2.2 Minimální cesty z každého vrcholu do každého

Floydův algoritmus

Floydův algoritmus je vhodné použít pro sítě s počtem hran blížícím se úplnému grafu. Je vhodný pro orientované i neorientované grafy, v našem případě budeme uvažovat pouze graf neorientovaný.

Pro sítě s malým počtem hran je z hlediska času výpočtu výhodné několikanásobné použití např. Dijkstrova algoritmu pro výpočet stromu minimálních tras, kdy každý vrchol považujeme v jednotlivých výpočtech za výchozí, tj. za kořen stromu.

Postup řešení:

1. krok: Sestavíme nejprve počáteční čtvercovou matici $C = (c_{ij})_{i,j=1}^{n}$, tj. matici přímých vzdáleností typu *n x n* tak, aby platilo pro prvky c_{ij}:

c_{ij} — píšu do první matice z přímých vzdáleností

$c_{ij} = o(h)$, jestliže $\exists h \in H: \; p(h) = (v_i, v_j), \; i \neq j$

$c_{ij} = 0$ pro $i = j$

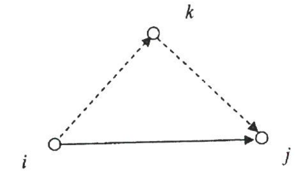
$c_{ij} = \infty$, jestliže neexistuje $h \in H: p(h) = (v_i, v_j), \; i \neq j$

Pro neorientovaný graf bude matice *C* i každá další matice, která z ní vznikne přepočtem pomocí Floydova algoritmu, symetrická , což nám výpočet urychlí.

V případě, že by hrana (v_i, v_i) byla smyčka, potom prvek c_{ij} bude roven ohodnocení smyčky. Případné nenulové prvky na hlavní diagonále neovlivní výpočet matice vzdáleností. V případě, že bychom chtěli zohlednit vzdálenosti nutné pro projetí uzly, můžeme je k vypočteným vzdálenostem jednotlivých tras přičíst.

2. krok: Postupně konstruujeme posloupnost matic $C_0, C_1, C_2, \cdots, C_{k-1}, C_{k=n}$, přičemž matice C_0 je počáteční (výchozí) matice přímých vzdáleností. V cyklu pro *k = 1, ..., n* hledáme zda cestu z vrcholu *i* do *j* nelze zkrátit přes vrchol *k*.

Motivace



Obr. 2.2c

3. krok: Přepočet prvků matice *C* provedeme podle vztahu:

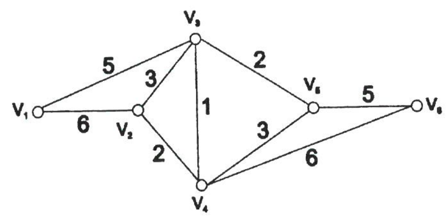
$$c_{ij}^{(k)} = \min\{c_{ij}^{(k-1)}, c_{ik}^{(k-1)} + c_{kj}^{(k-1)}\},$$

kde horní indexy určují, o kterou matici se jedná

4. krok: Výsledná matice *C* je maticí vzdáleností, neboli maticí distanční. Obdobný postup se dá použít pro celou řadu dalších úloh, např. pro maximální propustnost sítě (přepočet prvků potom bude podle vztahu:

$$c_{ij}^{(k)} = \max\{c_{ij}^{(k-1)}, \min(c_{ik}^{(k-1)}, c_{kj}^{(k-1)})\}$$

Příklad 2.3 – určete distanční matici a pomocí ní určete minimální cestu z vrcholu V_1 do vrcholu V_5.



Obr. 2.3

1. krok: Úlohu budeme řešit Floydovým algoritmem, nejdříve sestavíme počáteční čtvercovou matici, matici přímých vzdáleností $C = (c_{ij})$.

$$C_{k=0} = \begin{array}{c|cccccc} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ \hline v_1 & 0 & 6 & 5 & \infty & \infty & \infty \\ v_2 & 6 & 0 & 3 & 2 & \infty & \infty \\ v_3 & 5 & 3 & 0 & 1 & 2 & \infty \\ v_4 & \infty & 2 & 1 & 0 & 3 & 6 \\ v_5 & \infty & \infty & 2 & 3 & 0 & 5 \\ v_6 & \infty & \infty & \infty & 6 & 5 & 0 \end{array}$$

neexistuje přímá cesta

2. a 3. krok: Matice $C_{k=1}$......$C_{k=6}$ podle přepočtu $c_{ij}^{(k)} = \min\{c_{ij}^{(k-1)}, c_{ik}^{(k-1)} + c_{kj}^{(k-1)}\}$

$C_{k=1}$: v matici přímých vzdáleností vyškrtneme první řádek a první sloupec, prvky přepíšeme do další matice a ostatní prvky matice dle přepočtu přepočítáme.

Např. prvek nové matice $c_{32}^{1} = \min\{c_{32}^{0}, c_{31}^{0} + c_{12}^{0}\} = \min\{3, 5+6\} = 3$ se nezmění.

Matice $C_{k=1}$ zůstává stejná:

$$C_{k=1} = \begin{array}{cccccc} 0 & 6 & 5 & \infty & \infty & \infty \\ 6 & 0 & 3 & 2 & \infty & \infty \\ 5 & 3 & 0 & 1 & 2 & \infty \\ \infty & 2 & 1 & 0 & 3 & 6 \\ \infty & \infty & 2 & 3 & 0 & 5 \\ \infty & \infty & \infty & 6 & 5 & 0 \end{array}$$

dosazuji min vzdálen.
nebo součet hran

$C_{k=2}$: v matici $C_{k=1}$ vyškrtneme druhý řádek a druhý sloupec, prvky přepíšeme do další matice, ostatní přepočítáme.

Změna nastane u prvku c_{14}, c_{41}, protože matice je symetrická.

$c_{14}^{2} = \min\{c_{14}^{1}, c_{12}^{1} + c_{24}^{1}\} = \min\{\infty, 6+2\} = 8$

$$C_{k=2} = \begin{array}{cccccc} 0 & 6 & 5 & 8 & \infty & \infty \\ 6 & 0 & 3 & 2 & \infty & \infty \\ 5 & 3 & 0 & 1 & 2 & \infty \\ 8 & 2 & 1 & 0 & 3 & 6 \\ \infty & \infty & 2 & 3 & 0 & 5 \\ \infty & \infty & \infty & 6 & 5 & 0 \end{array}$$

$C_{k=3}$: změní se prvky $c_{14}, c_{41}, c_{15}, c_{51}, c_{25}, c_{52}$.

$c_{14}^{3} = \min\{c_{14}^{2}, c_{13}^{2} + c_{34}^{2}\} = \min\{8, 5+1\} = 6$

$c_{15}^{3} = \min\{c_{15}^{2}, c_{13}^{2} + c_{35}^{2}\} = \min\{\infty, 5+2\} = 7$

$c_{25}^{3} = \min\{c_{25}^{2}, c_{23}^{2} + c_{35}^{2}\} = \min\{\infty, 3+2\} = 5$

ky

do

$$C_{k=3} = \begin{array}{cccccc} 0 & 6 & 5 & 6 & 7 & \infty \\ 6 & 0 & 3 & 2 & 5 & \infty \\ 5 & 3 & 0 & 1 & 2 & \infty \\ 6 & 2 & 1 & 0 & 3 & 6 \\ 7 & 5 & 2 & 3 & 0 & 5 \\ \infty & \infty & \infty & 6 & 5 & 0 \end{array}$$

$C_{k=4}$: změní se prvky $c_{16}, c_{61}, c_{26}, c_{62}, c_{36}, c_{63}$

$$C_{k=4} = \begin{array}{cccccc} 0 & 6 & 5 & 6 & 7 & 12 \\ 6 & 0 & 3 & 2 & 5 & 8 \\ 5 & 3 & 0 & 1 & 2 & 7 \\ 6 & 2 & 1 & 0 & 3 & 6 \\ 7 & 5 & 2 & 3 & 0 & 5 \\ 12 & 8 & 7 & 6 & 5 & 0 \end{array} = C_{k=5} = C_{k=6}$$

4. krok: Tato matice je výslednou distanční maticí.

Hledáme minimální cestu z V_1 do V_5 pomocí distanční matice

Do matice přímých vzdáleností místo ∞ budeme psát *0* a matici doplníme o první sloupec (řádek) distanční matice, první proto, že hledáme cestu z V_1. – proto bez první sloupce

$$C = \begin{array}{cccccc|c} 0 & 6 & 5 & 0 & 0 & 0 & 0 \\ 6 & 0 & 3 & 2 & 0 & 0 & 6 \\ 5 & 3 & 0 & 1 & 2 & 0 & 5 \\ 0 & 2 & 1 & 0 & 3 & 6 & 6 \\ 0 & 0 & 2 & 3 & 0 & 5 & 7 \\ 0 & 0 & 0 & 6 & 5 & 0 & 12 \end{array}$$

Koncový vrchol je V_5, proto začneme cestu hledat na 5. řádku od konce prvkem c_{15}^{6} a hledáme jeho předchůdce p.

Musí platit: $c_{15}^{6} - c_{5p} = c_{1p}^{6}$

pro p = 1: $7 - 0 = 7 \neq 0$

pro p = 2: $7 - 0 = 7 \neq 6$

pro p = 3: $7 - 2 = 5 = 5$

pro p = 4: $7 - 3 = 4 \neq 6$

pro p = 5: nepočítáme

pro p = 6: $7 - 5 = 2 \neq 12$

Rovnost platí pro p = 3, do cesty zařadíme V_3 a stejným způsobem pokračujeme dál.

$$C_{k=3} = \begin{matrix} 0 & 6 & 5 & 6 & 7 & \infty \\ 6 & 0 & 3 & 2 & 5 & \infty \\ 5 & 3 & 0 & 1 & 2 & \infty \\ 6 & 2 & 1 & 0 & 3 & 6 \\ 7 & 5 & 2 & 3 & 0 & 5 \\ \infty & \infty & \infty & 6 & 5 & 0 \end{matrix}$$

$C_{k=4}$: změní se prvky $c_{16}, c_{61}, c_{26}, c_{62}, c_{36}, c_{63}$

$$\boxed{C_{k=4} = \begin{matrix} 0 & 6 & 5 & 6 & 7 & 12 \\ 6 & 0 & 3 & 2 & 5 & 8 \\ 5 & 3 & 0 & 1 & 2 & 7 \\ 6 & 2 & 1 & 0 & 3 & 6 \\ 7 & 5 & 2 & 3 & 0 & 5 \\ 12 & 8 & 7 & 6 & 5 & 0 \end{matrix} = C_{k=5} = C_{k=6}}$$

4. krok: Tato matice je výslednou distanční maticí.

Hledáme minimální cestu z V_1 do V_5 pomocí distanční matice

Do matice přímých vzdáleností místo ∞ budeme psát *0* a matici doplníme o první sloupec (řádek) distanční matice, první proto, že hledáme cestu z V_1. ~ proto len první sloupec

$$C = \left.\begin{matrix} 0 & 6 & 5 & 0 & 0 & 0 \\ 6 & 0 & 3 & 2 & 0 & 0 \\ 5 & 3 & 0 & 1 & 2 & 0 \\ 0 & 2 & 1 & 0 & 3 & 6 \\ 0 & 0 & 2 & 3 & 0 & 5 \\ 0 & 0 & 0 & 6 & 5 & 0 \end{matrix}\right| \begin{matrix} 0 \\ 6 \\ 5 \\ 6 \\ 7 \\ 12 \end{matrix}$$

5 3 0 1 2 0

5 − 5 = 0 = 0
5 − 3 = 2 ≠ 6
5 − 1 = 4 ≠ 6
5 − 2 = 3 ≠ 7
5 − 0 = 5 ≠ 12

Koncový vrchol je V_5, proto začneme cestu hledat na 5. řádku od konce prvkem c_{15}^{6} a hledáme jeho předchůdce p.

Musí platit: $c_{15}^{6} - c_{5p} = c_{1p}^{6}$

pro p = 1: $7 - 0 = 7 \neq 0$

pro p = 2: $7 - 0 = 7 \neq 6$

pro p = 3: $7 - 2 = 5 = 5$

pro p = 4: $7 - 3 = 4 \neq 6$

pro p = 5: nepočítáme

pro p = 6: $7 - 5 = 2 \neq 12$

Rovnost platí pro p = 3, do cesty zařadíme V_3 a stejným způsobem pokračujeme dál.

V případě, že existuje více rovností platí, že existuje více minimálních cest.

pro p = 1: $5 - 5 = 0 = 0$

pro p = 2: $5 - 3 = 2 \neq 6$

pro p = 3: nepočítáme

pro p = 4: $5 - 1 = 3 \neq 6$

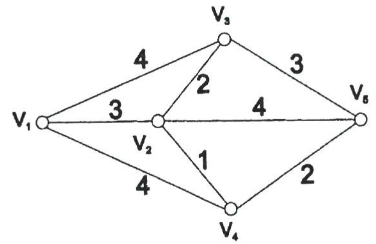
pro p = 5: $5 - 2 = 3 \neq 7$

pro p = 6: $5 - 0 = 5 \neq 12$

Rovnost platí pro p = 1, do cesty zařadíme V_1.

Nalezená cesta povede z V_1, V_3, V_5 a její délka bude z distanční matice 7 délkových jednotek.

Příklad 2.4 – aplikace Floydova algoritmu pro výpočet maximální propustnosti sítě, $o(h)$ jsou kapacity hran.



Obr. 2.4a

1. krok: Sestavíme nejprve počáteční čtvercovou matici $C = (c_{ij})_{i,j=1}^{n}$, tj. matici přímých vzdáleností typu *n x n* tak, aby platilo pro prvky c_{ij}:

c_{ij}

- $c_{ij} = o(h)$, *jestliže* $\exists h \in H: p(h) = (v_i, v_j), i \neq j$
- $c_{ij} = \infty$ *pro* $i = j$
- $c_{ij} = 0$, *jestliže neexistuje* $h \in H: p(h) = (v_i, v_j), i \neq j$