

GAaDO

Najkratšia cesta - algoritmy

Technická univerzita v Košiciach



2025–2026

Najkratšia cesta - algoritmy

Poznáme viacero algoritmov, ktoré dokážu nájsť najkratšiu cestu. Medzi algoritmy, ktoré hľadajú najkratšiu cestu z jedného vrchola do všetkých ostatných patria napríklad:

Poznáme viacero algoritmov, ktoré dokážu nájsť najkratšiu cestu. Medzi algoritmy, ktoré hľadajú najkratšiu cestu z jedného vrchola do všetkých ostatných patria napríklad:

- Fordov algoritmus
- Dijkstrov algoritmus
- Johnsonov algoritmus

Poznáme viacero algoritmov, ktoré dokážu nájsť najkratšiu cestu. Medzi algoritmy, ktoré hľadajú najkratšiu cestu z jedného vrchola do všetkých ostatných patria napríklad:

- Fordov algoritmus
- Dijkstrov algoritmus
- Johnsonov algoritmus

Najznámejší algoritmus hľadajúci najkratšiu cestu medzi všetkými vrcholmi grafu patrí Floydov algoritmus.

Floydov algoritmus

Floydov algoritmus

- **Krok 1.** Zostroj $C = (c_{ij})$, kde: $c_{ii} = 0$ pre $\forall i \in V$ a $\forall i, j, i \neq j$:

$$c_{ij} = \begin{cases} c(i, j) & \text{ak } ij \in H, \text{ resp. } \overrightarrow{ij} \in H \\ \infty & \text{ak } ij \notin H, \text{ resp. } \overrightarrow{ij} \notin H \end{cases}$$

Floydov algoritmus

Floydov algoritmus

- **Krok 1.** Zostroj $C = (c_{ij})$, kde: $c_{ii} = 0$ pre $\forall i \in V$ a $\forall i, j, i \neq j$:

$$c_{ij} = \begin{cases} c(i, j) & \text{ak } ij \in H, \text{ resp. } \overrightarrow{ij} \in H \\ \infty & \text{ak } ij \notin H, \text{ resp. } \overrightarrow{ij} \notin H \end{cases}$$

Zostroj $X = (x_{ij})$, kde $x_{ii} = i$ pre $\forall i \in V$ a $\forall i, j, i \neq j$:

$$x_{ij} = \begin{cases} i & \text{ak } ij \in H, \text{ resp. } \overrightarrow{ij} \in H \\ \infty & \text{ak } ij \notin H, \text{ resp. } \overrightarrow{ij} \notin H \end{cases}$$

Floydov algoritmus

- **Krok 1.** Zostroj $C = (c_{ij})$, kde: $c_{ii} = 0$ pre $\forall i \in V$ a $\forall i, j, i \neq j$:

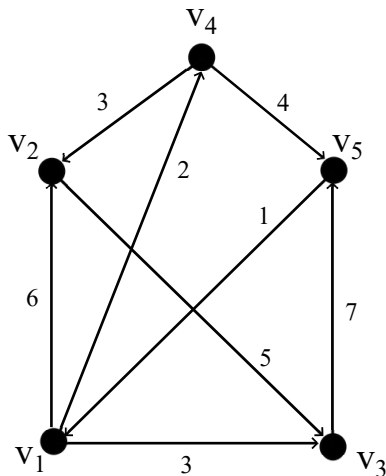
$$c_{ij} = \begin{cases} c(i, j) & \text{ak } ij \in H, \text{ resp. } \overrightarrow{ij} \in H \\ \infty & \text{ak } ij \notin H, \text{ resp. } \overrightarrow{ij} \notin H \end{cases}$$

Zostroj $X = (x_{ij})$, kde $x_{ii} = i$ pre $\forall i \in V$ a $\forall i, j, i \neq j$:

$$x_{ij} = \begin{cases} i & \text{ak } ij \in H, \text{ resp. } \overrightarrow{ij} \in H \\ \infty & \text{ak } ij \notin H, \text{ resp. } \overrightarrow{ij} \notin H \end{cases}$$

- **Krok 2.** Pre $\forall k = 1, 2, \dots, n$ urob: Pre $\forall i \neq k$ také, že $c_{ik} \neq \infty$, a pre $\forall j \neq k$ také, že $c_{kj} \neq \infty$, urob:
Ak $c_{ij} > c_{ik} + c_{kj}$, polož: $c_{ij} = c_{ik} + c_{kj}$ a $x_{ij} := x_{kj}$.

Floydov algoritmus



Napíšte maticu vzdialeností pre daný digraf a popíšte najkratšiu cestu z vrcholu v_2 do vrcholu v_4 .

Floydov algoritmus

1. iteračný krok:

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

v_1	v_1	v_1	v_1	—
—	v_2	v_2	—	—
—	—	v_3	—	v_3
—	v_4	—	v_4	v_4
v_5	—	—	—	v_5

Floydov algoritmus

1. iteračný krok:

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

v_1	v_1	v_1	v_1	—
—	v_2	v_2	—	—
—	—	v_3	—	v_3
—	v_4	—	v_4	v_4
v_5	—	—	—	v_5

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

v_1	v_1	v_1	v_1	—
—	v_2	v_2	—	—
—	—	v_3	—	v_3
—	v_4	—	v_4	v_4
v_5	v_1	v_1	v_1	v_5

Floydov algoritmus

2. iteračný krok:

$$\left(\begin{array}{c|c|c|c|c} 0 & 6 & 3 & 2 & \infty \\ \hline \infty & 0 & 5 & \infty & \infty \\ \hline \infty & \infty & 0 & \infty & 7 \\ \hline \infty & 3 & \infty & 0 & 4 \\ \hline 1 & 7 & 4 & 3 & 0 \end{array} \right)$$
$$\left(\begin{array}{c|c|c|c|c} v_1 & v_1 & v_1 & v_1 & - \\ \hline - & v_2 & v_2 & - & - \\ \hline - & - & v_3 & - & v_3 \\ \hline - & v_4 & - & v_4 & v_4 \\ \hline v_5 & v_1 & v_1 & v_1 & v_5 \end{array} \right)$$

Floydov algoritmus

2. iteračný krok:

$$\begin{pmatrix} 0 & 6 & 3 & 2 & \infty \\ \infty & 0 & 5 & \infty & \infty \\ \infty & \infty & 0 & \infty & 7 \\ \infty & 3 & \infty & 0 & 4 \\ 1 & 7 & 4 & 3 & 0 \end{pmatrix}$$

$$\begin{pmatrix} v_1 & v_1 & v_1 & v_1 & - \\ - & v_2 & v_2 & - & - \\ - & - & v_3 & - & v_3 \\ - & v_4 & - & v_4 & v_4 \\ v_5 & v_1 & v_1 & v_1 & v_5 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 6 & 3 & 2 & \infty \\ \infty & 0 & 5 & \infty & \infty \\ \infty & \infty & 0 & \infty & 7 \\ \infty & 3 & 8 & 0 & 4 \\ 1 & 7 & 4 & 3 & 0 \end{pmatrix}$$

$$\begin{pmatrix} v_1 & v_1 & v_1 & v_1 & - \\ - & v_2 & v_2 & - & - \\ - & - & v_3 & - & v_3 \\ - & v_4 & v_2 & v_4 & v_4 \\ v_5 & v_1 & v_1 & v_1 & v_5 \end{pmatrix}$$

Floydov algoritmus

3. iteračný krok:

$$\left(\begin{array}{cc|cc|c} 0 & 6 & 3 & 2 & \infty \\ \infty & 0 & 5 & \infty & \infty \\ \hline \infty & \infty & 0 & \infty & 7 \\ \hline \infty & 3 & 8 & 0 & 4 \\ 1 & 7 & 4 & 3 & 0 \end{array} \right)$$

$$\left(\begin{array}{cc|cc|c} v_1 & v_1 & v_1 & v_1 & - \\ - & v_2 & v_2 & - & - \\ \hline - & - & v_3 & - & v_3 \\ \hline - & v_4 & v_2 & v_4 & v_4 \\ v_5 & v_1 & v_1 & v_1 & v_5 \end{array} \right)$$

Floydov algoritmus

3. iteračný krok:

$$\begin{pmatrix} 0 & 6 & 3 & 2 & \infty \\ \infty & 0 & 5 & \infty & \infty \\ \hline \infty & \infty & 0 & \infty & 7 \\ \hline \infty & 3 & 8 & 0 & 4 \\ 1 & 7 & 4 & 3 & 0 \end{pmatrix}$$

$$\begin{pmatrix} v_1 & v_1 & v_1 & v_1 & - \\ - & v_2 & v_2 & - & - \\ \hline - & - & v_3 & - & v_3 \\ \hline - & v_4 & v_2 & v_4 & v_4 \\ v_5 & v_1 & v_1 & v_1 & v_5 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 6 & 3 & 2 & 10 \\ \infty & 0 & 5 & \infty & 12 \\ \hline \infty & \infty & 0 & \infty & 7 \\ \hline \infty & 3 & 8 & 0 & 4 \\ 1 & 7 & 4 & 3 & 0 \end{pmatrix}$$

$$\begin{pmatrix} v_1 & v_1 & v_1 & v_1 & v_3 \\ - & v_2 & v_2 & - & v_3 \\ \hline - & - & v_3 & - & v_3 \\ \hline - & v_4 & v_2 & v_4 & v_4 \\ v_5 & v_1 & v_1 & v_1 & v_5 \end{pmatrix}$$

4. iteračný krok:

$$\left(\begin{array}{ccc|c|c} 0 & 6 & 3 & 2 & 10 \\ \infty & 0 & 5 & \infty & 12 \\ \infty & \infty & 0 & \infty & 7 \\ \hline \infty & 3 & 8 & 0 & 4 \\ \hline 1 & 7 & 4 & 3 & 0 \end{array} \right)$$

$$\left(\begin{array}{ccccc} v_1 & v_1 & v_1 & v_1 & v_3 \\ - & v_2 & v_2 & - & v_3 \\ - & - & v_3 & - & v_3 \\ \hline - & v_4 & v_2 & v_4 & v_4 \\ \hline v_5 & v_1 & v_1 & v_1 & v_5 \end{array} \right)$$

Floydov algoritmus

4. iteračný krok:

$$\left(\begin{array}{ccc|c|c} 0 & 6 & 3 & 2 & 10 \\ \infty & 0 & 5 & \infty & 12 \\ \infty & \infty & 0 & \infty & 7 \\ \hline \infty & 3 & 8 & 0 & 4 \\ \hline 1 & 7 & 4 & 3 & 0 \end{array} \right)$$

$$\left(\begin{array}{ccccc} v_1 & v_1 & v_1 & v_1 & v_3 \\ - & v_2 & v_2 & - & v_3 \\ - & - & v_3 & - & v_3 \\ \hline - & v_4 & v_2 & v_4 & v_4 \\ \hline v_5 & v_1 & v_1 & v_1 & v_5 \end{array} \right)$$

$$\left(\begin{array}{ccccc} 0 & 5 & 3 & 2 & 6 \\ \infty & 0 & 5 & \infty & 12 \\ \infty & \infty & 0 & \infty & 7 \\ \infty & 3 & 8 & 0 & 4 \\ \hline 1 & 6 & 4 & 3 & 0 \end{array} \right)$$

$$\left(\begin{array}{ccccc} v_1 & v_4 & v_1 & v_1 & v_4 \\ - & v_2 & v_2 & - & v_3 \\ - & - & v_3 & - & v_3 \\ - & v_4 & v_2 & v_4 & v_4 \\ \hline v_5 & v_4 & v_1 & v_1 & v_5 \end{array} \right)$$

Floydov algoritmus

5. iteračný krok:

0	5	3	2	6
∞	0	5	∞	12
∞	∞	0	∞	7
∞	3	8	0	4
1	6	4	3	0

v_1	v_4	v_1	v_1	v_4
—	v_2	v_2	—	v_3
—	—	v_3	—	v_3
—	v_4	v_2	v_4	v_4
v_5	v_4	v_1	v_1	v_5

Floydov algoritmus

5. iteračný krok:

$$\left(\begin{array}{cccc|c} 0 & 5 & 3 & 2 & 6 \\ \infty & 0 & 5 & \infty & 12 \\ \infty & \infty & 0 & \infty & 7 \\ \infty & 3 & 8 & 0 & 4 \\ \hline 1 & 6 & 4 & 3 & 0 \end{array} \right)$$

$$\left(\begin{array}{cccc|c} v_1 & v_4 & v_1 & v_1 & v_4 \\ - & v_2 & v_2 & - & v_3 \\ - & - & v_3 & - & v_3 \\ - & v_4 & v_2 & v_4 & v_4 \\ \hline v_5 & v_4 & v_1 & v_1 & v_5 \end{array} \right)$$

Výsledná vzdialenostná matica:

$$\left(\begin{array}{ccccc} 0 & 5 & 3 & 2 & 6 \\ 13 & 0 & 5 & 15 & 12 \\ 8 & 13 & 0 & 10 & 7 \\ 5 & 3 & 8 & 0 & 4 \\ 1 & 6 & 4 & 3 & 0 \end{array} \right)$$

$$\left(\begin{array}{ccccc} v_1 & v_4 & v_1 & v_1 & v_4 \\ v_5 & v_2 & v_2 & v_1 & v_3 \\ v_5 & v_4 & v_3 & v_1 & v_3 \\ v_5 & v_4 & v_2 & v_4 & v_4 \\ v_5 & v_4 & v_1 & v_1 & v_5 \end{array} \right)$$

Zložitosť Floydovho algoritmu:

Zložitosť Floydovho algoritmu:

- Vytvorenie matíc C a X v Kroku 1. má zložitosť $O(n^2)$,

Zložitosť Floydovho algoritmu:

- Vytvorenie matíc C a X v Kroku 1. má zložitosť $O(n^2)$,
- V kroku 2 sa n -krát skontrolujú a prípadne aj zmenia prvky matice C a matice X , ktoré obe majú po n^2 prvkov, zložitosť kroku 3. je $O(n^3)$,

Zložitosť Floydovho algoritmu:

- Vytvorenie matíc C a X v Kroku 1. má zložitosť $O(n^2)$,
- V kroku 2 sa n -krát skontrolujú a prípadne aj zmenia prvky matice C a matice X , ktoré obe majú po n^2 prvkov, zložitosť kroku 3. je $O(n^3)$,
- Zložitosť celého algoritmu je $O(n^3)$.

Modifikácia Floydovho algoritmu:

- v Kroku 1. definujeme začiatočnú maticu C tak, že na diagonále namiesto núl budú hodnoty ∞ . (Matica X je bez zmeny.)
- Krok 2. je rovnaký. Po ukončení algoritmu budú prvky c_{ij} na diagonále rovné dĺžke najkratšieho ii -cyklu.
- Ak sa na hlavnej diagonále matice C objaví záporné číslo c_{ii} , máme cyklus zápornej ceny. Tento cyklus určíme pomocou matice smerníkov X

Cesta maximálnej spoľahlivosti

Definícia

Majme hranovo ohodnotený graf $G = (V, H, c)$ kde ohodnotenie c predstavuje spoľahlivosť hrany (pravdepodobnosť úspešného prechodu hranou), t.j. $0 \leq c(h) \leq 1$. Nech P je uv-cesta. Spoľahlivosť $s(P)$ cesty P definujeme:

$$s(P) = \prod_{h \in P} c(h).$$

uv-cesta **maximálnej spoľahlivosti** je tá uv-cesta P , ktorá má zo všetkých uv-ciest najväčšiu spoľahlivosť.

Cesta maximálnej spoľahlivosti

Veta

*Nech $G = (V, H, c)$, kde $c(h) > 0$ je spoľahlivosť hrany $h \in H$.
uv-cesta P je cestou maximálnej spoľahlivosti v grafe
 $G = (V, H, c)$ práve vtedy, ak P je najkratšou cestou v grafe
 $G = (V, H, c)$, kde pre cenu hrany c platí $c(i, j) = -\log_z(c(i, j))$,
(kde $z > 1$).*

Cesta maximálnej spoľahlivosti

Dôkaz:

- Funkcia $\log_z(x)$ je rastúca (pre $z > 1$),

Cesta maximálnej spoľahlivosti

Dôkaz:

- Funkcia $\log_z(x)$ je rastúca (pre $z > 1$),
- spoľahlivosť $s(P) = \prod_{h \in P} c(h)$ cesty P je maximálna práve vtedy, keď je maximálny výraz:

$$\log(s(P)) = \log \left(\prod_{h \in P} c(h) \right) = \sum_{h \in P} \log(c(h)).$$

Cesta maximálnej spoľahlivosti

Dôkaz:

- Funkcia $\log_z(x)$ je rastúca (pre $z > 1$),
- spoľahlivosť $s(P) = \prod_{h \in P} c(h)$ cesty P je maximálna práve vtedy, keď je maximálny výraz:

$$\log(s(P)) = \log\left(\prod_{h \in P} c(h)\right) = \sum_{h \in P} \log(c(h)).$$

- tento výraz je maximálny práve vtedy, keď je súčet $\sum_{h \in P} -\log(c(h))$ minimálny. Pretože $0 < c(h) \leq 1$, je $-\log(c(h))$ nezáporné číslo.

Cesta maximálnej spoľahlivosti

Dôkaz:

- Funkcia $\log_z(x)$ je rastúca (pre $z > 1$),
- spoľahlivosť $s(P) = \prod_{h \in P} c(h)$ cesty P je maximálna práve vtedy, keď je maximálny výraz:

$$\log(s(P)) = \log \left(\prod_{h \in P} c(h) \right) = \sum_{h \in P} \log(c(h)).$$

- tento výraz je maximálny práve vtedy, keď je súčet $\sum_{h \in P} -\log(c(h))$ minimálny. Pretože $0 < c(h) \leq 1$, je $-\log(c(h))$ nezáporné číslo.
- Pre cestu P je táto suma vlastne dĺžkou cesty v hranovo ohodnotenom grafe $G = (V, H, c)$ s cenou hrany $c(h) = -\log(c(h))$.

- Misionári a kanibali.

- Misionári a kanibali.
- Jazdec na šachovnici.

- Misionári a kanibali.
- Jazdec na šachovnici.
- Odmeriavanie vody.

- Misionári a kanibali.
- Jazdec na šachovnici.
- Odmeriavanie vody.
- Elektronický cestovný poriadok.