzie **metódy Forda a Fulkersona**, určenej pre orientované grafy a vyžadujúcej špeciálne číslovanie vrcholov, cez jej **Dantzigovu modifikáciu**, vhodnú aj pre neorientované grafy, až po **metódu Floydovu**, určenú pre hľadanie minimálnych ciest medzi všetkými vrcholmi grafu navzájom.

My ako základnú metódu pre určenie minimálnych ciest (MC) medzi jedným vopred určeným východiskovým vrcholom a všetkými ostatnými vrcholmi grafu (príp. medzi dvoma definovanými vrcholmi) uvedieme algoritmus **Dijkstrovej metódy** a ako vhodný postup pre hľadanie ciest medzi všetkými vrcholmi v grafe navzájom uvedieme tzv. **metódu sčítania v matici**.

Dijkstrova metóda

Uvedený algoritmus hľadania MC v grafe je založený na tvrdení:

Ak existuje MC medzi vrcholmi V_p a V_k a vrchol V_j na nej leží, potom minimálna cesta medzi vrcholmi V_p a V_j je súčasťou minimálnej cesty medzi vrcholmi V_p a V_k .

Princíp riešenia: Každému vrcholu v jednotlivých krokoch riešenia priradíme 2 hodnoty a to: **MC** = *hodnota minimálnej cesty* a **ST** = *stav vrcholu* (môže byť iba trvalý alebo dočasný).

V každom kroku riešenia potom vždy jeden vrchol určíme za tzv. *vrchol pracovný* V_{pr} .

Algoritmus hľadania minimálnej cesty v grafe:

1. [Iniciácia premenných]: určenému počiatočnému vrcholu cesty priradíme hodnotu $MC_p = 0$ a stav trvalý ($ST_p = T$). Znamená to, že zvolený počiatočný vrchol je už súčasťou minimálnej cesty. Ostatným vrcholom priradíme hodnotu $MC_j = N (\infty)$ a stav dočasný ($ST_j = D$). Počiatočný vrchol zvolíme za vrchol pracovný $V_p = V_{pr}$.

2. [Minimálna cesta]: Ak je pre ľubovoľný vrchol minulé hodnoty MC_j väčšie ako súčet minimálnej cesty vrcholu pracovného MC_{pr} a hrany $h_{pr,j}$ (t.j. z vrcholu V_{pr} do vrcholu V_j) priradíme do hodnoty MC_j hodnotu súčtu $MC_{pr} + h_{pr,j}$. Znamená to: ak je MC_j väčšie ako $MC_{pr} + h_{pr,j}$, potom

$$MC_j = MC_{pr} + h_{pr,j}.$$

V inom prípade ponecháme pôvodnú hodnotu MC_j .

Takto aktualizujeme hodnotu MC pre všetky vrcholy a nájdeme vrchol V_j , pre ktorý je uvedený súčet minimálny

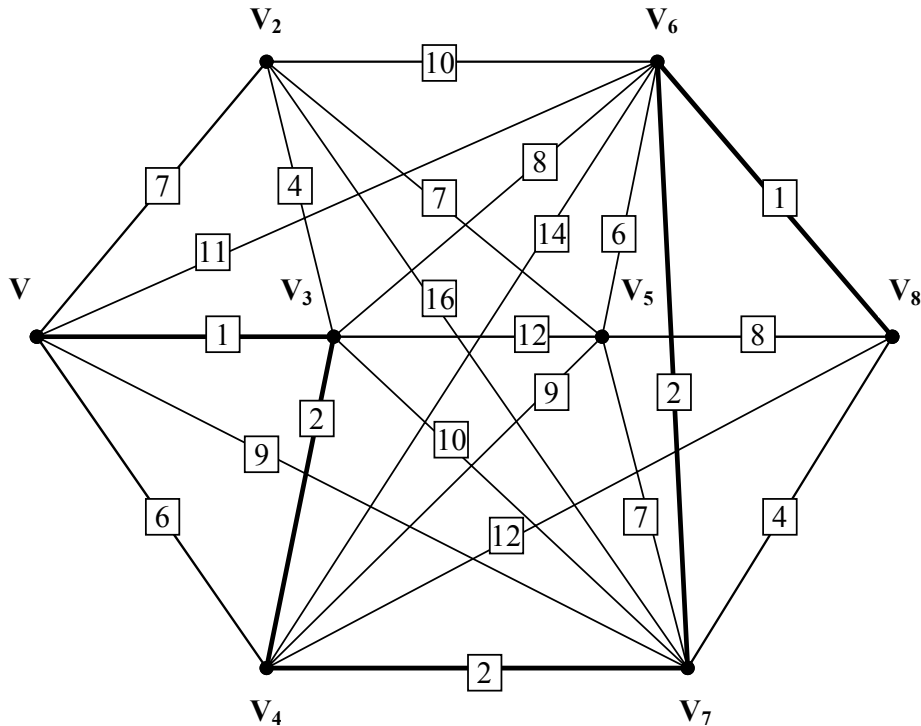
$$MC_{pr} + h_{pr,j} \rightarrow MIN.$$

Tomuto vrcholu určíme status trvalý a zvolíme ho za nový pracovný vrchol. Určená hodnota MC_j pre tento vrchol už obsahuje hodnotu minimálnej cesty z vrcholu počiatočného do vrcholu j .

3. [Opakovanie výpočtu]: Iteračný postup opakujeme, pokiaľ určený koncový vrchol V_k (pri min. ceste medzi dvoma vrcholmi) alebo všetky vrcholy grafu (pri min. ceste z určeného vrcholu do všetkých ostatných vrcholov) nenadobudnú status „trvalý“. V takom prípade už vo vektore MC budú uložené minimálne cesty pre všetky (alebo niektoré) vrcholy grafu.

Príklad 3: Dijkstrovou metódou určte minimálnu cestu z V_1 do V_8 pre graf zadáný na Obr.15.

Riešenie:



Obr.15 Minimálna cesta v grafe.

Tab.5 Incidenčná matica grafu z obr.15

V₁	7	1	6	N	11	9	N
	V₂	4	N	7	10	16	N
		V₃	2	12	8	10	N
			V₄	9	14	2	12
				V₅	6	7	8
					V₆	2	1
						V₇	4
							V₈

Podľa uvedeného postupu bude inicializácia premenných podľa *tab.6a*.

Tab.6a Minimálna cesta v grafe Dijkstrovou metódou – inicializácia.

Krok	Vrchol	1	2	3	4	5	6	7	8
0.	MC	<u>0</u>	N	N	N	N	N	N	N
	Stav	<u>T</u>	D	D	D	D	D	D	D

$$V_{pr} = V_1$$

Ak ku hodnotám $MC_{pr}=MC_1=0$ pripočítame hodnoty existujúcich hrán z vrcholu V_1 dostaneme nový iteračný krok, s hodnotami podľa *tab.6b*.

Tab.6b Minimálna cesta v grafe Dijkstrovou metódou – prvý krok riešenia.

Krok	Vrchol	1	2	3	4	5	6	7	8
1.	MC	0	7 ¹	1 ¹	6 ¹	N	11 ¹	9 ¹	N
	Stav	T	D	T	D	D	D	D	D

$$V_{pr} = V_3$$

Minimálna hodnota súčtu $MC_j + h_{pr,j}$ je v treťom stĺpci, preto bude $MC_3 = 1$, vrchol V_3 získa status „trvalý“ a zvolíme ho za vrchol pracovný ($V_{pr} = V_3$).

Poznámka: V pravom hornom rohu vektora MC sú čísla vrcholov, z ktorého sme MIN cestu získali.

Rovnako postupujeme ďalej. Výsledky po jednotlivých krokoch sú súhrnne uvedené v tab.6c.

Tab.6c Dijkstrova metóda - ďalšie kroky riešenia.

Krok	Vrchol	1	2	3	4	5	6	7	8	V_{pr}
2.	MC	0	5 ³	1 ¹	3 ³	13 ³	9 ³	9 ¹	N	V_4
	Stav	T	D	T	T	D	D	D	D	
3.	MC	0	5 ³	1 ¹	3 ³	12 ⁴	9 ³	5 ⁴	15 ⁴	V_2
	Stav	T	T	T	T	D	D	D	D	

4.	MC	0	5 ³	1 ¹	3 ³	12 ⁴	9 ³	<u>5</u> ⁴	15 ⁴	V₇
	Stav	T	T	T	T	D	D	<u>T</u>	D	
5.	MC	0	5 ³	1 ¹	3 ³	12 ⁴	<u>7</u> ⁷	5 ⁴	9 ⁴	V₆
	Stav	T	T	T	T	D	<u>T</u>	T	D	
6.	MC	0	5 ³	1 ¹	3 ³	12 ⁴	7 ⁷	5 ⁴	<u>8</u> ⁶	V₈
	Stav	T	T	T	T	D	T	T	<u>T</u>	

Pre zadanú úlohu (nájsť minimálnu cestu medzi vrcholmi V_1 a V_8) by bolo možné riešenie už ukončiť, pretože vrchol V_8 získal status „T-trvalý“.

Siedmy krok vykonáme iba preto, aby sme získali aj minimálnu cestu do vrcholu V_5 a získame tak maticu všetkých MC z východiskového vrcholu V_1 do všetkých ďalších vrcholov grafu.

7.	MC	0	5 ³	1 ¹	3 ³	<u>12</u> ⁴	7 ⁷	5 ⁴	8 ⁶	
	Stav	T	T	T	T	<u>T</u>	T	T	T	

Hodnoty minimálnych ciest do príslušných vrcholov sú uložené vo vektore MC.

Identifikáciu sledu vrcholov v minimálnych cestách: Pre určenie vrcholov, cez ktoré určená MC vedie je vhodné pri každom kroku riešenia ukladať počiatočné vrcholy hrán, z ktorých príslušné minimálne súčty vznikali (vid'. *malé čísla v pravom hornom rohu políčok MC*). Späťne potom môžeme určiť, že do vrcholu V_8 viedla cesta z vrcholu V_6 , do neho z vrcholu V_7 atď. postupne cez vrcholy V_4 a V_3 až do počiatočného vrcholu V_1 .

Hľadaným riešením je minimálna cesta $V_1 - V_3 - V_4 - V_7 - V_6 - V_8$ s hodnotou rovnou súčtu sledu príslušných hrán z minimálnej cesty a dostaneme $MC_{\min} = h_{13} + h_{34} + h_{47} + h_{76} + h_{68} = 8$.

Metóda sčítania v matici

Na obdobnom princípe pracuje aj algoritmus tzv. **metódy sčítania v incidenčnej matici**. Metóda je vhodná pre určovanie MC v grafe medzi všetkými vrcholmi v grafe navzájom. Je to opäť metóda iteračná a jej postup je nasledovný:

1. Pre určenie počtu potrebných iteračných krokov pre dosiahnutie optimálneho riešenia je nutné vyjadriť súvis medzi počtom hrán („ph“) MC a počtom iterácií („pi“) pomocou vzťahu

$$ph \leq n - 1 \leq 2^{pi},$$

kde n je počet vrcholov grafu.

2. Za východiskovú (multú) iteráciu považujeme úplnú incidenčnú maticu úlohy.
3. Na miesto každého nového prvku matice $c_{ij}^{(p)}$, pre $p=1,2,\dots,p_i$ zapíšeme min. hodnotu súčtov ($c_{ik} + c_{kj}$) pre $k=1,2,\dots,n$, ale iba ak je jeho hodnota je **menšia** ako pôvodný prvok c_{ij} , t.j. ak platí vzťah

$$c_{ij}^{(p)} = \min_k (c_{ik} + c_{kj})$$

tzn. ak $c_{ij}^{(p)}$ je menšie ako $c_{ij}^{(p-1)}$.

Príklad 4: Metódou sčítania v matici nájdite všetky MC v zadanom grafe medzi všetkými vrcholmi navzájom. Graf je zadaný incidenčnou maticou podľa Tab.7.

Riešenie:

Tab.7. Incidenčná matica grafu z obr.15

V₁	7	1	6	N	11	9	N
	V₂	4	N	7	10	16	N
		V₃	2	12	8	10	N
			V₄	9	14	2	12
				V₅	6	7	8
					V₆	2	1
						V₇	4
							V₈

Potrebný počet iterácií pre získanie optimálneho riešenia:

$$2^3 > 7 \Rightarrow \mathbf{pi = 3.}$$

Poznámka: Podobne ako u Dijkstrovej metódy, v pravých horných rohoch políček incidenčnej matice sú postupne zaznamenávané vrcholy, cez ktoré príslušná minimálna cesta vedie.

Vyššie uvedeným postupom (podobný postupu pri násobení matic) je možné určiť všetky hodnoty incidenčnej matice nielen pre 0.iteráciu, ale aj pre všetky ďalšie **pi = 3** iterácie.

Tab.7a Minimálna cesta v grafe metódou sčítania v matici – 0.iterácia.

V	1	2	3	4	5	6	7	8
1	0	7	1	6	N	11	9	N
2	7	0	4	N	7	10	16	N
3	1	4	0	2	12	8	10	N
4	6	N	2	0	9	14	2	12
5	N	7	12	9	0	6	7	8
6	11	10	8	14	6	0	2	1
7	9	16	10	2	7	2	0	4
8	N	N	N	12	8	1	4	0

Tab.7b Minimálna cesta v grafe metódou sčítania v matici – 1.iterácia.

Vrchol	1	2	3	4	5	6	7	8
1	0	5 ³	1	3 ³	13 ³	9 ³	8 ⁴	12 ⁶
2	5 ³	0	4	6 ³	7	10	12 ⁶	11 ⁶
3	1	4	0	2	11 ²	8	4 ⁴	9 ⁶
4	3 ³	6 ³	2	0	9	4 ⁷	2	6 ⁷
5	13 ³	7	11 ²	9	0	6	7	7 ⁶
6	9 ³	10	8	4 ⁷	6	0	2	1
7	8 ⁴	12 ⁶	4 ⁴	2	7	2	0	3 ⁶
8	12 ⁶	11 ⁶	9 ⁶	6 ⁷	7 ⁶	1	3 ⁶	0

Pri východiskovej matici 0.iterácie neexistovali cesty medzi 5 x 2 vrcholmi navzájom, ale už po 1. iterácii máme vzájomné cesty medzi všetkými vrcholmi grafu. **Zatiaľ však nie sú optimálne.**

Tab.7c Minimálna cesta v grafe metódou sčítania v matici – 2.iterácia.

V	1	2	3	4	5	6	7	8
1	0	5 ³	1	3 ³	12 ^{2, 3}	7 ^{3, 4, 7}	5 ^{3, 4}	9 ^{3, 4, 7}
2	5 ³	0	4	6 ³	7	10	8 ^{4, 3}	11 ⁶
3	1	4	0	2	11 ²	6 ^{4, 7}	4 ⁴	7 ^{4, 7, 6}
4	3 ³	6 ³	2	0	9	4 ⁷	2	5 ^{7, 6}
5	12 ^{2, 3}	7	11 ²	9	0	6	7	7 ⁶
6	7 ^{3, 4}	10	6 ^{4, 7}	4 ⁷	6	0	2	1
7	5 ^{3, 4}	8 ^{3, 4}	4 ⁴	2	7	2	0	3 ⁶
8	9 ^{3, 4, 7}	11 ⁶	7 ^{4, 7, 6}	5 ^{7, 6}	7 ⁶	1	3 ⁶	0

Tab.7d Minimálna cesta v grafe metódou sčítania v matici – 3.iterácia.

V	1	2	3	4	5	6	7	8
1	0	5 ³	1	3 ³	12 ^{3,4}	7 ^{3,4,7}	5 ^{3,4}	8 ^{3,4,7,6}
2	5 ³	0	4	6 ³	7	10	8 ^{3,4}	11 ⁶
3	1	4	0	2	11 ²	6 ^{4,7}	4 ⁴	7 ^{4,7,6}
4	3 ³	6 ³	2	0	9	4 ⁷	2	5 ^{7,6}
5	12 ^{3,4}	7	11 ²	9	0	6	7	7 ⁶
6	7 ^{3,4,7}	10	6 ^{4,7}	4 ⁷	6	0	2	1
7	5 ^{3,4}	8 ^{3,4}	4 ⁴	2	7	2	0	3 ⁶
8	8 ^{3,4,7,6}	11 ⁶	7 ^{4,7,6}	5 ^{7,6}	7 ⁶	1	3 ⁶	0

Pri porovnaní incidenčných matic po 2. a 3.iterácii vidíme, že pri 3.iterácii sa zmenila už iba hodnota MC medzi (podľa indexov) „najvzdialenejšími“ vrcholmi V_1 a V_8 .

Všeobecný matematický model úlohy minimálnej cesty – ako model úlohy LP

Uvažujme všeobecne zadanú sieť, tvorenú množinou vrcholov V , množinou hrán H a incidenciou definovanou incidenčnou maticou I ohodnoteného grafu. Úlohu nájdenia minimálnej cesty z počiatočného vrcholu V_p do zvoleného vrcholu koncového V_k môžeme zapísať aj matematickým modelom, odpovedajúcim podstate úloh celočíselného lineárneho programovania.

Ku každej hrane h_{ij} priradíme bivalentnú premennú x_{ij} , ktorá nadobudne hodnotu „1“ **ak príslušná hrana bude súčasťou hľadanej (min / max) cesty alebo hodnotu „0“, ak hrana súčasťou hľadanej cesty v grafe nebude.** Symbolom $V(i)$ označme množinu uzlov, ktoré sú s uzlom V_i spojené hranou (napr. v našom príklade je $V(1) = \{2, 3, 4, 6, 7\}$, $V(2) = \{1, 3, 5, 6, 7\}$ atď.).

Pomocou premenných x_{ij} je potom možné účelovú funkciu zapísať v tvare

$$z = 0,5 \times \left[\sum_{i=1}^n \sum_{j \in V(i)} h_{ij} \cdot x_{ij} \right] \rightarrow \text{MIN},$$

kde koeficient 0,5 znamená, že každú hranu v účelovej funkcii zarátame dvakrát. Ďalej je potrebné určiť ako dosiahnuť, aby hľadané riešenie, určené premennými x_{ij} , odpovedalo ceste v grafe.

Je zrejmé, že okrem počiatočného uzla V_p a uzla koncového V_k , je každý uzol ležiaci na hľadanej min / max ceste incidentný (t.j. susedí) s dvoma hranami, uzol počiatočný a koncový s jednou hranou.

Pre počiatočný a koncový vrchol musí preto platiť

$$\sum_{j \in V(i)} x_{pj} = 1 \quad , \quad \sum_{j \in V(k)} x_{kj} = 1 ,$$

a pre ostatné uzly

$$\sum_{j \in V(i)} x_{ij} = 2 \cdot y_i \quad i = 1, 2 \dots n ,$$

kde y_i je rovnako ako premenná x_{ij} **bivalentná premenná** ($y_i \in (0;1)$) nadobúdajúca hodnotu „1“, ak uzol na ceste leží alebo hodnotu „0“ v prípade, ak súčasťou hľadanej MC nie je.

Pre zadaný Príklad 4 môžeme matematický model napísať v tvare :

Účelová funkcia

$$z = h_{12} \cdot x_{12} + h_{13} \cdot x_{13} + h_{14} \cdot x_{14} + h_{15} \cdot x_{15} + \dots + h_{78} \cdot x_{78} \rightarrow \text{MIN}$$

Obmedzujúce podmienky

$$x_{12} + x_{13} + x_{14} + x_{16} + x_{17} = 1$$

$$x_{48} + x_{58} + x_{68} + x_{78} = 1$$

$$x_{21} + x_{23} + x_{25} + x_{26} + x_{27} - 2 \cdot y_2 = 1$$

$$x_{31} + x_{32} + x_{34} + x_{35} + x_{36} + x_{37} - 2 \cdot y_3 = 1$$

$$\vdots \qquad \qquad \qquad \vdots$$

$$x_{71} + x_{72} + x_{73} + x_{74} + x_{75} + x_{76} + x_{78} - 2 \cdot y_7 = 1$$

$$x_{ij}, y_i \in (0, 1) \qquad \text{pre } i = 1, 2 \dots 8, j \in V(i)$$

Je opäť jasné, že „ručné“ riešenie takéhoto matematického modelu neprichádza prakticky do úvahy, pretože nutnosť zaviesť 8 umelých premenných (+ 23 premenných v modeli), k riešeniu je nutné využiť dvojfázový simplexový algoritmus a následne realizovať ešte celočíselné riešenie je manuálne, ale aj prostredníctvom softvérových nástrojov náročné.

8.6.3 Optimálne toky v grafoch

Uvažujme sieťový graf, kde východiskový uzol u_1 bude predstavovať tzv. **zdroj**, ktorý produkuje nejaké jednotky prepravovaného substrátu a výstupný uzol u_n je cieľovým miestom, tzv. **ústie**, do ktorého sa majú vstupujúce jednotky prepraviť. Ohodnotenie jednotlivých hrán siete predstavuje **maximálnu priepustnosť tejto hrany**, t.j. maximálnu hodnotu toku (počet jednotiek), ktorý je možné za jednotku času po konkrétnej hrane poslať. Cieľ riešenia: zistiť celkovú maximálnu kapacitu (priepustnosť) siete, t.j. koľko jednotiek je možné za jednotku času prepraviť zo zdroja do ústia tak, aby boli rešpektované kapacity siete obmedzené priepustnosťou hrán.

Aplikácie: Zaistenie maximálnej priepustnosti dopravnej siete, kde vrcholy môžu predstavovať dopravné uzly, hrany spojenie medzi nimi a ohodnotenie hrany môže predstavovať maximálny počet vozidiel, ktoré je možné medzi dvomi uzlami prepraviť. Napríklad, pre určenie priepustnosti dopravného uzla (napr. mestská aglomerácia z hľadiska diaľkovej dopravy, zriaďovacie stanice v železničnej doprave, vnútorná štruktúra prekladiska a iné).

Je zrejmé, že na podobnom princípe sa bude jednať aj o prepravu komodity v produktovodoch, analýzu priepustnej výkonnosti elektrickej distribučnej siete, telekomunikačných sietí (napr. kapacita internetového pripojenia) a pod.

Analýza problému:

Majme sieťový graf $G = (V, H)$ s ohodnotenými hranami. Graf má jeden vstupný vrchol u_1 (zdroj), výstupný vrchol u_n (ústie) a ohodnotenie hrán h_{ij} (v daných jednotkách) udáva maximálnu kapacitu tzv. priepustnosť hrany. **Úloha:** určiť maximálny počet jednotiek, ktoré je možné v celej sieti prepraviť zo zdroja do ústia.

Teoretické východiská pre riešenie optimálnej priepustnosti siete:

Kapacitu hrany h_{ij} označme k_{ij} a x_{ij} je objem toku po tejto hrane. Objem všetkých tokov, vychádzajúcich z uzla u_i je $\sum x_{ij}$ a objem všetkých tokov do uzla u_i prichádzajúcich je $\sum x_{ji}$. Okrem uzlov u_1 a u_n musí platiť, že súčet všetkých tokov do uzla u_i vstupujúcich sa musí rovnať súčtu všetkých tokov, ktoré z tohto uzla vychádzajú a teda platí

$$\sum_{j=1}^n x_{ij} = \sum_{j=1}^n x_{ji} \quad \text{pre } i \neq 1, i \neq n.$$

- Objem všetkých tokov, ktoré opúšťajú uzol u_1 je možné zapísať výrazom $\sum_{j=1}^n x_{1j} - \sum_{j=1}^n x_{j1}$.
- Objem všetkých tokov, prichádzajúcich do uzla u_n zapíšeme výrazom $\sum_{j=1}^n x_{jn} - \sum_{j=1}^n x_{nj}$.

Celý problém optimálneho toku je možné formulovať aj ako úlohu lineárneho programovania.

Všeobecný model úlohy určenia maximálneho toku v sieti – ako model úlohy LP

maximalizovať

$$z = \sum_{j=1}^n x_{1j} - \sum_{j=1}^n x_{j1} \rightarrow \text{MAX}$$

za podmienok

$$\sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = 0, \quad \text{pre } i \neq 1, i \neq n$$
$$0 \leq x_{ij} \leq k_{ij}, \quad \text{pre } i, j = 1, 2, \dots, n.$$

Pojem **tok**, ktorý budeme používať je abstrakciou bežne zaužívaných fyzikálnych tokov (tok kvapaliny, elektrický prúd, prúd vozidiel, tok materiálu apod.).

Podnetom pre riešenie optimálnych tokov v sieťach boli najmä problémy s kapacitami v rámci technológií prepravy produktovodmi alebo pohybu dopravného prúdu po dopravnej infraštruktúre s určitou kapacitou pre zaistenie požadovanej dopravnej obslužnosti.

Z pohľadu riešených úloh môže mať takýto rozhodovací problém tiež viac variantov, napr.:

- *maximálny / minimálny tok medzi dvoma vrcholmi,*
- *maximálny / minimálny tok medzi viacerými vrcholmi,*
- *maximálny / minimálny najlacnejší tok medzi dvoma alebo viacerými vrcholmi,*
- *maximálny / minimálny viac produktový tok atď.*

Z uvedených problémov je základnou úloha určenia **maximálneho toku v sieti** medzi dvoma určenými vrcholmi. Uvedieme si iba jednu z viacerých možností riešenia.

Základné východiskové predpoklady pre riešenie:

- rieši sa preprava jedného homogénneho produktu,
- v sieti je vždy určený jeden vrchol počiatkový - **zdroj** a jeden vrchol koncový - **ústie**,
- hrany v grafe sú obvykle jednosmerné a ich ohodnotenie h_{ij} znamená kapacitu hrany (v niektorých úlohách sa môže vyskytovať horná kapacita hrany hk_{ij} a dolná kapacita hrany dk_{ij}),
- žiadny tok po hrane nemôže v žiadnom prípade prekročiť jej kapacitu,
- ak prepravované množstvo po hrane x_{ij} je rovné kapacite hrany h_{ij} , tok na hrane je **nasýtený**,
- vždy platí tzv. **podmienka zachovania toku**: Čo do vrcholu pritečie, musí z neho aj odtiecť.
- predpokladáme, že v sieti nedochádza k žiadnym stratám prepravovaného produktu.

S uvažovaním uvedených predpokladov môžeme sformulovať matematický model úlohy (iba zjednodušený, nakoľko detailný model by bol podobný modelu minimálnej cesty) a platí :

$$z = \sum x_{ij} \Rightarrow \text{MAX}$$
$$0 \leq x_{ij} \leq h_{ij}$$

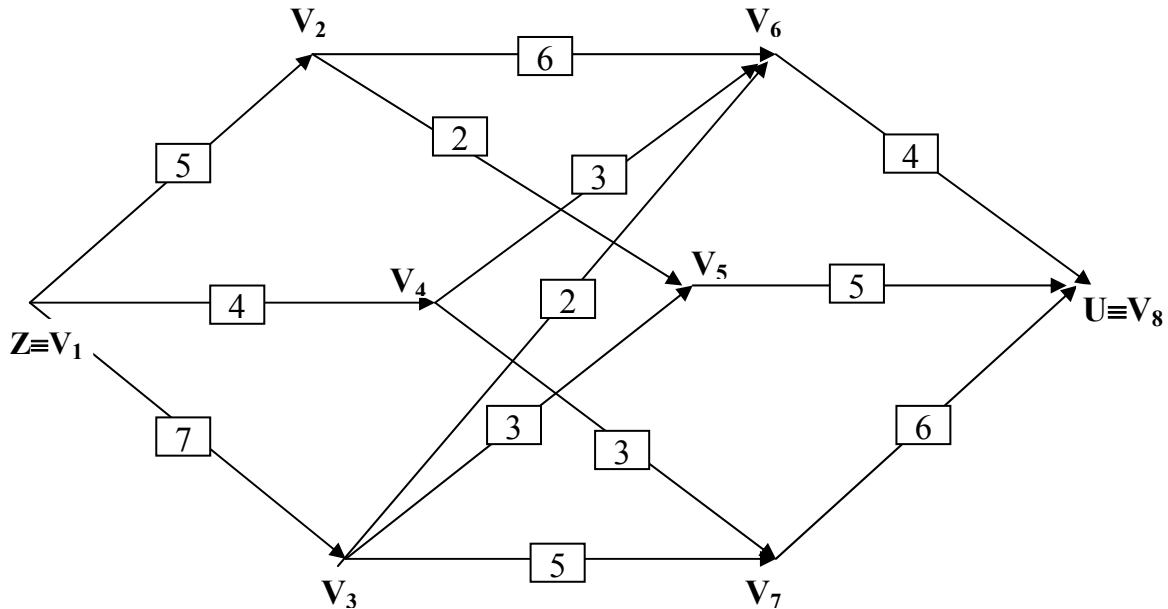
Riešenie predpokladá, že zo zdroja do ústia vedie obvykle viac ciest. Jeden z princípov riešenia spočíva v určení tzv. rezu grafom. Platí, že ak z každej takejto cesty grafom vyberieme jednu hranu dostaneme tzv. **rez grafom R** a súčet tokov na týchto hranách nazývame **kapacita rezu R(x)**.

Pre maximálny tok v grafe potom musí platiť veta (Ford a Fulkerson) podľa ktorej platí:
Hodnota maximálneho toku na grafe sa rovná kapacite minimálneho rezu.

Algoritmus určenia maximálneho toku v rovinnej sieti :

1. Zostrojíme najvyššie položenú dráhu $D^{(0)}(z, n)$ zo zdroja do ústia.
2. Na dráhe $D^{(0)}(z, n)$ určíme hranu s minimálnou kapacitou $h_{min.}$, ktorú z grafu odstránime a kapacity ostatných hrán dráhy znížime o $h_{min.}$
3. Ak naďalej existuje ďalšia dráha $D^{(1)}(z, n)$ pokračuje sa bodom 1.
4. Ak ďalšia dráha $D(z, n)$ už neexistuje (graf už nie je sieťou), riešenie končí a maximálny tok sa rovná súčtu kapacít odstránených hrán.

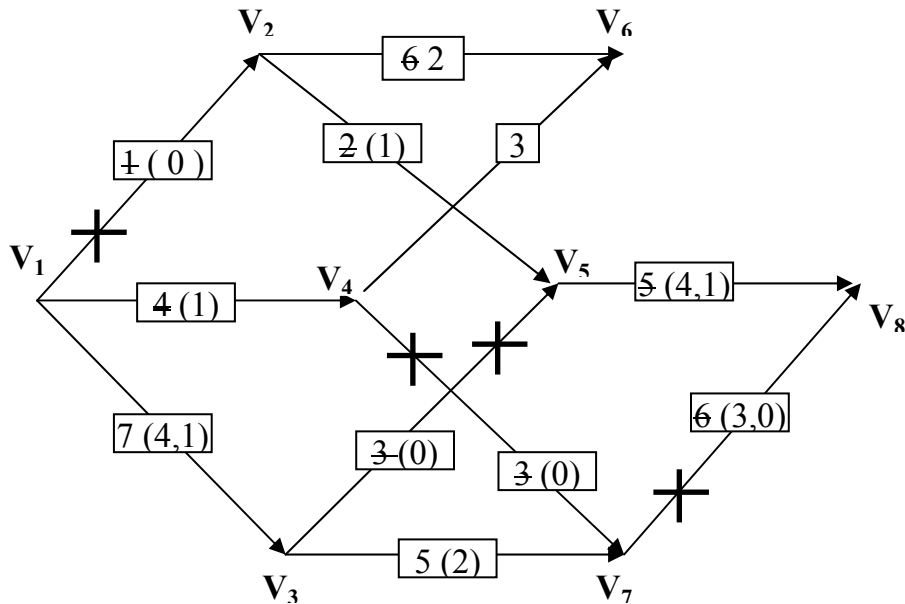
Príklad 5: V rovinnom grafe zadanom podľa Obr.16 určte maximálny možný tok, ktorý je obmedzený kapacitou jednotlivých hrán grafu. Zdroj je vrchol V_1 , ústie je vrchol V_8 a kapacity jednotlivých hrán sú určené hodnotami k_{ij} , definovanými pri jednotlivých hranách.



Obr.15 Maximálny tok v sieti - zadanie

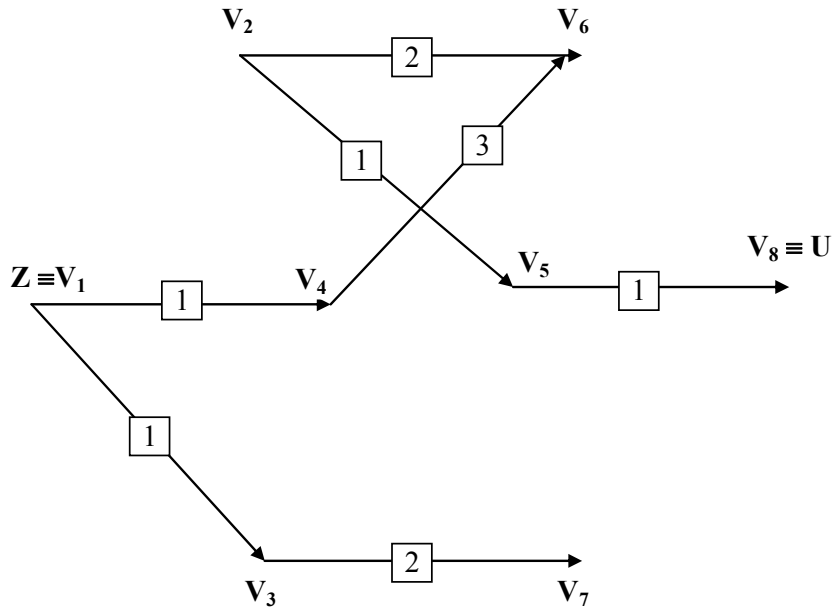
Riešenie:

Prvou najvyššie položenou dráhou zo zdroja do ústia je zrejme dráha - V_1, V_2, V_6 a V_8 - tvorená hranami h_{12}, h_{26}, h_{68} . Minimálnu kapacitu z nich má hrana $h_{68}=4$, ktorú z grafu vynecháme a kapacity hrán h_{12} a h_{26} znížime o 4 ($h_{12}=1, h_{26}=2$). Graf po 1. iterácii je na obr.15a.



Obr.15a Maximálny tok v sieti - graf po 1. iterácii a ďalšie riešenie.

Je zrejmé, že ďalšia využiteľná najvyššie položená cesta je tvorená hranami h_{12} , h_{25} a h_{58} . Tu je min. kapacita hrany $h_{12}=1$. Hranu h_{12} odstránime (prečiarkneme) a kapacity hrán h_{25} a h_{58} zmenšíme o 1 (hodnoty uvedené v zátvorkách). V ďalšom kroku odstránime hrany $h_{47}=3$, $h_{35}=3$ a $h_{78}=6$. Takto dostaneme graf podľa obr.15b. Vidíme, že už žiadny tok zo Z do Ú nie je možné vytvoriť.



Obr.15b Maximálny tok v sieti – konečná podoba grafu.

Maximálny tok v zadanej sieti: je určený súčtom kapacít „odstránených“ hrán (v momente ich odstraňovania) a pre zadaný príklad to znamená, že maximálny možný tok v sieti je

$$z = h_{12} + h_{68} + h_{47} + h_{35} + h_{78} = 4 + 1 + 3 + 3 + 3 = 14.$$

Výsledok: všetky uvedené hrany budú nasýtené (bude vyčerpaná ich kapacita), naopak u hrán ostatných voľná kapacita zostáva a je prípadne využiteľná pre ďalšie požiadavky (obr.15b).

Pre výpočet maximálneho toku v sieti často využívame aj nasledovný algoritmus:

Algoritmus určenia maximálneho toku (priepustnosti) v sieti:

1. *Nájdeme ľubovoľnú cestu zo vstupného do výstupného uzla siete s kladnými kapacitami hrán, pričom uvažujeme cesty bez ohľadu na orientáciu hrán. Pokiaľ taká cesta neexistuje, bolo nájsené optimálne riešenie.*
2. *Na ceste nájdenej v kroku 1 nájdeme hranu s najnižšou kapacitou – označme ju napr. Δ . O túto hodnotu Δ zvýšime celkový tok v sieti.*
3. *Na rovnakej ceste znížime kapacitu všetkých hrán v smere zo zdroja do ústia o hodnotu Δ a zvýšime kapacitu všetkých hrán o rovnakú hodnotu v smere opačnom. Vrátime sa do kroku 1.*
4. *Postup opakujeme až pokiaľ nebude existovať cesta zo zdroja do ústia obsahujúca iba kladné kapacity hodnoty toku hrán. **Doplniť príklad na maximálny tok v sieti, str. 179.***

8.6.4 Ďalšie typické úlohy teórie grafov

- **Úloha o farbení grafu** – ide o úlohu určenia minimálneho počtu farieb tak, aby boli všetky vrcholy grafu ofarbené a žiadna hrana nespájala vrcholy sfarbené rovnakou farbou. *Aplikácie: určenie minimálneho možného počtu frekvencií vysielateľov tak, aby bolo pokryté celé územie a susedné vysielacie sa navzájom nerušil; nájdenie minimálneho počtu farieb, ktoré musíme použiť pri tlači tzv. politickej mapy sveta, kde je potrebné, aby žiadne dva susedné štáty neboli sfarbené rovnakou farbou a iné.*
- **Úloha čínskeho poštára** – cieľom je nájdenie najkratšieho možného sledu, t.j. trasy v grafe tak, aby táto trasa obsahovala všetky hrany grafu (analógia s poštárom, ktorý pri roznášaní pošty musí prejsť všetkými ulicami svojho okrsku). Ide v podstate o nájdenie tzv. *Eulerovho sledu s minimálnou celkovou dĺžkou*. Ak by bolo možné Eulerov ťah v grafe nájsť, tento ťah by bol vlastne optimálnym riešením aj tejto úlohy. Platí totiž, pokiaľ musíme prejsť po všetkých hranách grafu G , potom nie je možné nájsť kratšiu trasu ako tú, pri ktorej prechádzame každou hranou práve raz. ***Dĺžka Eulerovho ťahu je rovná súčtu ohodnotení všetkých hrán grafu G .***
- **Úloha obchodného cestujúceho** – ide o nájdenie najkratšieho možného sledu, t.j. trasy v grafe tak, aby táto trasa obsahovala všetky vrcholy grafu (analógia s obchodným cestujúcim, ktorý

musí navštíviť všetky miesta, kde sídlia jeho obchodný partneri). Z hľadiska teórie grafov ide o nájdenie tzv. *Hamiltonovskej kružnice s minimálnou celkovou dĺžkou*.

- **Úloha optimálneho umiestnenia depa** – ide o nájdenie vrcholu alebo miesta na hrane v zadanej sieti, ktoré by mohlo slúžiť ako obslužné stredisko pre ostatné vrcholy a bolo v danej sieti vhodne umiestnené z hľadiska vzdialeností do obsluhovaných vrcholov. Predpokladá sa, že každý vrchol je ohodnotený číslom vyjadrujúcim náročnosť jeho obsluhy (napr. požadovaná početnosť obsluhy). Vzdialenostne optimálne umiestnenie depa v danej sieti je určené bodom, ktorý má najmenšiu tzv. **váženú excentricitu**. Túto pre konkrétny bod určíme ako maximálnu hodnotu zo súčiny vzdialeností ku všetkým vrcholom grafu a ohodnotení týchto vrcholov.

Záver:

Ďalšiu pozornosť budeme venovať skupine úloh riešených na grafoch, ktoré nazývame **sieťová analýza**, resp. často aj ako **analýza projektov**. **Sieťový graf (sieť)** – je orientovaný, nezáporne ohodnotený, súvislý, acyklický graf (neobsahuje cyklus), obsahujúci práve jeden vstupný vrchol, z ktorého hrany iba vychádzajú a práve jeden výstupný vrchol, do ktorého hrany iba vstupujú.

<https://managementmania.com/sk/metoda-kritickej-cesty-cpm-critical-path-method>

<http://www.svetdopravy.sk/sietove-analyzy-cestnej-siete-v-prostredi-geografickych-informacnych-systemov/>

Jedným z najjednoduchších algoritmov je tzv. **metóda najbližšieho suseda**. Podstata spočíva v tom, že každému uzlu je priradená hodnota $t_1, t_2, \dots, t_n$.

Algoritmus hľadania minimálnej cesty v grafe:

1. Hodnota vo východiskovom uzle (predpokladajme, že východiskovým uzlom je napr. uzol u_1 , ale vo všeobecnosti to môže byť ktorýkoľvek uzol v grafe) je rovná nule, t.j. $t_1=0$.
2. V ďalších krokoch počítame hodnoty t_k v ďalších uzloch nasledovne:

$$t_k = \min_j (t_i + y_{ij})$$

kde i sú indexy uzlov, pre ktoré hodnotu t_i z predchádzajúcich krokov už poznáme, j sú indexy uzlov, pre ktoré je hodnota t_j zatiaľ ešte neznáma a platí, že z uzla u_i vedie do uzla u_j hrana h_{ij} s ohodnotením y_{ij} .

3. Predchádzajúci krok iteračného postupu opakujeme dovtedy, pokiaľ nie je vyrátaná hodnota t_j pre určený koncový uzol (alebo pre všetky uzly grafu – v závislosti od zadania úlohy).
4. Hodnoty t_i , pre $i = 2, 3, \dots, n$ potom predstavujú dĺžku najkratšej cesty medzi uzlom u_1 a u_i . Najkratšia cesta je tvorená hranami, pre ktoré platí

$$t_j - t_i = y_{ij}.$$

