Pochôdzky v grafoch

Stanislav Palúch

Fakulta riadenia a informatiky, Žilinská univerzita

11. apríla 2011

Eulerovské sledy a eulerovské ťahy

Definícia

Hovoríme, že sled $s(u, v)$ v súvislom grafe $G = (V, H)$ je **eulerovský**,*ak obsahuje všetky hrany grafu G.*

Poznámka

Pretože ťah je špeciálnym prípadom sledu, je definíciou eulerovského sledu presne vymedzený pojem **eulerovský ťah** *ako taký ťah $t(u, v)$ v súvislom grafe G, ktorý obsahuje všetky hrany grafu G.*
Keďže ťah obsahuje každú hranu grafu G práve raz, postupnosť vrcholov a hrán ťahu $t(u, v)$ predstavuje postup, ako nakresliť diagram grafu G "jedným ťahom".

Definícia

Hovoríme, že graf $G = (V, H)$ je **eulerovský**, *ak v ňom existuje uzavretý eulerovský ťah.*

Eulerovské sledy a eulerovské ťahy

Definícia

Hovoríme, že sled $s(u, v)$ v súvislom grafe $G = (V, H)$ je **eulerovský***,ak obsahuje všetky hrany grafu G.*

Poznámka

Pretože ťah je špeciálnym prípadom sledu, je definíciou eulerovského sledu presne vymedzený pojem **eulerovský ťah** *ako taký ťah $t(u, v)$ v súvislom grafe G, ktorý obsahuje všetky hrany grafu G.*
Keďže ťah obsahuje každú hranu grafu G práve raz, postupnosť vrcholov a hrán ťahu $t(u, v)$ predstavuje postup, ako nakresliť diagram grafu G "jedným ťahom".

Definícia

Hovoríme, že graf $G = (V, H)$ je **eulerovský***, ak v ňom existuje uzavretý eulerovský ťah.*

Eulerovské sledy a eulerovské ťahy

Definícia

Hovoríme, že sled $s(u, v)$ v súvislom grafe $G = (V, H)$ je **eulerovský***,ak obsahuje všetky hrany grafu G.*

Poznámka

Pretože ťah je špeciálnym prípadom sledu, je definíciou eulerovského sledu presne vymedzený pojem **eulerovský ťah** *ako taký ťah $t(u, v)$ v súvislom grafe G, ktorý obsahuje všetky hrany grafu G.*
Keďže ťah obsahuje každú hranu grafu G práve raz, postupnosť vrcholov a hrán ťahu $t(u, v)$ predstavuje postup, ako nakresliť diagram grafu G "jedným ťahom".

Definícia

Hovoríme, že graf $G = (V, H)$ je **eulerovský***, ak v ňom existuje uzavretý eulerovský ťah.*

Eulerova veta o existencii eulerovského ťahu

Veta

(Euler, 1736.) *Súvislý graf* $G = (V, H)$ *je eulerovský práve vtedy, keď stupne všetkých vrcholov grafu* G *sú párne.*

DÔKAZ.

1. Ak v grafe G existuje uzavretý eulerovský ťah $\mathcal{T}$, potom stupeň každého vrchola je párny, pretože počet hrán ktorými ťah $\mathcal{T}$ z každého vrchola v vyšiel sa rovná počtu hrán, ktorými sme do vrchola v vošli.

2. Konštrukciu uzavretého eulerovského ťahu v súvislom grafe, ktorý má všetky vrcholy párneho stupňa popisuje nasledujúci algoritmus:

Eulerova veta o existencii eulerovského ťahu

Veta

(Euler, 1736.) *Súvislý graf* $G = (V, H)$ *je eulerovský práve vtedy, keď stupne všetkých vrcholov grafu* G *sú párne.*

DÔKAZ.

1. Ak v grafe G existuje uzavretý eulerovský ťah $\mathcal{T}$, potom stupeň každého vrchola je párny, pretože počet hrán ktorými ťah $\mathcal{T}$ z každého vrchola v vyšiel sa rovná počtu hrán, ktorými sme do vrchola v vošli.

2. Konštrukciu uzavretého eulerovského ťahu v súvislom grafe, ktorý má všetky vrcholy párneho stupňa popisuje nasledujúci algoritmus:

Eulerova veta o existencii eulerovského ťahu

Veta

(Euler, 1736.) *Súvislý graf $G = (V, H)$ je eulerovský práve vtedy, keď stupne všetkých vrcholov grafu G sú párne.*

Dôkaz.

1. Ak v grafe G existuje uzavretý eulerovský ťah $\mathcal{T}$, potom stupeň každého vrchola je párny, pretože počet hrán ktorými ťah $\mathcal{T}$ z každého vrchola v vyšiel sa rovná počtu hrán, ktorými sme do vrchola v vošli.
2. Konštrukciu uzavretého eulerovského ťahu v súvislom grafe, ktorý má všetky vrcholy párneho stupňa popisuje nasledujúci algoritmus:

Algoritmus na konštrukciu eulerovského ťahu

Algoritmus

- **Krok 1.** *Začni z ľubovoľného vrchola z, polož $\mathcal{T} = (z)$ a postupne predlžuj ťah $\mathcal{T}$ pokiaľ sa dá. Ukončíš vo vrchole z.*
- **Krok 2.** *Nájdi prvý vrchol v ťahu $\mathcal{T}$, ktorý má ešte aspoň jednu hranu nepoužitú v ťahu $\mathcal{T}$.*

  Ak taký vrchol v neexistuje, STOP.

  Ťah $\mathcal{T}$ je hľadaným uzavretým eulerovským ťahom.
- **Krok 3.** *Vytvor ťah $\mathcal{S}$ takto:*

  Polož $\mathcal{S} = (v)$ a postupne predlžuj ťah $\mathcal{S}$ doteraz nepoužitými hranami, pokiaľ sa dá. Ukončíš vo vrchole v.
- **Krok 4.** *Rozdeľ ťah $\mathcal{T}$ na z–v ťah $\mathcal{T}_1$ a v–z ťah $\mathcal{T}_2$, t. j. $\mathcal{T} = \mathcal{T}_1 \oplus \mathcal{T}_2$.*

  Polož $\mathcal{T} = \mathcal{T}_1 \oplus \mathcal{S} \oplus \mathcal{T}_2$.

  GOTO Krok 2.

♣

Algoritmus na konštrukciu eulerovského ťahu

Algoritmus

- **Krok 1.** *Začni z ľubovoľného vrchola z, polož $\mathcal{T} = (z)$ a postupne predlžuj ťah $\mathcal{T}$ pokiaľ sa dá. Ukončíš vo vrchole z.*
- **Krok 2.** *Nájdi prvý vrchol v ťahu $\mathcal{T}$, ktorý má ešte aspoň jednu hranu nepoužitú v ťahu $\mathcal{T}$.*

  Ak taký vrchol v neexistuje, STOP.

  Ťah $\mathcal{T}$ je hľadaným uzavretým eulerovským ťahom.
- **Krok 3.** *Vytvor ťah $\mathcal{S}$ takto:*

  Polož $\mathcal{S} = (v)$ a postupne predlžuj ťah $\mathcal{S}$ doteraz nepoužitými hranami, pokiaľ sa dá. Ukončíš vo vrchole v.
- **Krok 4.** *Rozdeľ ťah $\mathcal{T}$ na z–v ťah $\mathcal{T}_1$ a v–z ťah $\mathcal{T}_2$, t. j. $\mathcal{T} = \mathcal{T}_1 \oplus \mathcal{T}_2$.*

  Polož $\mathcal{T} = \mathcal{T}_1 \oplus \mathcal{S} \oplus \mathcal{T}_2$.

  GOTO Krok 2.

♣

Algoritmus na konštrukciu eulerovského ťahu

Algoritmus

- **Krok 1.** *Začni z ľubovoľného vrcholu z, polož* $\mathcal{T} = (z)$ *a postupne predlžuj ťah* $\mathcal{T}$ *pokiaľ sa dá. Ukončíš vo vrchole* z.
- **Krok 2.** *Nájdi prvý vrchol v ťahu* $\mathcal{T}$, *ktorý má ešte aspoň jednu hranu nepoužitú v ťahu* $\mathcal{T}$.

  Ak taký vrchol v neexistuje, STOP.

  Ťah $\mathcal{T}$ *je hľadaným uzavretým eulerovským ťahom.*
- **Krok 3.** *Vytvor ťah* $\mathcal{S}$ *takto:*

  Polož $\mathcal{S} = (v)$ *a postupne predlžuj ťah* $\mathcal{S}$ *doteraz nepoužitými hranami, pokiaľ sa dá. Ukončíš vo vrchole* v.
- **Krok 4.** *Rozdeľ ťah* $\mathcal{T}$ *na z–v ťah* $\mathcal{T}_1$ *a v–z ťah* $\mathcal{T}_2$, *t. j.* $\mathcal{T} = \mathcal{T}_1 \oplus \mathcal{T}_2$.

  Polož $\mathcal{T} = \mathcal{T}_1 \oplus \mathcal{S} \oplus \mathcal{T}_2$.

  GOTO Krok 2.

♣

Algoritmus na konštrukciu eulerovského ťahu

Algoritmus

- **Krok 1.** *Začni z ľubovoľného vrchola* z*, polož* $\mathcal{T} = (z)$ *a postupne predlžuj ťah* $\mathcal{T}$ *pokiaľ sa dá. Ukončíš vo vrchole* z.
- **Krok 2.** *Nájdi prvý vrchol v ťahu* $\mathcal{T}$*, ktorý má ešte aspoň jednu hranu nepoužitú v ťahu* $\mathcal{T}$.

  Ak taký vrchol v *neexistuje, STOP.*

  Ťah $\mathcal{T}$ *je hľadaným uzavretým eulerovským ťahom.*
- **Krok 3.** *Vytvor ťah* $\mathcal{S}$ *takto:*

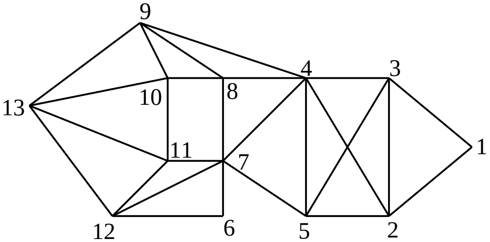
  Polož $\mathcal{S} = (v)$ *a postupne predlžuj ťah* $\mathcal{S}$ *doteraz nepoužitými hranami, pokiaľ sa dá. Ukončíš vo vrchole* v.
- **Krok 4.** *Rozdeľ ťah* $\mathcal{T}$ *na* z*–*v *ťah* $\mathcal{T}_1$ *a* v*–*z *ťah* $\mathcal{T}_2$*, t. j.* $\mathcal{T} = \mathcal{T}_1 \oplus \mathcal{T}_2$.

  Polož $\mathcal{T} = \mathcal{T}_1 \oplus \mathcal{S} \oplus \mathcal{T}_2$.

  GOTO Krok 2.

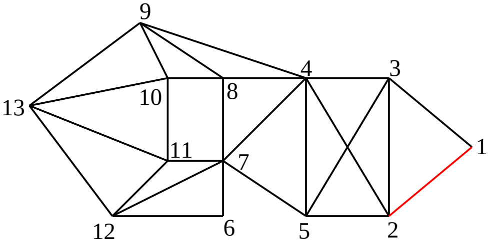
♣

Príklad



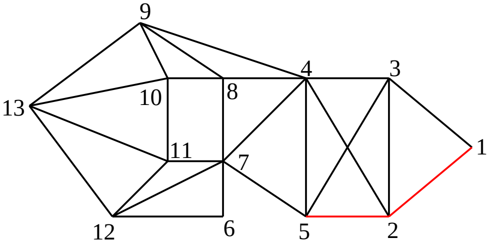
$\mathcal{T} = (1)$

Príklad



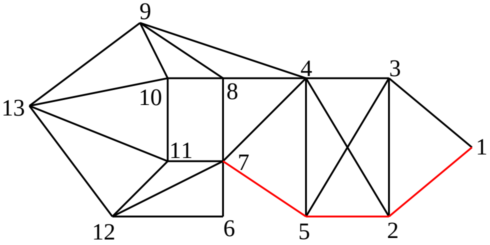
$\mathcal{T} = (1, 2)$

Príklad



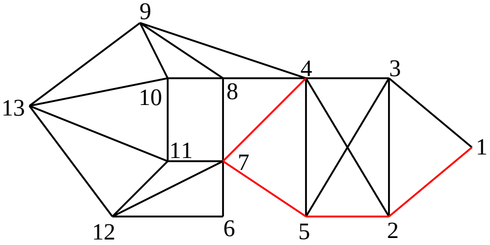
$\mathcal{T} = (1, 2, 5)$

Príklad



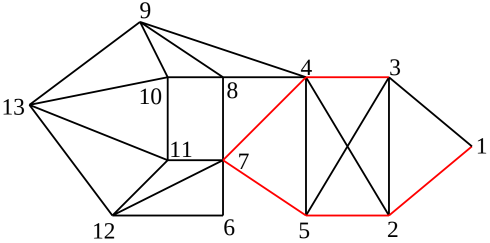
$\mathcal{T} = (1, 2, 5, 7)$

Príklad



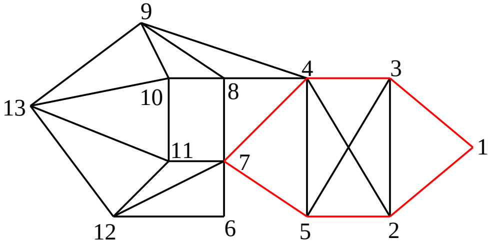
$\mathcal{T} = (1, 2, 5, 7, 4)$

Príklad



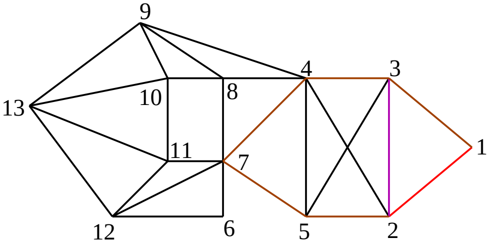
$\mathcal{T} = (1, 2, 5, 7, 4, 3)$

Príklad



$\mathcal{T} = (1, 2, 5, 7, 4, 3, 1)$

Príklad

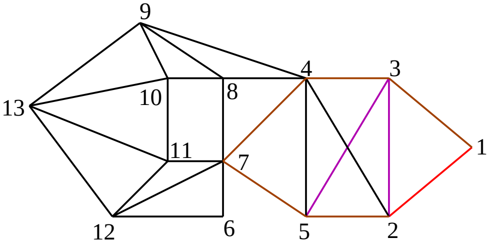


$\mathcal{T} = (1, 2, 5, 7, 4, 3, 1)$

$\mathcal{T}_1 = (1, 2)$, $\mathcal{T}_2 = (2, 5, 7, 4, 3, 1)$

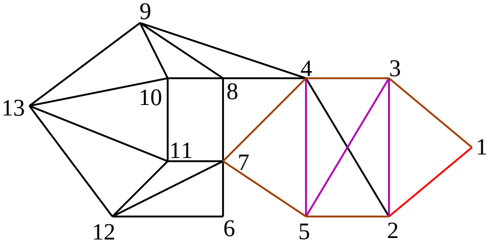
$\mathcal{S} = (2, 3)$

Príklad



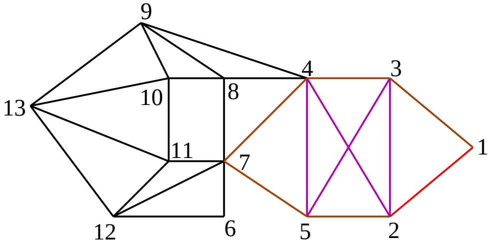
$\mathcal{T} = (1, 2, 5, 7, 4, 3, 1)$
$\mathcal{T}_1 = (1, 2)$, $\mathcal{T}_2 = (2, 5, 7, 4, 3, 1)$
$\mathcal{S} = (2, 3, 5)$

Príklad



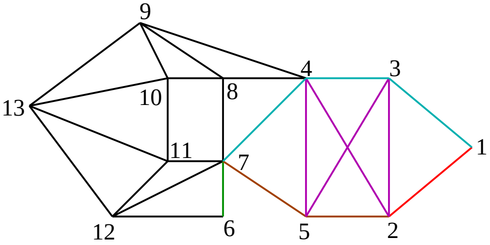
$\mathcal{T} = (1, 2, 5, 7, 4, 3, 1)$
$\mathcal{T}_1 = (1, 2)$, $\mathcal{T}_2 = (2, 5, 7, 4, 3, 1)$
$\mathcal{S} = (2, 3, 5, 4)$

Príklad



$\mathcal{T} = (1, 2, 5, 7, 4, 3, 1)$
$\mathcal{T}_1 = (1, 2)$, $\mathcal{T}_2 = (2, 5, 7, 4, 3, 1)$
$\mathcal{S} = (2, 3, 5, 4, 2)$

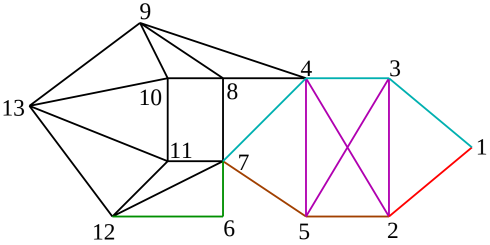
Príklad



$$\mathcal{T} = (1, \underbrace{2, 3, 5, 4, 2}_{\mathcal{S}}, 5, 7, 4, 3, 1)$$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7)$, $\mathcal{T}_2 = (7, 4, 3, 1)$
$\mathcal{S} = (7, 6)$

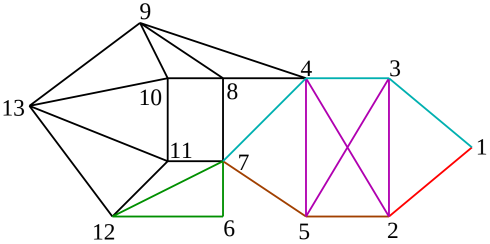
Príklad



$$\mathcal{T} = (1, \underbrace{2, 3, 5, 4, 2}_{\mathcal{S}}, 5, 7, 4, 3, 1)$$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7)$, $\mathcal{T}_2 = (7, 4, 3, 1)$
$\mathcal{S} = (7, 6, 12)$

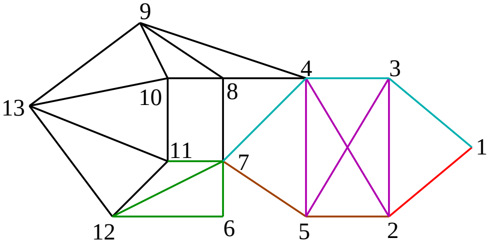
Príklad



$\mathcal{T} = (1, \underbrace{2, 3, 5, 4, 2}_{\mathcal{S}}, 5, 7, 4, 3, 1)$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7)$, $\mathcal{T}_2 = (7, 4, 3, 1)$
$\mathcal{S} = (7, 6, 12, 7)$

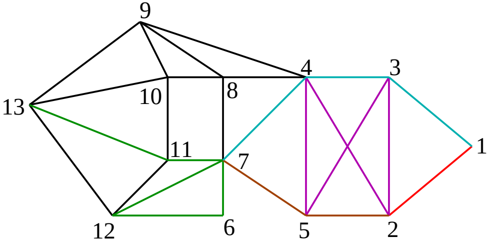
Príklad



$\mathcal{T} = (1, \underbrace{2, 3, 5, 4, 2}_{\mathcal{S}}, 5, 7, 4, 3, 1)$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7)$, $\mathcal{T}_2 = (7, 4, 3, 1)$
$\mathcal{S} = (7, 6, 12, 7, 11)$

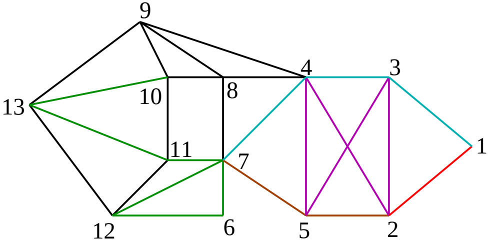
Príklad



$\mathcal{T} = (1, \underbrace{2, 3, 5, 4, 2}_{\mathcal{S}}, 5, 7, 4, 3, 1)$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7)$, $\mathcal{T}_2 = (7, 4, 3, 1)$
$\mathcal{S} = (7, 6, 12, 7, 11, 13)$

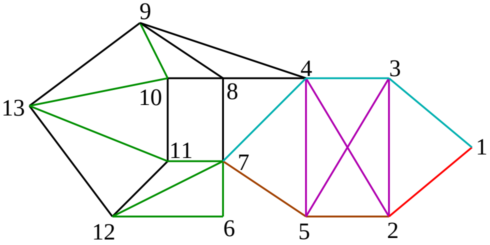
Príklad



$\mathcal{T} = (1, \underbrace{2, 3, 5, 4, 2}_{\mathcal{S}}, 5, 7, 4, 3, 1)$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7)$, $\mathcal{T}_2 = (7, 4, 3, 1)$
$\mathcal{S} = (7, 6, 12, 7, 11, 13, 10)$

Príklad

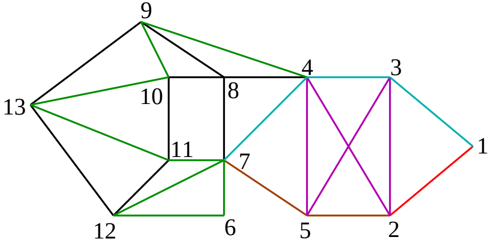


$$\mathcal{T} = (1, \underbrace{2, 3, 5, 4, 2}_{\mathcal{S}}, 5, 7, 4, 3, 1)$$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7)$, $\mathcal{T}_2 = (7, 4, 3, 1)$

$\mathcal{S} = (7, 6, 12, 7, 11, 13, 10, 9)$

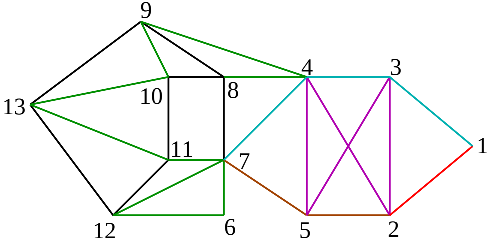
Príklad



$$\mathcal{T} = (1, \underbrace{2, 3, 5, 4, 2}_{\mathcal{S}}, 5, 7, 4, 3, 1)$$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7)$, $\mathcal{T}_2 = (7, 4, 3, 1)$
$\mathcal{S} = (7, 6, 12, 7, 11, 13, 10, 9, 4)$

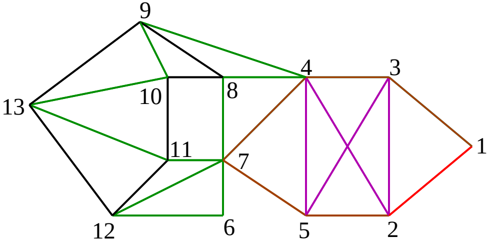
Príklad



$\mathcal{T} = (1, \underbrace{2, 3, 5, 4, 2}_{\mathcal{S}}, 5, 7, 4, 3, 1)$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7)$, $\mathcal{T}_2 = (7, 4, 3, 1)$
$\mathcal{S} = (7, 6, 12, 7, 11, 13, 10, 9, 4, 8)$

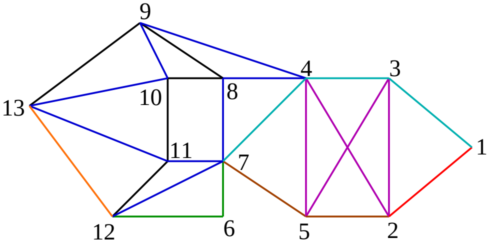
Príklad



$\mathcal{T} = (1, \underbrace{2, 3, 5, 4, 2}_{\mathcal{S}}, 5, 7, 4, 3, 1)$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7)$, $\mathcal{T}_2 = (7, 4, 3, 1)$
$\mathcal{S} = (7, 6, 12, 7, 11, 13, 10, 9, 4, 8, 7)$

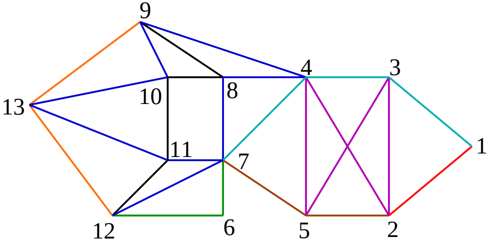
Príklad



$$\mathcal{T} = (1, 2, 3, 5, 4, 2, 5, \underbrace{7, 6, 12, 7, 11, 13, 10, 9, 4, 8, 7}_{\mathcal{S}}, 4, 3, 1)$$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7, 6, 12)$, $\mathcal{T}_2 = (12, 7, 11, 13, 10, 9, 4, 8, 7, 4, 3, 1)$
$\mathcal{S} = (12, 13)$

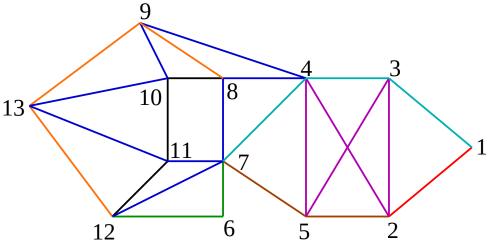
Príklad



$$\mathcal{T} = (1, 2, 3, 5, 4, 2, 5, \underbrace{7, 6, 12, 7, 11, 13, 10, 9, 4, 8, 7}_{\mathcal{S}}, 4, 3, 1)$$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7, 6, 12)$, $\mathcal{T}_2 = (12, 7, 11, 13, 10, 9, 4, 8, 7, 4, 3, 1)$
$\mathcal{S} = (12, 13, 9)$

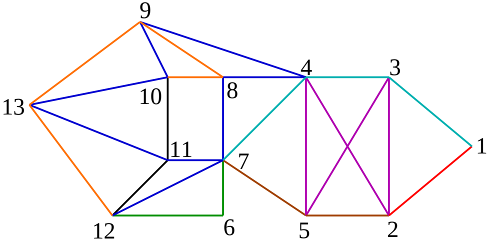
Príklad



$$\mathcal{T} = (1,2,3,5,4,2,5,\underbrace{7,6,12,7,11,13,10,9,4,8,7}_{\mathcal{S}},4,3,1)$$

$\mathcal{T}_1 = (1,2,3,5,4,2,5,7,6,12)$, $\mathcal{T}_2 = (12,7,11,13,10,9,4,8,7,4,3,1)$
$\mathcal{S} = (12,13,9,8)$

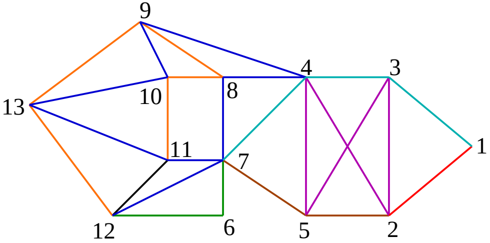
Príklad



$$\mathcal{T} = (1, 2, 3, 5, 4, 2, 5, \underbrace{7, 6, 12, 7, 11, 13, 10, 9, 4, 8, 7}_{\mathcal{S}}, 4, 3, 1)$$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7, 6, 12)$, $\mathcal{T}_2 = (12, 7, 11, 13, 10, 9, 4, 8, 7, 4, 3, 1)$
$\mathcal{S} = (12, 13, 9, 8, 10)$

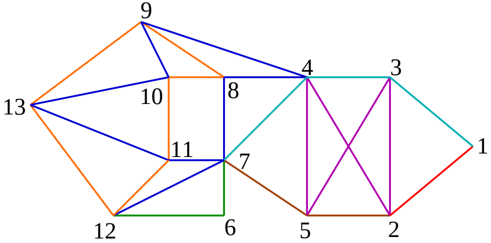
Príklad



$$\mathcal{T} = (1, 2, 3, 5, 4, 2, 5, \underbrace{7, 6, 12, 7, 11, 13, 10, 9, 4, 8, 7}_{\mathcal{S}}, 4, 3, 1)$$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7, 6, 12)$, $\mathcal{T}_2 = (12, 7, 11, 13, 10, 9, 4, 8, 7, 4, 3, 1)$
$\mathcal{S} = (12, 13, 9, 8, 10, 11)$

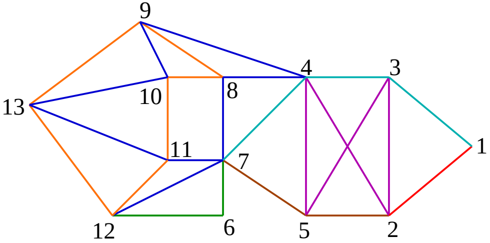
Príklad



$$\mathcal{T} = (1, 2, 3, 5, 4, 2, 5, \underbrace{7, 6, 12, 7, 11, 13, 10, 9, 4, 8, 7}_{\mathcal{S}}, 4, 3, 1)$$

$\mathcal{T}_1 = (1, 2, 3, 5, 4, 2, 5, 7, 6, 12)$, $\mathcal{T}_2 = (12, 7, 11, 13, 10, 9, 4, 8, 7, 4, 3, 1)$
$\mathcal{S} = (12, 13, 9, 8, 10, 11, 12)$

Príklad



$$\mathcal{T} = (1,2,3,5,4,2,5,7,6,\underbrace{12,13,9,8,10,11,12}_{\mathcal{S}},7,11,13,10,9,4,8,7,4,3,1)$$

Fleuryho algoritmus

Algoritmus

Fleuryho algoritmus na hľadanie uzavretého eulerovského ťahu v súvislom grafe $G = (V, H)$, v ktorom majú všetky vrcholy párny stupeň.

- **Krok 1.** *Začni v ľubovoľnom vrchole a do ťahu $\mathcal{T}$ zaraď ľubovoľnú s ním incidentnú hranu.*
- **Krok 2.** *Ak sú do ťahu $\mathcal{T}$ zaradené všetky hrany grafu G, STOP.*
- **Krok 3.** *Ako ďalšiu hranu zaraď do ťahu $\mathcal{T}$ takú hranu incidentnú s jeho posledným vrcholom, po vybratí ktorej sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na*
  - *dva netriviálne komponenty*
  - *netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$.*
- *GOTO krok 2.*

♣

Fleuryho algoritmus

Algoritmus

Fleuryho algoritmus na hľadanie uzavretého eulerovského ťahu v súvislom grafe $G = (V, H)$, v ktorom majú všetky vrcholy párny stupeň.

- **Krok 1.** *Začni v ľubovoľnom vrchole a do ťahu $\mathcal{T}$ zaraď ľubovoľnú s ním incidentnú hranu.*
- **Krok 2.** *Ak sú do ťahu $\mathcal{T}$ zaradené všetky hrany grafu G, STOP.*
- **Krok 3.** *Ako ďalšiu hranu zaraď do ťahu $\mathcal{T}$ takú hranu incidentnú s jeho posledným vrcholom, po vybratí ktorej sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na*
  - *dva netriviálne komponenty*
  - *netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$.*
- *GOTO krok 2.*

♣

Fleuryho algoritmus

Algoritmus

Fleuryho algoritmus na hľadanie uzavretého eulerovského ťahu v súvislom grafe $G = (V, H)$, v ktorom majú všetky vrcholy párny stupeň.

- **Krok 1.** *Začni v ľubovoľnom vrchole a do ťahu $\mathcal{T}$ zaraď ľubovoľnú s ním incidentnú hranu.*
- **Krok 2.** *Ak sú do ťahu $\mathcal{T}$ zaradené všetky hrany grafu G, STOP.*
- **Krok 3.** *Ako ďalšiu hranu zaraď do ťahu $\mathcal{T}$ takú hranu incidentnú s jeho posledným vrcholom, po vybratí ktorej sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na*
  - *dva netriviálne komponenty*
  - *netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$.*
- *GOTO krok 2.*

♣

Fleuryho algoritmus

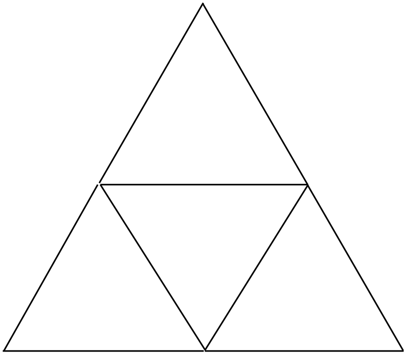
Algoritmus

Fleuryho algoritmus na hľadanie uzavretého eulerovského ťahu v súvislom grafe $G = (V, H)$, v ktorom majú všetky vrcholy párny stupeň.

- **Krok 1.** *Začni v ľubovoľnom vrchole a do ťahu $\mathcal{T}$ zaraď ľubovoľnú s ním incidentnú hranu.*
- **Krok 2.** *Ak sú do ťahu $\mathcal{T}$ zaradené všetky hrany grafu G, STOP.*
- **Krok 3.** *Ako ďalšiu hranu zaraď do ťahu $\mathcal{T}$ takú hranu incidentnú s jeho posledným vrcholom, po vybratí ktorej sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na*
    - *dva netriviálne komponenty*
    - *netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$.*
- *GOTO krok 2.*

♣

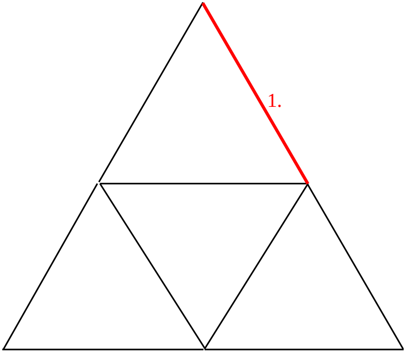
Fleuryho algoritmus – príklad



Poznámka

Kontrola, či sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na dva netriviálne komponenty alebo netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$, je pri ručnom kreslení diagramu jedným ťahom intuitívna. Pri počítačovej realizácii by bolo potrebné túto kontrolu algoritmicky vyriešiť.

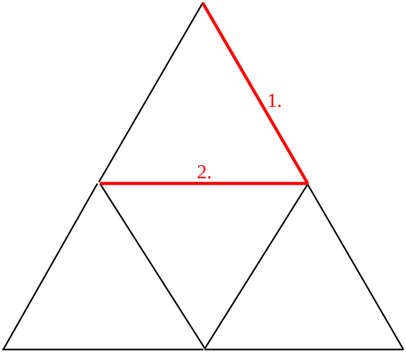
Fleuryho algoritmus – príklad



Poznámka

Kontrola, či sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na dva netriviálne komponenty alebo netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$, je pri ručnom kreslení diagramu jedným ťahom intuitívna. Pri počítačovej realizácii by bolo potrebné túto kontrolu algoritmicky vyriešiť.

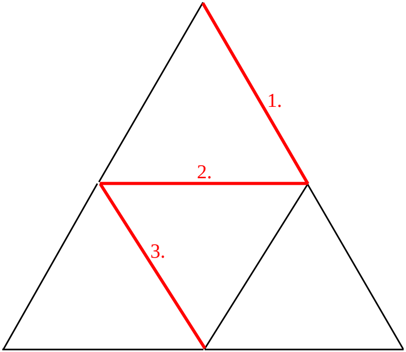
Fleuryho algoritmus – príklad



Poznámka

Kontrola, či sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na dva netriviálne komponenty alebo netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$, je pri ručnom kreslení diagramu jedným ťahom intuitívna. Pri počítačovej realizácii by bolo potrebné túto kontrolu algoritmicky vyriešiť.

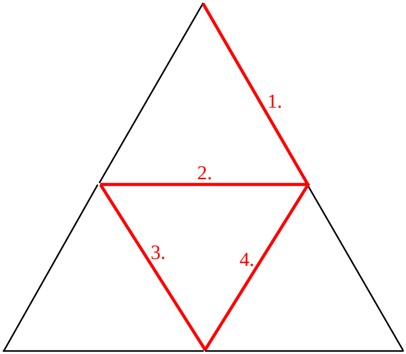
Fleuryho algoritmus – príklad



Poznámka

Kontrola, či sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na dva netriviálne komponenty alebo netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$, je pri ručnom kreslení diagramu jedným ťahom intuitívna. Pri počítačovej realizácii by bolo potrebné túto kontrolu algoritmicky vyriešiť.

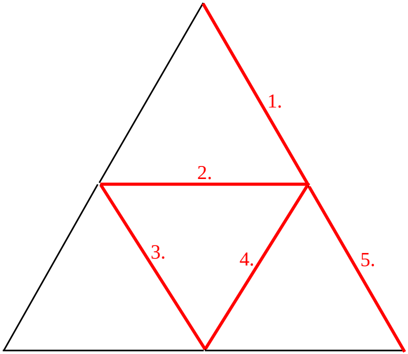
Fleuryho algoritmus – príklad



Poznámka

Kontrola, či sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na dva netriviálne komponenty alebo netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$, je pri ručnom kreslení diagramu jedným ťahom intuitívna. Pri počítačovej realizácii by bolo potrebné túto kontrolu algoritmicky vyriešiť.

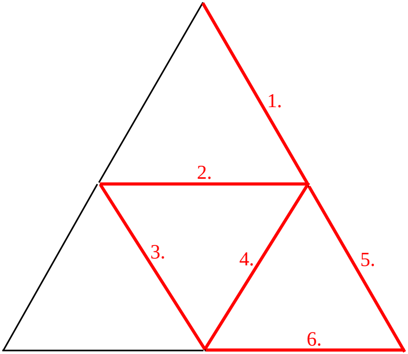
Fleuryho algoritmus – príklad



Poznámka

Kontrola, či sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na dva netriviálne komponenty alebo netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$, je pri ručnom kreslení diagramu jedným ťahom intuitívna. Pri počítačovej realizácii by bolo potrebné túto kontrolu algoritmicky vyriešiť.

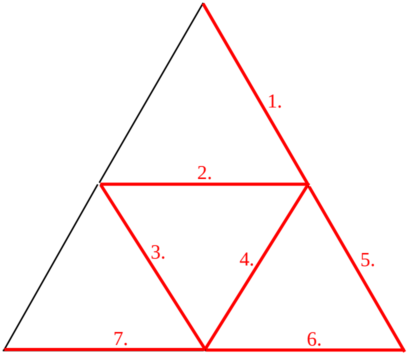
Fleuryho algoritmus – príklad



Poznámka

Kontrola, či sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na dva netriviálne komponenty alebo netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$, je pri ručnom kreslení diagramu jedným ťahom intuitívna. Pri počítačovej realizácii by bolo potrebné túto kontrolu algoritmicky vyriešiť.

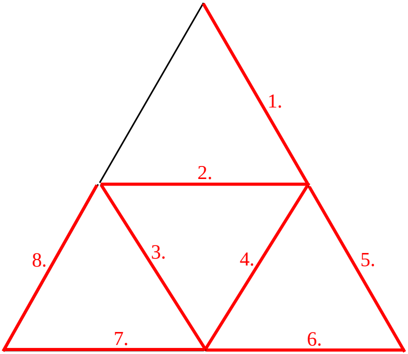
Fleuryho algoritmus – príklad



Poznámka

Kontrola, či sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na dva netriviálne komponenty alebo netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$, je pri ručnom kreslení diagramu jedným ťahom intuitívna. Pri počítačovej realizácii by bolo potrebné túto kontrolu algoritmicky vyriešiť.

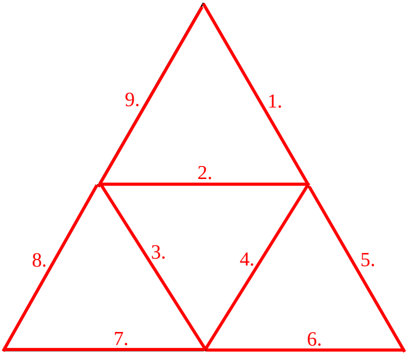
Fleuryho algoritmus – príklad



Poznámka

Kontrola, či sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na dva netriviálne komponenty alebo netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$, je pri ručnom kreslení diagramu jedným ťahom intuitívna. Pri počítačovej realizácii by bolo potrebné túto kontrolu algoritmicky vyriešiť.

Fleuryho algoritmus – príklad



Poznámka

Kontrola, či sa podgraf grafu G pozostávajúci z nevybratých hrán a s nimi incidentných vrcholov nerozpadne na dva netriviálne komponenty alebo netriviálny komponent a izolovaný začiatok ťahu $\mathcal{T}$, je pri ručnom kreslení diagramu jedným ťahom intuitívna. Pri počítačovej realizácii by bolo potrebné túto kontrolu algoritmicky vyriešiť.

Labyrintový algoritmus

Algoritmus

Labyrintový algoritmus na hľadanie uzavretého eulerovského ťahu v súvislom grafe $G = (V, H)$, v ktorom majú všetky vrcholy párny stupeň.

- **Krok 1.** *Začni z ľubovoľného vrchola* $u \in V$.

  Nech sled $\mathcal{S}$ *inicializačne pozostáva z jediného vrchola* u.

  Polož $w := u$ *– vrchol* w *je posledný vrchol doteraz vytvoreného sledu* $\mathcal{S}$.

Labyrintový algoritmus – pokračovanie

Algoritmus (– pokračovanie)

- **Krok 2.** *Ako ďalšiu hranu vyber podľa nižšie uvedených pravidiel do sledu $\mathcal{S}$ hranu $\{w, v\}$. Zaznač si smer použitia hrany $\{w, v\}$.*

  Ak doteraz vrchol v ešte nebol zaradený do sledu $\mathcal{S}$, označ hranu $\{w, v\}$ ako hranu prvého príchodu.

  Ďalej zaznamenaj tzv. **spätnú postupnosť** *— poradie hrán, v ktorom sa v slede $\mathcal{S}$ vyskytujú po druhýkrát.*

  Pri výbere hrany dodržuj nasledujúce pravidlá:

(L1): *Každú hranu možno v jednom smere použiť iba raz*
(L2): *Poradie zaraďovania hrán:*
  – nepoužité hrany
  – hrany použité raz
  – hrana prvého príchodu (ak niet inej možnosti)

- **Krok 3.** *Ak taká hrana neexistuje – STOP. Spätná postupnosť určuje hľadaný eulerovský ťah.*

- **Krok 4.** *Inak polož $w := v$ a pokračuj krokom 2.*

Labyrintový algoritmus – pokračovanie

Algoritmus (– pokračovanie)

- **Krok 2.** *Ako ďalšiu hranu vyber podľa nižšie uvedených pravidiel do sledu $\mathcal{S}$ hranu $\{w, v\}$. Zaznač si smer použitia hrany $\{w, v\}$.*

  Ak doteraz vrchol v ešte nebol zaradený do sledu $\mathcal{S}$, označ hranu $\{w, v\}$ ako hranu prvého príchodu.

  Ďalej zaznamenaj tzv. **spätnú postupnosť** *— poradie hrán, v ktorom sa v slede $\mathcal{S}$ vyskytujú po druhýkrát.*

  Pri výbere hrany dodržuj nasledujúce pravidlá:

  (L1): *Každú hranu možno v jednom smere použiť iba raz*
  (L2): *Poradie zaraďovania hrán:*
  – *nepoužité hrany*
  – *hrany použité raz*
  – *hrana prvého príchodu (ak niet inej možnosti)*

- **Krok 3.** *Ak taká hrana neexistuje – STOP.*
  Spätná postupnosť určuje hľadaný eulerovský ťah.

- **Krok 4.** *Inak polož $w := v$ a pokračuj krokom 2.*

Labyrintový algoritmus – pokračovanie

Algoritmus (– pokračovanie)

- **Krok 2.** *Ako ďalšiu hranu vyber podľa nižšie uvedených pravidiel do sledu $\mathcal{S}$ hranu $\{w, v\}$. Zaznač si smer použitia hrany $\{w, v\}$.*

  Ak doteraz vrchol v ešte nebol zaradený do sledu $\mathcal{S}$, označ hranu $\{w, v\}$ ako hranu prvého príchodu.

  Ďalej zaznamenaj tzv. **spätnú postupnosť** *— poradie hrán, v ktorom sa v slede $\mathcal{S}$ vyskytujú po druhýkrát.*

  Pri výbere hrany dodržuj nasledujúce pravidlá:

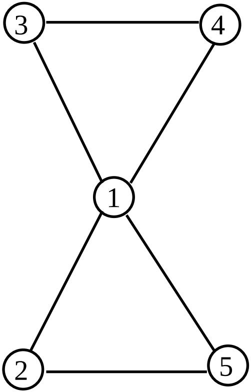
  (L1): *Každú hranu možno v jednom smere použiť iba raz*
  (L2): *Poradie zaraďovania hrán:*
  – nepoužité hrany
  – hrany použité raz
  – hrana prvého príchodu (ak niet inej možnosti)

- **Krok 3.** *Ak taká hrana neexistuje – STOP.*
  Spätná postupnosť určuje hľadaný eulerovský ťah.

- **Krok 4.** *Inak polož $w := v$ a pokračuj krokom 2.*

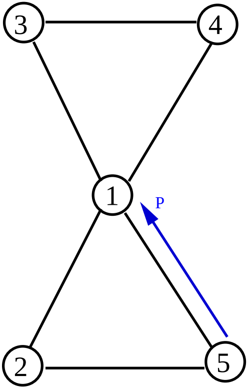
Labyrintový algoritmus – príklad

Hrany	smer použitia hrany						navštívený vrchol				
sledu $\mathcal{S}$	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				$\Leftarrow$			●				
$\{1,2\}$	$\Rightarrow$							●			
$\{2,5\}$					$\rightarrow$						
$\boldsymbol{\{5,2\}}$					$\leftarrow$						
$\boldsymbol{\{2,1\}}$	$\leftarrow$										
$\{1,3\}$		$\Rightarrow$							●		
$\{3,4\}$						$\Rightarrow$				●	
$\{4,1\}$			$\leftarrow$								
$\boldsymbol{\{1,4\}}$			$\rightarrow$								
$\boldsymbol{\{4,3\}}$						$\leftarrow$					
$\boldsymbol{\{3,1\}}$		$\leftarrow$									
$\boldsymbol{\{1,5\}}$				$\rightarrow$							



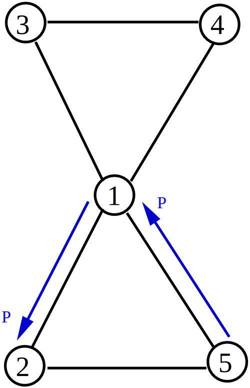
Labyrintový algoritmus – príklad

Hrany sledu $\mathcal{S}$	smer použitia hrany						navštívený vrchol				
	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				$\Leftarrow$			●				
$\{1,2\}$	$\Rightarrow$							●			
$\{2,5\}$					$\rightarrow$						
$\boldsymbol{\{5,2\}}$					$\leftarrow$						
$\boldsymbol{\{2,1\}}$	$\leftarrow$										
$\{1,3\}$		$\Rightarrow$							●		
$\{3,4\}$						$\Rightarrow$				●	
$\{4,1\}$			$\leftarrow$								
$\boldsymbol{\{1,4\}}$			$\rightarrow$								
$\boldsymbol{\{4,3\}}$						$\leftarrow$					
$\boldsymbol{\{3,1\}}$		$\leftarrow$									
$\boldsymbol{\{1,5\}}$				$\rightarrow$							



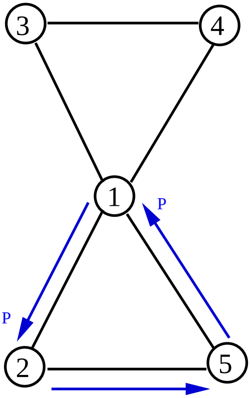
Labyrintový algoritmus – príklad

Hrany sledu $\mathcal{S}$	smer použitia hrany						navštívený vrchol				
	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				$\Leftarrow$			●				
$\{1,2\}$	$\Rightarrow$							●			
$\{2,5\}$					$\rightarrow$						
$\{5,2\}$					$\leftarrow$						
$\{2,1\}$	$\leftarrow$										
$\{1,3\}$		$\Rightarrow$							●		
$\{3,4\}$						$\Rightarrow$				●	
$\{4,1\}$			$\leftarrow$								
$\{1,4\}$			$\rightarrow$								
$\{4,3\}$						$\leftarrow$					
$\{3,1\}$		$\leftarrow$									
$\{1,5\}$				$\rightarrow$							



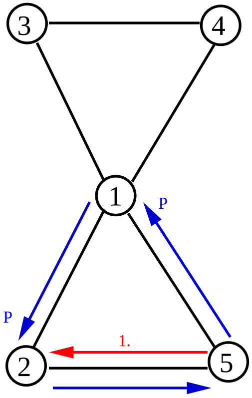
Labyrintový algoritmus – príklad

Hrany	smer použitia hrany						navštívený vrchol				
sledu $\mathcal{S}$	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											•
$\{5,1\}$				⇐			•				
$\{1,2\}$	⇒							•			
$\{2,5\}$					→						
$\{5,2\}$					←						
$\{2,1\}$	←										
$\{1,3\}$		⇒							•		
$\{3,4\}$						⇒				•	
$\{4,1\}$			←								
$\{1,4\}$			→								
$\{4,3\}$						←					
$\{3,1\}$		←									
$\{1,5\}$				→							



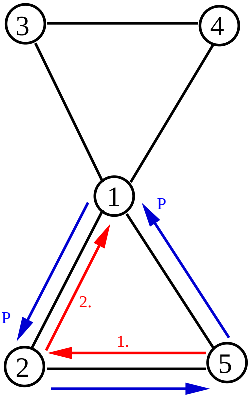
Labyrintový algoritmus – príklad

Hrany sledu $\mathcal{S}$	smer použitia hrany						navštívený vrchol				
	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				$\Leftarrow$			●				
$\{1,2\}$	$\Rightarrow$							●			
$\{2,5\}$					$\rightarrow$						
$\boldsymbol{\{5,2\}}$					$\leftarrow$						
$\boldsymbol{\{2,1\}}$	$\leftarrow$										
$\{1,3\}$		$\Rightarrow$							●		
$\{3,4\}$						$\Rightarrow$				●	
$\{4,1\}$			$\leftarrow$								
$\boldsymbol{\{1,4\}}$			$\rightarrow$								
$\boldsymbol{\{4,3\}}$						$\leftarrow$					
$\boldsymbol{\{3,1\}}$		$\leftarrow$									
$\boldsymbol{\{1,5\}}$				$\rightarrow$							



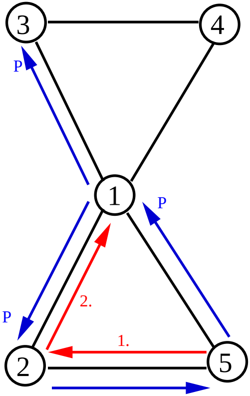
Labyrintový algoritmus – príklad

Hrany sledu $\mathcal{S}$	smer použitia hrany						navštívený vrchol				
	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				$\Leftarrow$			●				
$\{1,2\}$	$\Rightarrow$							●			
$\{2,5\}$					$\rightarrow$						
$\mathbf{\{5,2\}}$					$\leftarrow$						
$\mathbf{\{2,1\}}$	$\leftarrow$										
$\{1,3\}$		$\Rightarrow$							●		
$\{3,4\}$						$\Rightarrow$				●	
$\{4,1\}$			$\leftarrow$								
$\mathbf{\{1,4\}}$			$\rightarrow$								
$\mathbf{\{4,3\}}$						$\leftarrow$					
$\mathbf{\{3,1\}}$		$\leftarrow$									
$\mathbf{\{1,5\}}$				$\rightarrow$							



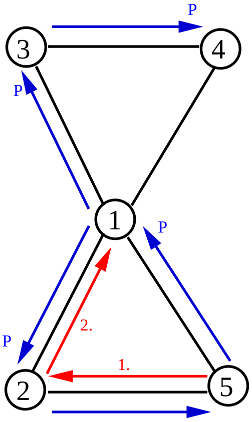
Labyrintový algoritmus – príklad

Hrany sledu $\mathcal{S}$	smer použitia hrany						navštívený vrchol				
	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				⇐			●				
$\{1,2\}$	⇒							●			
$\{2,5\}$					→						
$\{5,2\}$					←						
$\{2,1\}$	←										
$\{1,3\}$		⇒							●		
$\{3,4\}$						⇒				●	
$\{4,1\}$			←								
$\{1,4\}$			→								
$\{4,3\}$						←					
$\{3,1\}$		←									
$\{1,5\}$				→							



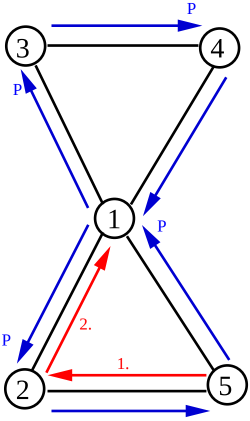
Labyrintový algoritmus – príklad

Hrany sledu $\mathcal{S}$	smer použitia hrany						navštívený vrchol				
	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				$\Leftarrow$			●				
$\{1,2\}$	$\Rightarrow$							●			
$\{2,5\}$					$\rightarrow$						
$\boldsymbol{\{5,2\}}$					$\leftarrow$						
$\boldsymbol{\{2,1\}}$	$\leftarrow$										
$\{1,3\}$		$\Rightarrow$							●		
$\{3,4\}$						$\Rightarrow$				●	
$\{4,1\}$			$\leftarrow$								
$\boldsymbol{\{1,4\}}$			$\rightarrow$								
$\boldsymbol{\{4,3\}}$						$\leftarrow$					
$\boldsymbol{\{3,1\}}$		$\leftarrow$									
$\boldsymbol{\{1,5\}}$				$\rightarrow$							



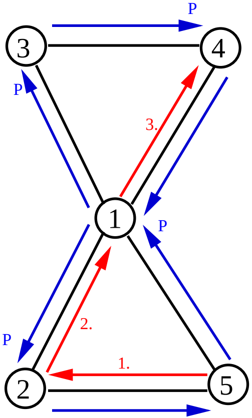
Labyrintový algoritmus – príklad

Hrany	smer použitia hrany						navštívený vrchol				
sledu $\mathcal{S}$	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5, 1\}$				⇐			●				
$\{1, 2\}$	⇒							●			
$\{2, 5\}$					→						
$\{5, 2\}$					←						
$\{2, 1\}$	←										
$\{1, 3\}$		⇒							●		
$\{3, 4\}$						⇒				●	
$\{4, 1\}$			←								
$\{1, 4\}$			→								
$\{4, 3\}$						←					
$\{3, 1\}$		←									
$\{1, 5\}$				→							



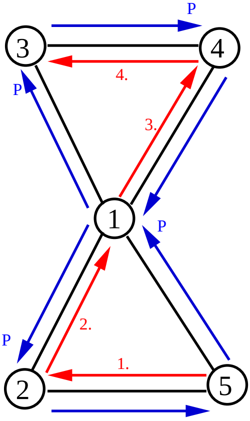
Labyrintový algoritmus – príklad

Hrany	smer použitia hrany						navštívený vrchol				
sledu $\mathcal{S}$	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				⇐			●				
$\{1,2\}$	⇒							●			
$\{2,5\}$					→						
$\{5,2\}$					←						
$\{2,1\}$	←										
$\{1,3\}$		⇒							●		
$\{3,4\}$						⇒				●	
$\{4,1\}$			←								
$\{1,4\}$			→								
$\{4,3\}$						←					
$\{3,1\}$		←									
$\{1,5\}$				→							



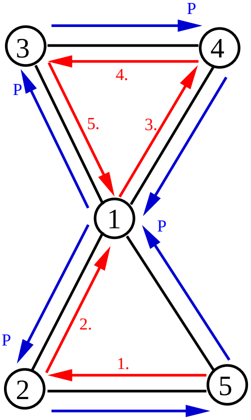
Labyrintový algoritmus – príklad

Hrany	smer použitia hrany						navštívený vrchol				
sledu $\mathcal{S}$	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				$\Leftarrow$			●				
$\{1,2\}$	$\Rightarrow$							●			
$\{2,5\}$					$\rightarrow$						
$\boldsymbol{\{5,2\}}$					$\leftarrow$						
$\boldsymbol{\{2,1\}}$	$\leftarrow$										
$\{1,3\}$		$\Rightarrow$							●		
$\{3,4\}$						$\Rightarrow$				●	
$\{4,1\}$			$\leftarrow$								
$\boldsymbol{\{1,4\}}$			$\rightarrow$								
$\boldsymbol{\{4,3\}}$						$\leftarrow$					
$\boldsymbol{\{3,1\}}$		$\leftarrow$									
$\boldsymbol{\{1,5\}}$				$\rightarrow$							



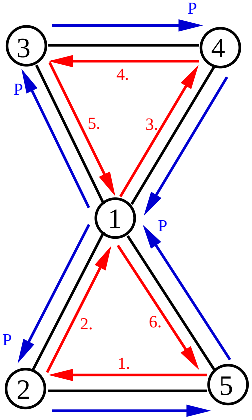
Labyrintový algoritmus – príklad

Hrany	smer použitia hrany						navštívený vrchol				
sledu $\mathcal{S}$	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				$\Leftarrow$			●				
$\{1,2\}$	$\Rightarrow$							●			
$\{2,5\}$					$\rightarrow$						
$\{5,2\}$					$\leftarrow$						
$\{2,1\}$	$\leftarrow$										
$\{1,3\}$		$\Rightarrow$							●		
$\{3,4\}$						$\Rightarrow$				●	
$\{4,1\}$			$\leftarrow$								
$\{1,4\}$			$\rightarrow$								
$\{4,3\}$						$\leftarrow$					
$\{3,1\}$		$\leftarrow$									
$\{1,5\}$				$\rightarrow$							



Labyrintový algoritmus – príklad

Hrany sledu $\mathcal{S}$	smer použitia hrany						navštívený vrchol				
	{1,2}	{1,3}	{1,4}	{1,5}	{2,5}	{3,4}	1	2	3	4	5
-											●
$\{5,1\}$				⇐			●				
$\{1,2\}$	⇒							●			
$\{2,5\}$					→						
$\boldsymbol{\{5,2\}}$					←						
$\boldsymbol{\{2,1\}}$	←										
$\{1,3\}$		⇒							●		
$\{3,4\}$						⇒				●	
$\{4,1\}$			←								
$\boldsymbol{\{1,4\}}$			→								
$\boldsymbol{\{4,3\}}$						←					
$\boldsymbol{\{3,1\}}$		←									
$\boldsymbol{\{1,5\}}$				→							



Úloha čínskeho poštára

Chinese postman problem

Slovná formulácia úlohy čínskeho poštára:
Poštár má vyjsť z pošty, prejsť všetky ulice svojho rajónu a vrátiť sa na poštu tak, aby sa čo najmenej nachodil.

Matematická formulácia úlohy čínskeho poštára.
V súvislom hranovo ohodnotenom grafe nájsť uzavretý eulerovský sled najmenšej dĺžky.

Úloha čínskeho poštára

Poznámka

- *Model cestnej siete poštára – súvislý hranovo ohodnotený graf* $G = (V, H, c)$.
- *Keby mal graf* G *všetky vrcholy párneho stupňa, stačilo by nájsť v* G *uzavretý eulerovský ťah.*
- *Ak má graf* G *vrcholy nepárneho stupňa, je ich* $2t$ *(párny počet).*
- *Pridaním fiktívnych hrán typu* $\{\text{nepárny}, \text{nepárny}\}$ *s dĺžkou rovnajúcou sa vzdialenosti príslušných vrcholov v* G *možno z* G *vyrobiť eulerovský graf alebo multigraf.*
- *Uzavretý eulerovský ťah v rozšírenom grafe predstavuje trasu poštára, pričom fiktívne hrany predstavujú najkratšie cesty medzi ich koncovými vrcholmi a tieto cesty poštár prejde naprázdno – bez roznášania pošty.*
- **Čím meší súčet dĺžok pridaných fiktívnych hrán, tým lepšie výsledné riešenie.**

Úloha čínskeho poštára

Definícia

Nech $G = (V, H, c)$ je hranovo ohodnotený graf.

Párenie *v grafe G je taký jeho podgraf P, v ktorom má každý vrchol stupeň 1.*

Cena párenia *P je súčet ohodnotení jeho hrán.*

Hovoríme, že párenie P je **maximálne párenie** *v grafe G, ak P nie je podgrafom žiadneho iného párenia v G.*

Párenie P je **najpočetnejšie párenie** *v grafe G ak P má zo všetkých párení najväčší počet hrán.*

Párenie P je **úplné párenie** *v G, ak P je faktorovým podgrafom grafu G (P obsahuje všetky vrcholy grafu G).*

a) Maximálne párenie, ktoré nie je ani najpočetnejšie, ani úplné.

b) Najpočetnejšie párenie, ktoré nie je úplné.

c) Úplné párenie v K_6.

Úloha čínskeho poštára

Definícia

Nech $G = (V, H, c)$ je hranovo ohodnotený graf.

Párenie *v grafe G je taký jeho podgraf P, v ktorom má každý vrchol stupeň 1.*

Cena párenia *P je súčet ohodnotení jeho hrán.*

Hovoríme, že párenie P je **maximálne párenie** *v grafe G, ak P nie je podgrafom žiadneho iného párenia v G.*

Párenie P je **najpočetnejšie párenie** *v grafe G ak P má zo všetkých párení najväčší počet hrán.*

Párenie P je **úplné párenie** *v G, ak P je faktorovým podgrafom grafu G (P obsahuje všetky vrcholy grafu G).*

a) Maximálne párenie, ktoré nie je ani najpočetnejšie, ani úplné.

b) Najpočetnejšie párenie, ktoré nie je úplné.

c) Úplné párenie v K_6.

Úloha čínskeho poštára

Definícia

Nech $G = (V, H, c)$ je hranovo ohodnotený graf.
Párenie *v grafe G je taký jeho podgraf P, v ktorom má každý vrchol stupeň 1.*
Cena párenia *P je súčet ohodnotení jeho hrán.*
Hovoríme, že párenie P je **maximálne párenie** *v grafe G, ak P nie je podgrafom žiadneho iného párenia v G.*
Párenie P je **najpočetnejšie párenie** *v grafe G ak P má zo všetkých párení najväčší počet hrán.*
Párenie P je **úplné párenie** *v G, ak P je faktorovým podgrafom grafu G (P obsahuje všetky vrcholy grafu G).*

a) Maximálne párenie, ktoré nie je ani najpočetnejšie, ani úplné.

b) Najpočetnejšie párenie, ktoré nie je úplné.

c) Úplné párenie v K_6.

Úloha čínskeho poštára

Definícia

Nech $G = (V, H, c)$ je hranovo ohodnotený graf.

Párenie *v grafe G je taký jeho podgraf P, v ktorom má každý vrchol stupeň 1.*

Cena párenia *P je súčet ohodnotení jeho hrán.*

Hovoríme, že párenie P je **maximálne párenie** *v grafe G, ak P nie je podgrafom žiadneho iného párenia v G.*

Párenie P je **najpočetnejšie párenie** *v grafe G ak P má zo všetkých párení najväčší počet hrán.*

Párenie P je **úplné párenie** *v G, ak P je faktorovým podgrafom grafu G (P obsahuje všetky vrcholy grafu G).*

a) Maximálne párenie, ktoré nie je ani najpočetnejšie, ani úplné.

b) Najpočetnejšie párenie, ktoré nie je úplné.

c) Úplné párenie v K_6.

Úloha čínskeho poštára

Definícia

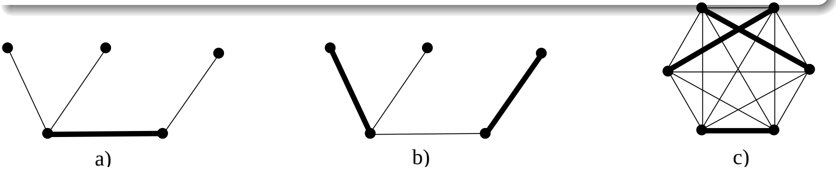
Nech $G = (V, H, c)$ je hranovo ohodnotený graf.
Párenie *v grafe G je taký jeho podgraf P, v ktorom má každý vrchol stupeň 1.*
Cena párenia *P je súčet ohodnotení jeho hrán.*
Hovoríme, že párenie P je **maximálne párenie** *v grafe G, ak P nie je podgrafom žiadneho iného párenia v G.*
Párenie P je **najpočetnejšie párenie** *v grafe G ak P má zo všetkých párení najväčší počet hrán.*
Párenie P je **úplné párenie** *v G, ak P je faktorovým podgrafom grafu G (P obsahuje všetky vrcholy grafu G).*

a) Maximálne párenie, ktoré nie je ani najpočetnejšie, ani úplné.
b) Najpočetnejšie párenie, ktoré nie je úplné.
c) Úplné párenie v K_6.

Úloha čínskeho poštára

Definícia

Nech $G = (V, H, c)$ je hranovo ohodnotený graf.
Párenie *v grafe G je taký jeho podgraf P, v ktorom má každý vrchol stupeň 1.*
Cena párenia *P je súčet ohodnotení jeho hrán.*
Hovoríme, že párenie P je **maximálne párenie** *v grafe G, ak P nie je podgrafom žiadneho iného párenia v G.*
Párenie P je **najpočetnejšie párenie** *v grafe G ak P má zo všetkých párení najväčší počet hrán.*
Párenie P je **úplné párenie** *v G, ak P je faktorovým podgrafom grafu G (P obsahuje všetky vrcholy grafu G).*



a) Maximálne párenie, ktoré nie je ani najpočetnejšie, ani úplné.
b) Najpočetnejšie párenie, ktoré nie je úplné.
c) Úplné párenie v K_6.

Edmondsov algoritmus

Algoritmus

Edmondsov algoritmus na hľadanie najkratšieho uzavretého eulerovského sledu v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$.

- **Krok 1.** *V grafe G nájdi všetky vrcholy nepárneho stupňa. Tých je párny počet $2t$.*
  Z vrcholov nepárneho stupňa zostroj úplný graf K_{2t}. Jeho hrany ohodnoť **vzdialenosťami koncových vrcholov hrany** *v pôvodnom grafe G.*
- **Krok 2.** *V grafe K_{2t} nájdi úplné párenie s minimálnou cenou.*
- **Krok 3.** *Hrany párenia pridaj k hranovej množine pôvodného grafu G. Dostaneš tak multigraf $\overline{G}$, v ktorom majú všetky vrcholy párny stupeň. V multigrafe $\overline{G}$ zostroj uzavretý eulerovský ťah $\mathcal{T}$.*
- **Krok 4.** *Hrany párenia v ťahu $\mathcal{T}$ nahraď príslušnými najkratšími cestami v grafe G a označ ich ako prejdené naprázdno. Dostaneš tak najkratší eulerovský uzavretý sled v grafe G.*

♣

Edmondsov algoritmus

Algoritmus

Edmondsov algoritmus na hľadanie najkratšieho uzavretého eulerovského sledu v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$.

- **Krok 1.** *V grafe G nájdi všetky vrcholy nepárneho stupňa. Tých je párny počet $2t$.*
  Z vrcholov nepárneho stupňa zostroj úplný graf K_{2t}. Jeho hrany ohodnoť **vzdialenosťami koncových vrcholov hrany** *v pôvodnom grafe G.*
- **Krok 2.** *V grafe K_{2t} nájdi úplné párenie s minimálnou cenou.*
- **Krok 3.** *Hrany párenia pridaj k hranovej množine pôvodného grafu G. Dostaneš tak multigraf $\overline{G}$, v ktorom majú všetky vrcholy párny stupeň. V multigrafe $\overline{G}$ zostroj uzavretý eulerovský ťah $\mathcal{T}$.*
- **Krok 4.** *Hrany párenia v ťahu $\mathcal{T}$ nahraď príslušnými najkratšími cestami v grafe G a označ ich ako prejdené naprázdno. Dostaneš tak najkratší eulerovský uzavretý sled v grafe G.*

♣

Edmondsov algoritmus

Algoritmus

Edmondsov algoritmus na hľadanie najkratšieho uzavretého eulerovského sledu v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$.

- **Krok 1.** *V grafe G nájdi všetky vrcholy nepárneho stupňa. Tých je párny počet $2t$.*
  Z vrcholov nepárneho stupňa zostroj úplný graf K_{2t}. Jeho hrany ohodnoť **vzdialenosťami koncových vrcholov hrany** *v pôvodnom grafe G.*
- **Krok 2.** *V grafe K_{2t} nájdi úplné párenie s minimálnou cenou.*
- **Krok 3.** *Hrany párenia pridaj k hranovej množine pôvodného grafu G. Dostaneš tak multigraf $\overline{G}$, v ktorom majú všetky vrcholy párny stupeň. V multigrafe $\overline{G}$ zostroj uzavretý eulerovský ťah $\mathcal{T}$.*
- **Krok 4.** *Hrany párenia v ťahu $\mathcal{T}$ nahraď príslušnými najkratšími cestami v grafe G a označ ich ako prejdené naprázdno. Dostaneš tak najkratší eulerovský uzavretý sled v grafe G.*

♣

Edmondsov algoritmus

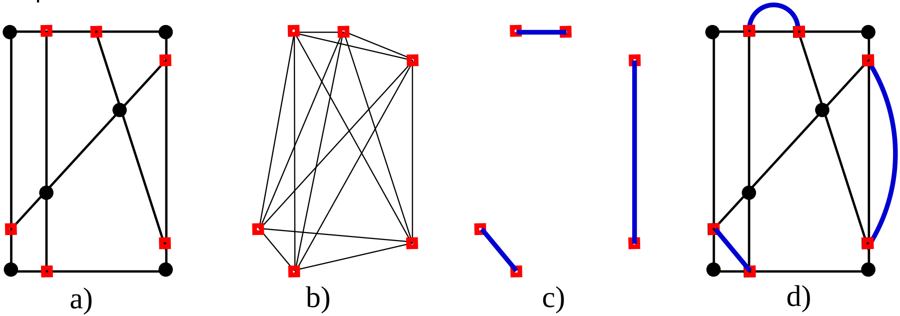
Algoritmus

Edmondsov algoritmus na hľadanie najkratšieho uzavretého eulerovského sledu v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$.

- **Krok 1.** *V grafe* G *nájdi všetky vrcholy nepárneho stupňa. Tých je párny počet* $2t$*.*
  Z vrcholov nepárneho stupňa zostroj úplný graf K_{2t}*. Jeho hrany ohodnoť* **vzdialenosťami koncových vrcholov hrany** *v pôvodnom grafe* G*.*
- **Krok 2.** *V grafe* K_{2t} *nájdi úplné párenie s minimálnou cenou.*
- **Krok 3.** *Hrany párenia pridaj k hranovej množine pôvodného grafu* G*. Dostaneš tak multigraf* $\overline{G}$*, v ktorom majú všetky vrcholy párny stupeň. V multigrafe* $\overline{G}$ *zostroj uzavretý eulerovský ťah* $\mathcal{T}$*.*
- **Krok 4.** *Hrany párenia v ťahu* $\mathcal{T}$ *nahraď príslušnými najkratšími cestami v grafe* G *a označ ich ako prejdené naprázdno. Dostaneš tak najkratší eulerovský uzavretý sled v grafe* G*.*

♣

Edmondsov algoritmus



Postup pri Edmondsovom algoritme.
a) Pôvodný graf, vrcholy nepárneho stupňa sú vyznačené štvorčekmi.
b) Pomocný úplný graf K_{2t} zostrojený podľa kroku 1. algoritmu 5.
c) Úplné párenie s minimálnou cenou v K_{2t}.
d) Multigraf $\overline{G}$ zostrojený podľa kroku 3. algoritmu 5,
kde už existuje eulerovský uzavretý ťah.

Hamiltonovský sled, hamiltonovský cyklus

Definícia

Sled v grafe G sa nazýva **hamiltonovský sled** *v grafe G, ak obsahuje všetky vrcholy grafu G.*

Poznámka

Predchádzajúca definícia definuje i hamiltonovskú cestu i hamiltonovský cyklus, pretože obe sú špeciálnym prípadom hamiltonovského sledu.

Definícia

Hovoríme, že graf G je **hamiltonovský**, *ak v ňom existuje hamiltonovský cyklus.*

Hamiltonovský sled, hamiltonovský cyklus

Neexistuje jednoduché kritérium na zixstenie toho, či je daný graf hamiltonovský.
Máme niekoľko hrubých postačujúcich podmienok:

Veta

Nech v grafe $G = (V, H)$ s aspoň troma vrcholmi pre každé dva také vrcholy vrcholy u, v, ktoré nie sú susedné, platí

$$\deg(u) + \deg(v) \geq |V|.$$

Potom je G hamiltonovský graf.

Veta

Nech v grafe $G = (V, H)$ s aspoň troma vrcholmi platí pre každý vrchol $v \in V$

$$\deg(v) \geq \frac{1}{2}.|V|.$$

Potom je G hamiltonovský graf.

Hamiltonovský sled, hamiltonovský cyklus

Neexistuje jednoduché kritérium na zixstenie toho, či je daný graf hamiltonovský.
Máme niekoľko hrubých postačujúcich podmienok:

Veta

Nech v grafe $G = (V, H)$ s aspoň troma vrcholmi pre každé dva také vrcholy vrcholy u, v, ktoré nie sú susedné, platí

$$\deg(u) + \deg(v) \geq |V|.$$

Potom je G hamiltonovský graf.

Veta

Nech v grafe $G = (V, H)$ s aspoň troma vrcholmi platí pre každý vrchol $v \in V$

$$\deg(v) \geq \frac{1}{2}.|V|.$$

Potom je G hamiltonovský graf.

Úloha obchodného cestujúceho – TSP

Úloha obchodného cestujúceho

Travelling Salesman Problem - TSP

Úloha obchodného cestujúceho – TSP

Slovná formulácia úlohy obchodného cestujúceho:
Obchodný cestujúci má navštíviť všetkých svojich zákazníkov a vrátiť sa domov tak, aby sa čo najmenej nachodil.

Matematická formulácia úlohy obchodného cestujúceho:
Ak dovoľujeme navštíviť to isté miesto viackrát, úlohu obchodného cestujúceho môžeme formulovať nasledovne:

V súvislom hranovo ohodnotenom grafe nájsť najkratší uzavretý hamiltonovský sled.

Ak zakazujeme navštíviť to isté miesto viackrát, úlohu obchodného cestujúceho formulujeme takto:

V súvislom hranovo ohodnotenom grafe nájsť najkratší hamiltonovský cyklus.

Úloha obchodného cestujúceho – TSP

Slovná formulácia úlohy obchodného cestujúceho:
Obchodný cestujúci má navštíviť všetkých svojich zákazníkov a vrátiť sa domov tak, aby sa čo najmenej nachodil.

Matematická formulácia úlohy obchodného cestujúceho:
Ak dovoľujeme navštíviť to isté miesto viackrát, úlohu obchodného cestujúceho môžeme formulovať nasledovne:

V súvislom hranovo ohodnotenom grafe nájsť najkratší uzavretý hamiltonovský sled.

Ak zakazujeme navštíviť to isté miesto viackrát, úlohu obchodného cestujúceho formulujeme takto:

V súvislom hranovo ohodnotenom grafe nájsť najkratší hamiltonovský cyklus.

Úloha obchodného cestujúceho – TSP

Slovná formulácia úlohy obchodného cestujúceho:
Obchodný cestujúci má navštíviť všetkých svojich zákazníkov a vrátiť sa domov tak, aby sa čo najmenej nachodil.

Matematická formulácia úlohy obchodného cestujúceho:
Ak dovoľujeme navštíviť to isté miesto viackrát, úlohu obchodného cestujúceho môžeme formulovať nasledovne:

V súvislom hranovo ohodnotenom grafe nájsť najkratší uzavretý hamiltonovský sled.

Ak zakazujeme navštíviť to isté miesto viackrát, úlohu obchodného cestujúceho formulujeme takto:

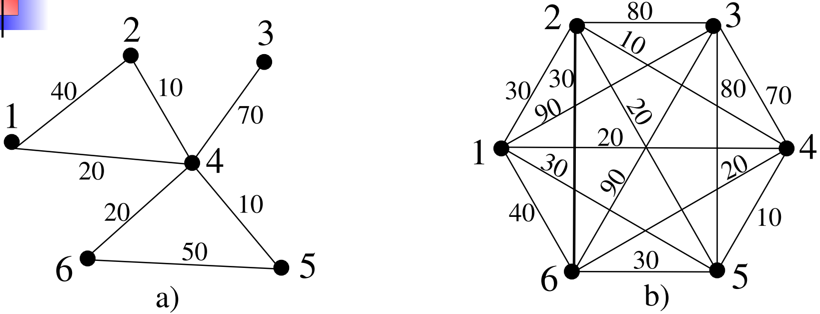
V súvislom hranovo ohodnotenom grafe nájsť najkratší hamiltonovský cyklus.

Úloha obchodného cestujúceho – TSP

Poznámka

V praktických prípadoch niet dôvodu zakazovať prejsť cez jedno miesto naviac. Navyše v modeloch mnohých skutočných sitácií hamiltonovský cyklus vôbec neexistuje. Preto sa sústredíme na hľadanie najkratšieho uzavretého hamiltonovského sledu.

Najkratší uzavretý hamiltonovský sled



a) b)

V grafe $G = (V, H, c)$ a) hamiltonovský cyklus neexistuje. Keďže nám stačí nájsť hamiltonovský sled, hľadáme ho pomocou hamiltonovského cyklu v úplnom grafe $\overline{G} = (G, E, d)$) (obr. b), ktorý má hrany ohodnotené vzdialenosťami v pôvodnom grafe G.

V grafe $\overline{G}$ platí trojuholníková nerovnosť t. j.:
$\forall u, v, w \in V \quad u, v, w$ po dvoch rôzne, platí:

$$d(u,v) \leq d(u,w) + d(w,v).$$

Najkratší hamilt. cyklus v úplnom grafe s $\triangle$ nerovnosťou

V úplnom grafe $\overline{G}$ už každá permutácia vrcholov definuje hamiltonovský cyklus.

Ak fixujeme prvý vrchol, potom máme $(n-1)!$ rôznych hamiltonovských cyklov.

Pre exaktné hľadanie najkratšieho hamiltonovského cyklu niet podstatne lepšieho algoritmu, ako systematické prehľadanie všetkých $(n-1)!$ permutácií.

Doba výpočtu pri prekontrolovaní 10^9 permutácií/sec.

n	$(n-1)!$	sekundy	minúty	dni	roky
10	3,6E+05	0,36 ms	-	-	-
15	8,7E+10	87,17	1,45	-	-
20	1,2E+17	1,2E+08	2000000	1400	3,9
25	6,2E+23	6,2E+14	1,0E+13	7,2E+09	2,0E+07
30	8,8E+30	8,8E+21	1,5E+20	1,0E+17	2,8E+14
35	3,0E+38	3,0E+29	4,9E+27	3,4E+24	9,4E+21
40	2,0E+46	2,0E+37	3,4E+35	2,4E+32	6,5E+29
45	2,7E+54	2,7E+45	4,4E+43	3,1E+40	8,4E+37
50	6,1E+62	6,1E+53	1,0E+52	7,0E+48	1,9E+46

Doba od Veľkého Tresku $1,4 * 10^{10}$ rokov.

Najkratší hamilt. cyklus v úplnom grafe s △ nerovnosťou

Dôsledok: Nutnosť používať algoritmy, ktoré dávajú dostatočne dobré, ale nie zaručene optimálne riešenie – suboptimálne algoritmy, heuristiky.

Algoritmus

Pažravá metóda – Greedy Algorithm. Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe $G = (V, H, c)$ s aspoň tromi vrcholmi a s trojuholníkovou nerovnosťou.

- **Krok 1.** *Začni v ľubovoľnom vrchole a do (budúceho) hamiltonovského cyklu vlož najlacnejšiu hranu incidentnú s týmto vrcholom.*
- **Krok 2.** *Ak je vybratých $n - 1$ hrán, uzavri cyklus. STOP*
- **Krok 3.** *Inak vyber takú najlacnejšiu nevybranú hranu incidentnú s posledným vrcholom doteraz vybranej postupnosti, ktorá nie je incidentná so žiadnym iným vrcholom vybranej postupnosti. GOTO Krok 2.*

♣

Najkratší hamilt. cyklus v úplnom grafe s △ nerovnosťou

Dôsledok: Nutnosť používať algoritmy, ktoré dávajú dostatočne dobré, ale nie zaručene optimálne riešenie – suboptimálne algoritmy, heuristiky.

Algoritmus

Pažravá metóda – Greedy Algorithm. Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe $G = (V, H, c)$ s aspoň tromi vrcholmi a s trojuholníkovou nerovnosťou.

- **Krok 1.** *Začni v ľubovoľnom vrchole a do (budúceho) hamiltonovského cyklu vlož najlacnejšiu hranu incidentnú s týmto vrcholom.*
- **Krok 2.** *Ak je vybratých* $n-1$ *hrán, uzavri cyklus. STOP*
- **Krok 3.** *Inak vyber takú najlacnejšiu nevybranú hranu incidentnú s posledným vrcholom doteraz vybranej postupnosti, ktorá nie je incidentná so žiadnym iným vrcholom vybranej postupnosti.*

  GOTO Krok 2.

♣

Najkratší hamilt. cyklus v úplnom grafe s △ nerovnosťou

Dôsledok: Nutnosť používať algoritmy, ktoré dávajú dostatočne dobré, ale nie zaručene optimálne riešenie – suboptimálne algoritmy, heuristiky.

Algoritmus

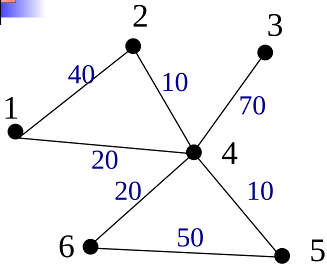
Pažravá metóda – Greedy Algorithm. Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe $G = (V, H, c)$ s aspoň tromi vrcholmi a s trojuholníkovou nerovnosťou.

- **Krok 1.** *Začni v ľubovoľnom vrchole a do (budúceho) hamiltonovského cyklu vlož najlacnejšiu hranu incidentnú s týmto vrcholom.*
- **Krok 2.** *Ak je vybratých $n-1$ hrán, uzavri cyklus. STOP*
- **Krok 3.** *Inak vyber takú najlacnejšiu nevybranú hranu incidentnú s posledným vrcholom doteraz vybranej postupnosti, ktorá nie je incidentná so žiadnym iným vrcholom vybranej postupnosti.*

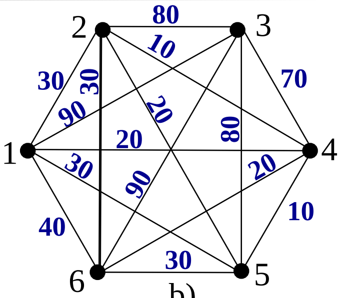
  GOTO Krok 2.

♣

Greedy algoritmus pre TSP



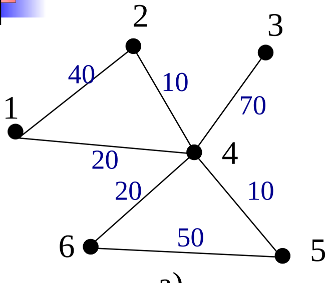
a)



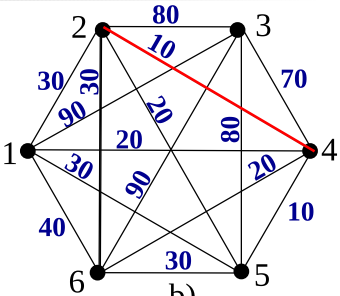
b)

$\mathcal{C} = (2)$

Greedy algoritmus pre TSP



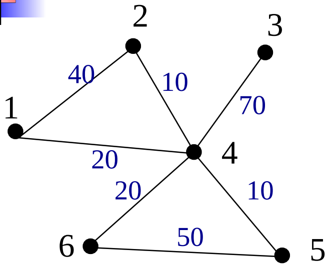
a)



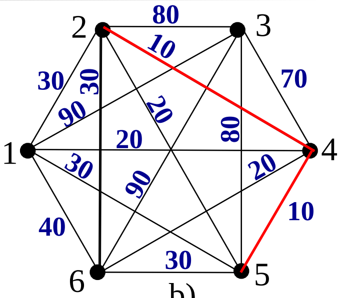
b)

$\mathcal{C} = (2, \{2, 4\}, 4)$

Greedy algoritmus pre TSP



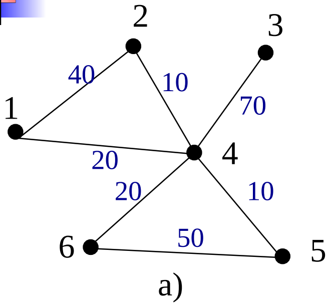
a)



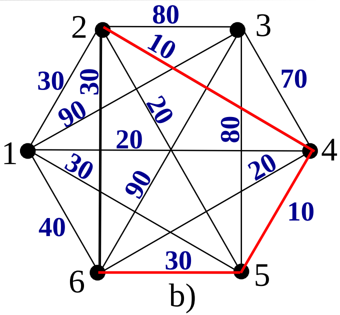
b)

$\mathcal{C} = (2, \{2, 4\}, 4, \{4, 5\}, 5)$

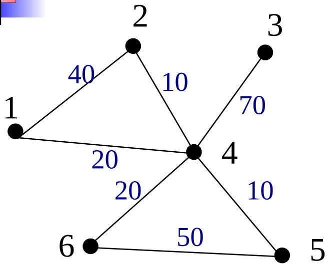
Greedy algoritmus pre TSP



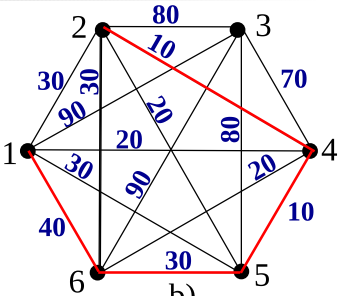
$\mathcal{C} = (2, \{2,4\}, 4, \{4,5\}, 5, \{5,6\}, 6)$



Greedy algoritmus pre TSP



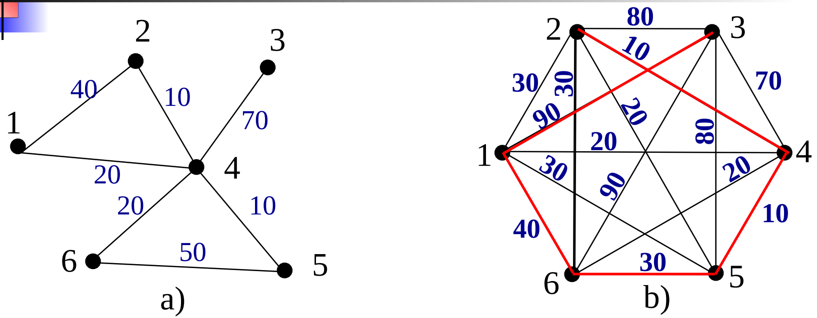
a)



b)

$$\mathcal{C} = (2, \{2,4\}, 4, \{4,5\}, 5, \{5,6\}, 6, \{6,1\}, 1)$$

Greedy algoritmus pre TSP

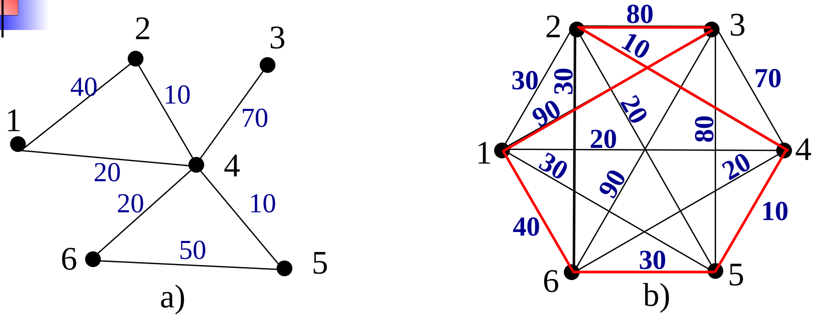


a)

b)

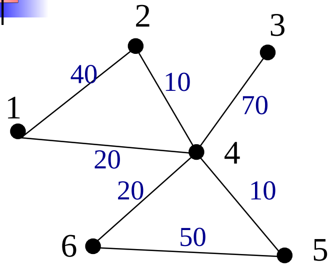
$$\mathcal{C} = (2, \{2, 4\}, 4, \{4, 5\}, 5, \{5, 6\}, 6, \{6, 1\}, 1, \{1, 3\}, 3)$$

Greedy algoritmus pre TSP

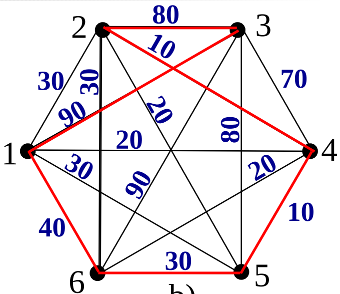


$\mathcal{C} = (2, \{2, 4\}, 4, \{4, 5\}, 5, \{5, 6\}, 6, \{6, 1\}, 1, \{1, 3\}, 3, \{3, 2\}, 2)$

Greedy algoritmus pre TSP



a)



b)

$\mathcal{C} = (2, \{2, 4\}, 4, \{4, 5\}, 5, \{5, 6\}, 6, \{6, 1\}, 1, \{1, 3\}, 3, \{3, 2\}, 2)$

Každú hranu cyklu $\mathcal{C}$ nahradíme príslušnou najkratšou cestou v pôvodnom grafe G

$(2, \{2, 4\}, 4) \to (2, \{2, 4\}, 4)$
$(4, \{4, 5\}, 5) \to (4, \{4, 5\}, 5)$
$(5, \{5, 6\}, 6) \to (5, \{5, 4\}, 4, \{4, 6\}, 6)$
$(6, \{6, 1\}, 1) \to (6, \{6, 4\}, 4, \{4, 1\}, 1)$
$(1, \{1, 3\}, 3) \to (1, \{1, 4\}, 4, \{4, 3\}, 3)$
$(3, \{3, 2\}, 2) \to (3, \{3, 4\}, 4, \{4, 2\}, 2)$

Metóda zdvojenia kostry.

Algoritmus

Metóda zdvojenia kostry. (Kim – 1975). Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe $G = (V, H, c)$ s trojuholníkovou nerovnosťou.

- **Krok 1.** *V grafe G zostroj najlacnejšiu kostru K.*
- **Krok 2.** *V kostre K zostroj uzavretý sled $\mathcal{S}$, ktorý obsahuje každú hranu práve dvakrát. (Použi napr. Tarryho algoritmus).*
- **Krok 3.** *Z uzavretého sledu $\mathcal{S}$ vytvor hamiltonovský cyklus takto: Postupne prechádzaj sledom $\mathcal{S}$ a keď narazíš na taký vrchol, alebo úsek niekoľkých vrcholov za sebou, ktoré sa už v slede vyskytujú, premosti takýto vrchol alebo úsek priamou hranou.*

♣

Metóda zdvojenia kostry.

Algoritmus

Metóda zdvojenia kostry. (Kim – 1975). Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe $G = (V, H, c)$ s trojuholníkovou nerovnosťou.

- **Krok 1.** *V grafe G zostroj najlacnejšiu kostru K.*
- **Krok 2.** *V kostre K zostroj uzavretý sled $\mathcal{S}$, ktorý obsahuje každú hranu práve dvakrát. (Použi napr. Tarryho algoritmus).*
- **Krok 3.** *Z uzavretého sledu $\mathcal{S}$ vytvor hamiltonovský cyklus takto: Postupne prechádzaj sledom $\mathcal{S}$ a keď narazíš na taký vrchol, alebo úsek niekoľkých vrcholov za sebou, ktoré sa už v slede vyskytujú, premosti takýto vrchol alebo úsek priamou hranou.*

♣

Metóda zdvojenia kostry.

Algoritmus

Metóda zdvojenia kostry. (Kim – 1975). Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe $G = (V, H, c)$ s trojuholníkovou nerovnosťou.

- **Krok 1.** *V grafe* G *zostroj najlacnejšiu kostru* K.
- **Krok 2.** *V kostre* K *zostroj uzavretý sled* $\mathcal{S}$, *ktorý obsahuje každú hranu práve dvakrát. (Použi napr. Tarryho algoritmus).*
- **Krok 3.** *Z uzavretého sledu* $\mathcal{S}$ *vytvor hamiltonovský cyklus takto: Postupne prechádzaj sledom* $\mathcal{S}$ *a keď narazíš na taký vrchol, alebo úsek niekoľkých vrcholov za sebou, ktoré sa už v slede vyskytujú, premosti takýto vrchol alebo úsek priamou hranou.*

♣

Metóda zdvojenia kostry

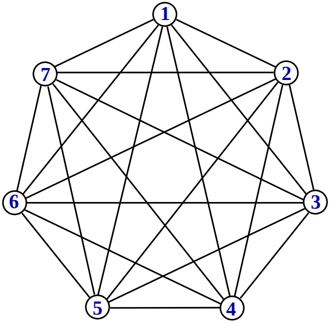
Veta

Nech $G = (V, H, c)$ je úplný graf, v ktorom platí trojuholníková nerovnosť. Nech $c(MZK)$ je dĺžka hamiltonovského cyklu získaného metódou zdvojenia kostry, nech $c(OPT)$ je dĺžka najkratšieho hamiltonovského cyklu v grafe G. Potom

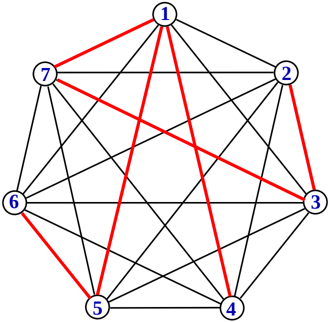
$$\frac{c(MZK)}{c(OPT)} < 2.$$

Naviac posledný odhad už nemožno zlepšiť – pre každé $\varepsilon > 0$ existuje taký graf G_ε, že preň je $c(MZK)/c(OPT) > 2 - \varepsilon$.

Metóda zdvojenia kostry – príklad

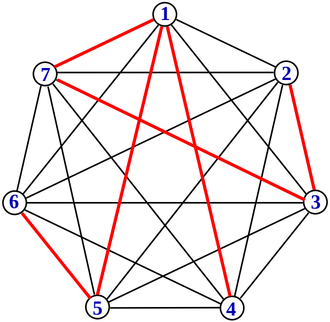


Metóda zdvojenia kostry – príklad



$$\mathcal{T} = (1, \{1,7\}, 7, \{7,3\}, 3, \{3,2\}, 2, \{2,3\}, 3, \{3,7\}, 7, \{7,1\}, 1,$$
$$\{4,1\}, 4, \{4,1,\}, 1, \{1,5\}, 5, \{5,6\}, 6, \{6,5\}, 5, \{5,1\}, 1)$$

Metóda zdvojenia kostry – príklad

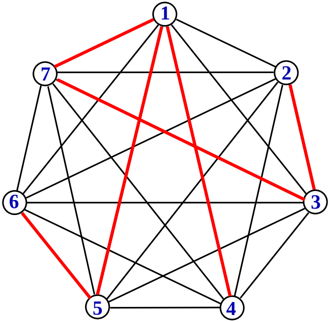


$$\mathcal{T} = (1, \{1,7\}, 7, \{7,3\}, 3, \{3,2\}, 2, \{2,3\}, 3, \{3,7\}, 7, \{7,1\}, 1,$$
$$\{4,1\}, 4, \{4,1,\}, 1, \{1,5\}, 5, \{5,6\}, 6, \{6,5\}, 5, \{5,1\}, 1)$$

Skrátene

$$\mathcal{T} = (1, 7, 3, 2, \underbrace{\mathbf{3, 7, 1}}_{\text{premostiť hranou } \{2,4\}}, 4, 1, 5, 6, 5, 1)$$

Metóda zdvojenia kostry – príklad



$$\mathcal{T} = (1, \{1,7\}, 7, \{7,3\}, 3, \{3,2\}, 2, \{2,3\}, 3, \{3,7\}, 7, \{7,1\}, 1,$$

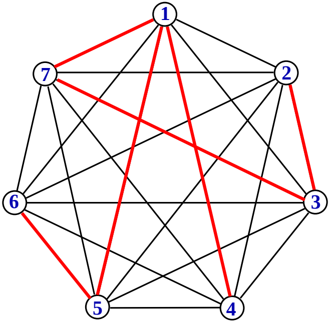
$$\{4,1\}, 4, \{4,1,\}, 1, \{1,5\}, 5, \{5,6\}, 6, \{6,5\}, 5, \{5,1\}, 1)$$

Skrátene

$$\mathcal{T} = (1, 7, 3, 2, \underbrace{\mathbf{3}, \mathbf{7}, \mathbf{1}}_{\text{premostiť hranou } \{2,4\}}, 4, 1, 5, 6, 5, 1)$$

$$\mathcal{T} = (1, 7, 3, 2, 4, \mathbf{1}, 5, 6, 5, 1)$$

Metóda zdvojenia kostry – príklad



$$\mathcal{T} = (1, \{1,7\}, 7, \{7,3\}, 3, \{3,2\}, 2, \{2,3\}, 3, \{3,7\}, 7, \{7,1\}, 1,$$

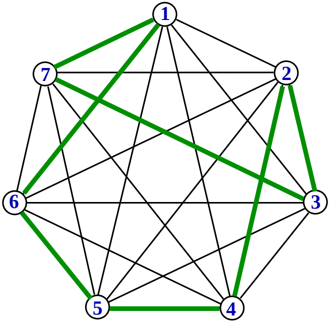
Skrátene $\{4,1\}, 4, \{4,1,\}, 1, \{1,5\}, 5, \{5,6\}, 6, \{6,5\}, 5, \{5,1\}, 1)$

$$\mathcal{T} = (1, 7, 3, 2, \underbrace{\mathbf{3}, \mathbf{7}, \mathbf{1}}_{\text{premostiť hranou } \{2,4\}}, 4, 1, 5, 6, 5, 1)$$

$$\mathcal{T} = (1, 7, 3, 2, 4, \mathbf{1}, 5, 6, 5, 1)$$

$$\mathcal{T} = (1, 7, 3, 2, 4, 5, 6, \mathbf{5}, 1)$$

Metóda zdvojenia kostry – príklad



$$\mathcal{T} = (1, \{1,7\}, 7, \{7,3\}, 3, \{3,2\}, 2, \{2,3\}, 3, \{3,7\}, 7, \{7,1\}, 1,$$

Skrátene $\{4,1\}, 4, \{4,1,\}, 1, \{1,5\}, 5, \{5,6\}, 6, \{6,5\}, 5, \{5,1\}, 1)$

$$\mathcal{T} = (1, 7, 3, 2, \underbrace{\mathbf{3}, \mathbf{7}, \mathbf{1}}_{\text{premostiť hranou } \{2,4\}}, 4, 1, 5, 6, 5, 1)$$

$$\mathcal{T} = (1, 7, 3, 2, 4, \mathbf{1}, 5, 6, 5, 1)$$

$$\mathcal{T} = (1, 7, 3, 2, 4, 5, 6, \mathbf{5}, 1)$$

$$\mathcal{T} = (1, 7, 3, 2, 4, 5, 6, 1)$$

Algoritmus kostry a párenia.

Algoritmus

Algoritmus kostry a párenia. (Christofides – 1976.) *Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe* $G = (V, H, c)$ *s trojuholníkovou nerovnosťou.*

- **Krok 1.** *V grafe* G *zostroj najlacnejšiu kostru* K.
- **Krok 2.** *V kostre* K *nájdi všetky vrcholy nepárneho stupňa. Tých je* $2t$.
- **Krok 3.** *Z vrcholov nepárneho stupňa zostroj úplný graf* K_{2t}, *jeho hrany ohodnoť ohodnoteniami príslušných hrán v pôvodnom grafe* G.
- **Krok 4.** *V grafe* K_{2t} *nájdi úplné párenie s minimálnou cenou.*
- **Krok 5.** *Hrany párenia dodaj k hranovej množine najlacnejšej kostry* K. *Dostaneš tak graf (multigraf)* $\overline{G}$, *ktorý má všetky vrcholy párneho stupňa.*
- **Krok 6.** *V grafe (resp. multigrafe)* $\overline{G}$ *zostroj uzavretý eulerovský ťah* $\mathcal{T}$.
- **Krok 7.** *Z uzavretého ťahu* $\mathcal{T}$ *vytvor hamiltonovský cyklus takto: Postupne prechádzaj ťahom* $\mathcal{T}$ *a keď narazíš na taký vrchol, alebo úsek niekoľkých vrcholov za sebou, ktoré sa už v slede vyskytujú, premosti takýto vrchol alebo úsek priamou hranou.*

♣

Algoritmus kostry a párenia.

Algoritmus

Algoritmus kostry a párenia. (Christofides – 1976.) *Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe* $G = (V, H, c)$ *s trojuholníkovou nerovnosťou.*

- **Krok 1.** *V grafe* G *zostroj najlacnejšiu kostru* K.
- **Krok 2.** *V kostre* K *nájdi všetky vrcholy nepárneho stupňa. Tých je* $2t$.
- **Krok 3.** *Z vrcholov nepárneho stupňa zostroj úplný graf* K_{2t}, *jeho hrany ohodnoť ohodnoteniami príslušných hrán v pôvodnom grafe* G.
- **Krok 4.** *V grafe* K_{2t} *nájdi úplné párenie s minimálnou cenou.*
- **Krok 5.** *Hrany párenia dodaj k hranovej množine najlacnejšej kostry* K. *Dostaneš tak graf (multigraf)* $\overline{G}$, *ktorý má všetky vrcholy párneho stupňa.*
- **Krok 6.** *V grafe (resp. multigrafe)* $\overline{G}$ *zostroj uzavretý eulerovský ťah* $\mathcal{T}$.
- **Krok 7.** *Z uzavretého ťahu* $\mathcal{T}$ *vytvor hamiltonovský cyklus takto: Postupne prechádzaj ťahom* $\mathcal{T}$ *a keď narazíš na taký vrchol, alebo úsek niekoľkých vrcholov za sebou, ktoré sa už v slede vyskytujú, premosti takýto vrchol alebo úsek priamou hranou.*

♣

Algoritmus kostry a párenia.

Algoritmus

Algoritmus kostry a párenia. (Christofides – 1976.) *Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe* $G = (V, H, c)$ *s trojuholníkovou nerovnosťou.*

- **Krok 1.** *V grafe* G *zostroj najlacnejšiu kostru* K.
- **Krok 2.** *V kostre* K *nájdi všetky vrcholy nepárneho stupňa. Tých je* $2t$.
- **Krok 3.** *Z vrcholov nepárneho stupňa zostroj úplný graf* K_{2t}, *jeho hrany ohodnoť ohodnoteniami príslušných hrán v pôvodnom grafe* G.
- **Krok 4.** *V grafe* K_{2t} *nájdi úplné párenie s minimálnou cenou.*
- **Krok 5.** *Hrany párenia dodaj k hranovej množine najlacnejšej kostry* K. *Dostaneš tak graf (multigraf)* $\overline{G}$, *ktorý má všetky vrcholy párneho stupňa.*
- **Krok 6.** *V grafe (resp. multigrafe)* $\overline{G}$ *zostroj uzavretý eulerovský ťah* $\mathcal{T}$.
- **Krok 7.** *Z uzavretého ťahu* $\mathcal{T}$ *vytvor hamiltonovský cyklus takto: Postupne prechádzaj ťahom* $\mathcal{T}$ *a keď narazíš na taký vrchol, alebo úsek niekoľkých vrcholov za sebou, ktoré sa už v slede vyskytujú, premosti takýto vrchol alebo úsek priamou hranou.*

♣

Algoritmus kostry a párenia.

Algoritmus

Algoritmus kostry a párenia. (Christofides – 1976.) *Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe* $G = (V, H, c)$ *s trojuholníkovou nerovnosťou.*

- **Krok 1.** *V grafe* G *zostroj najlacnejšiu kostru* K.
- **Krok 2.** *V kostre* K *nájdi všetky vrcholy nepárneho stupňa. Tých je* $2t$.
- **Krok 3.** *Z vrcholov nepárneho stupňa zostroj úplný graf* K_{2t}, *jeho hrany ohodnoť ohodnoteniami príslušných hrán v pôvodnom grafe* G.
- **Krok 4.** *V grafe* K_{2t} *nájdi úplné párenie s minimálnou cenou.*
- **Krok 5.** *Hrany párenia dodaj k hranovej množine najlacnejšej kostry* K. *Dostaneš tak graf (multigraf)* $\overline{G}$, *ktorý má všetky vrcholy párneho stupňa.*
- **Krok 6.** *V grafe (resp. multigrafe)* $\overline{G}$ *zostroj uzavretý eulerovský ťah* $\mathcal{T}$.
- **Krok 7.** *Z uzavretého ťahu* $\mathcal{T}$ *vytvor hamiltonovský cyklus takto: Postupne prechádzaj ťahom* $\mathcal{T}$ *a keď narazíš na taký vrchol, alebo úsek niekoľkých vrcholov za sebou, ktoré sa už v slede vyskytujú, premosti takýto vrchol alebo úsek priamou hranou.*

♣

Algoritmus kostry a párenia.

Algoritmus

Algoritmus kostry a párenia. (Christofides – 1976.) *Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe* $G = (V, H, c)$ *s trojuholníkovou nerovnosťou.*

- **Krok 1.** *V grafe* G *zostroj najlacnejšiu kostru* K.
- **Krok 2.** *V kostre* K *nájdi všetky vrcholy nepárneho stupňa. Tých je* $2t$.
- **Krok 3.** *Z vrcholov nepárneho stupňa zostroj úplný graf* K_{2t}, *jeho hrany ohodnoť ohodnoteniami príslušných hrán v pôvodnom grafe* G.
- **Krok 4.** *V grafe* K_{2t} *nájdi úplné párenie s minimálnou cenou.*
- **Krok 5.** *Hrany párenia dodaj k hranovej množine najlacnejšej kostry* K. *Dostaneš tak graf (multigraf)* $\overline{G}$, *ktorý má všetky vrcholy párneho stupňa.*
- **Krok 6.** *V grafe (resp. multigrafe)* $\overline{G}$ *zostroj uzavretý eulerovský ťah* $\mathcal{T}$.
- **Krok 7.** *Z uzavretého ťahu* $\mathcal{T}$ *vytvor hamiltonovský cyklus takto: Postupne prechádzaj ťahom* $\mathcal{T}$ *a keď narazíš na taký vrchol, alebo úsek niekoľkých vrcholov za sebou, ktoré sa už v slede vyskytujú, premosti takýto vrchol alebo úsek priamou hranou.*

♣

Algoritmus kostry a párenia.

Algoritmus

Algoritmus kostry a párenia. (Christofides – 1976.) *Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe* $G = (V, H, c)$ *s trojuholníkovou nerovnosťou.*

- **Krok 1.** *V grafe* G *zostroj najlacnejšiu kostru* K.
- **Krok 2.** *V kostre* K *nájdi všetky vrcholy nepárneho stupňa. Tých je* $2t$.
- **Krok 3.** *Z vrcholov nepárneho stupňa zostroj úplný graf* K_{2t}, *jeho hrany ohodnoť ohodnoteniami príslušných hrán v pôvodnom grafe* G.
- **Krok 4.** *V grafe* K_{2t} *nájdi úplné párenie s minimálnou cenou.*
- **Krok 5.** *Hrany párenia dodaj k hranovej množine najlacnejšej kostry* K. *Dostaneš tak graf (multigraf)* $\overline{G}$, *ktorý má všetky vrcholy párneho stupňa.*
- **Krok 6.** *V grafe (resp. multigrafe)* $\overline{G}$ *zostroj uzavretý eulerovský ťah* $\mathcal{T}$.
- **Krok 7.** *Z uzavretého ťahu* $\mathcal{T}$ *vytvor hamiltonovský cyklus takto: Postupne prechádzaj ťahom* $\mathcal{T}$ *a keď narazíš na taký vrchol, alebo úsek niekoľkých vrcholov za sebou, ktoré sa už v slede vyskytujú, premosti takýto vrchol alebo úsek priamou hranou.*

♣

Algoritmus kostry a párenia.

Algoritmus

Algoritmus kostry a párenia. (Christofides – 1976.) *Heuristika na hľadanie suboptimálneho riešenia úlohy obchodného cestujúceho v úplnom grafe $G = (V, H, c)$ s trojuholníkovou nerovnosťou.*

- **Krok 1.** *V grafe G zostroj najlacnejšiu kostru K.*
- **Krok 2.** *V kostre K nájdi všetky vrcholy nepárneho stupňa. Tých je $2t$.*
- **Krok 3.** *Z vrcholov nepárneho stupňa zostroj úplný graf K_{2t}, jeho hrany ohodnoť ohodnoteniami príslušných hrán v pôvodnom grafe G.*
- **Krok 4.** *V grafe K_{2t} nájdi úplné párenie s minimálnou cenou.*
- **Krok 5.** *Hrany párenia dodaj k hranovej množine najlacnejšej kostry K. Dostaneš tak graf (multigraf) $\overline{G}$, ktorý má všetky vrcholy párneho stupňa.*
- **Krok 6.** *V grafe (resp. multigrafe) $\overline{G}$ zostroj uzavretý eulerovský ťah $\mathcal{T}$.*
- **Krok 7.** *Z uzavretého ťahu $\mathcal{T}$ vytvor hamiltonovský cyklus takto: Postupne prechádzaj ťahom $\mathcal{T}$ a keď narazíš na taký vrchol, alebo úsek niekoľkých vrcholov za sebou, ktoré sa už v slede vyskytujú, premosti takýto vrchol alebo úsek priamou hranou.*

♣

Algoritmus kostry a párenia.

Veta

Nech $G = (V, H, c)$ je úplný graf, v ktorom platí trojuholníková nerovnosť. Nech $c(MKP)$ je dĺžka hamiltonovského cyklu získaného metódou kostry a párenia, nech $c(OPT)$ je dĺžka najkratšieho hamiltonovského cyklu v grafe G. Potom

$$\frac{c(MKP)}{c(OPT)} < \frac{3}{2}.$$

Naviac posledný odhad už nemožno zlepšiť – pre každé $\varepsilon > 0$ existuje taký graf G_ε, že preň je $c(MKP)/c(OPT) > 3/2 - \varepsilon$.

Poznámka

Nie je známy polynomiálny algoritmus ALG, pre ktorý by bol zaručený lepší pomer $c(ALG)/c(OPT)$ než $3/2$.

Algoritmus kostry a párenia.

Veta

Nech $G = (V, H, c)$ je úplný graf, v ktorom platí trojuholníková nerovnosť. Nech $c(MKP)$ je dĺžka hamiltonovského cyklu získaného metódou kostry a párenia, nech $c(OPT)$ je dĺžka najkratšieho hamiltonovského cyklu v grafe G. Potom

$$\frac{c(MKP)}{c(OPT)} < \frac{3}{2}.$$

Naviac posledný odhad už nemožno zlepšiť – pre každé $\varepsilon > 0$ existuje taký graf G_ε, že preň je $c(MKP)/c(OPT) > 3/2 - \varepsilon$.

Poznámka

Nie je známy polynomiálny algoritmus ALG, pre ktorý by bol zaručený lepší pomer $c(ALG)/c(OPT)$ než $3/2$.

Vkladacia heuristika pre TSP

Algoritmus

Vkladacia heuristika na konštrukciu suboptimálneho hamiltonovského cyklu v úplnom grafe $G = (V, H, c)$ s trojuholníkovou nerovnosťou.

- **Krok 1.** *Do cyklu zaraď hranu* $h = \{u, v\}$ *s najmenšou cenou. Nájdi vrchol* $w \in V$, *pre ktorý je súčet* $c\{u, w\} + c\{w, v\}$ *najmenší. Vytvor cyklus* $C = (u, \{u, w\}, w, \{w, v\}, v, \{v, u\}, u)$.
- **Krok 2.** *Ak cyklus* C *obsahuje všetky vrcholy grafu* G, *STOP. Inak pokračuj krokom 3.*
- **Krok 3.** *Pre každú hranu* $h = \{u, v\}$ *cyklu* C *vypočítaj*

  $$z(h) = \min\{c\{u, w\} + c\{w, v\} - c\{u, v\} \mid w \in V - C\}.$$

  Vezmi hranu $h = \{u, v\}$ *s minimálnym* $z(h)$ *a* w *vrchol, pre ktorý nastalo minimum v (9). Vytvor cyklus* C' *tak, že nahradíš hranu* $\{u, v\}$ *dvojicou hrán* $\{u, w\}$, $\{w, v\}$. *Polož* $C := C'$.

  GOTO Krok 2.

♣

Vkladacia heuristika pre TSP

Algoritmus

Vkladacia heuristika na konštrukciu suboptimálneho hamiltonovského cyklu v úplnom grafe $G = (V, H, c)$ s trojuholníkovou nerovnosťou.

- **Krok 1.** *Do cyklu zaraď hranu* $h = \{u, v\}$ *s najmenšou cenou. Nájdi vrchol* $w \in V$, *pre ktorý je súčet* $c\{u, w\} + c\{w, v\}$ *najmenší. Vytvor cyklus* $C = (u, \{u, w\}, w, \{w, v\}, v, \{v, u\}, u)$.
- **Krok 2.** *Ak cyklus* C *obsahuje všetky vrcholy grafu* G, *STOP. Inak pokračuj krokom 3.*
- **Krok 3.** *Pre každú hranu* $h = \{u, v\}$ *cyklu* C *vypočítaj*

  $$z(h) = \min\{c\{u, w\} + c\{w, v\} - c\{u, v\} \mid w \in V - C\}.$$

  Vezmi hranu $h = \{u, v\}$ *s minimálnym* $z(h)$ *a* w *vrchol, pre ktorý nastalo minimum v (9). Vytvor cyklus* C' *tak, že nahradíš hranu* $\{u, v\}$ *dvojicou hrán* $\{u, w\}$, $\{w, v\}$. *Polož* $C := C'$.

  GOTO Krok 2.

♣

Vkladacia heuristika pre TSP

Algoritmus

Vkladacia heuristika na konštrukciu suboptimálneho hamiltonovského cyklu v úplnom grafe $G = (V, H, c)$ s trojuholníkovou nerovnosťou.

- **Krok 1.** *Do cyklu zaraď hranu* $h = \{u, v\}$ *s najmenšou cenou. Nájdi vrchol* $w \in V$, *pre ktorý je súčet* $c\{u, w\} + c\{w, v\}$ *najmenší. Vytvor cyklus* $C = (u, \{u, w\}, w, \{w, v\}, v, \{v, u\}, u)$.
- **Krok 2.** *Ak cyklus* C *obsahuje všetky vrcholy grafu* G, *STOP. Inak pokračuj krokom 3.*
- **Krok 3.** *Pre každú hranu* $h = \{u, v\}$ *cyklu* C *vypočítaj*

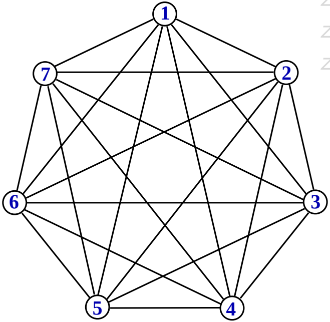
  $$z(h) = \min\{c\{u, w\} + c\{w, v\} - c\{u, v\} \mid w \in V - C\}.$$

  Vezmi hranu $h = \{u, v\}$ *s minimálnym* $z(h)$ *a* w *vrchol, pre ktorý nastalo minimum v* (9). *Vytvor cyklus* C' *tak, že nahradíš hranu* $\{u, v\}$ *dvojicou hrán* $\{u, w\}$, $\{w, v\}$. *Polož* $C := C'$.

  GOTO Krok 2.

♣

Vkladacia heuristika pre TSP



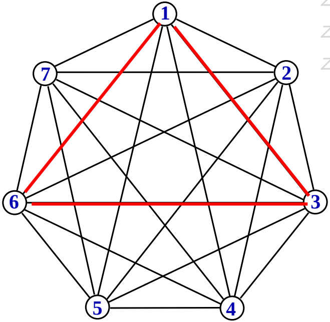
$$z(h) = c\{6,5\} + c\{5,3\} - c\{6,3\}$$
$$z(h) = \min\{z(h),\ c\{6,4\} + c\{4,3\} - c\{6,3\}\}$$
$$z(h) = \min\{z(h),\ c\{6,2\} + c\{2,3\} - c\{6,3\}\}$$
$$z(h) = \min\{z(h),\ c\{6,7\} + c\{7,3\} - c\{6,3\}\}$$

$$z(h) = \min\left\{\begin{array}{l} c\{6,5\} + c\{5,3\} - c\{6,3\}, \\ c\{6,4\} + c\{4,3\} - c\{6,3\}, \\ c\{6,2\} + c\{2,3\} - c\{6,3\}, \\ c\{6,7\} + c\{7,3\} - c\{6,3\} \end{array}\right\}$$

Poznámka

Algoritmus vkladacia heuristika vytvára postupne cykly tak, že do súčasného cyklu vkladná taký vrchol, ktorým ho najmenej predĺži. Je to typická vytvárajúca heuristika.

Vkladacia heuristika pre TSP



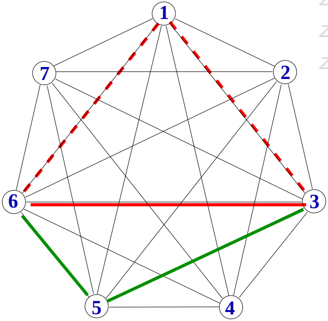
$$\begin{aligned}
z(h) &= \quad c\{6,5\} + c\{5,3\} - c\{6,3\}\\
z(h) &= \min\{z(h),\ c\{6,4\} + c\{4,3\} - c\{6,3\}\}\\
z(h) &= \min\{z(h),\ c\{6,2\} + c\{2,3\} - c\{6,3\}\}\\
z(h) &= \min\{z(h),\ c\{6,7\} + c\{7,3\} - c\{6,3\}\}
\end{aligned}$$

$$z(h) = \min \left\{ \begin{array}{l} c\{6,5\} + c\{5,3\} - c\{6,3\}, \\ c\{6,4\} + c\{4,3\} - c\{6,3\}, \\ c\{6,2\} + c\{2,3\} - c\{6,3\}, \\ c\{6,7\} + c\{7,3\} - c\{6,3\} \end{array} \right\}$$

Poznámka

Algoritmus vkladacia heuristika vytvára postupne cykly tak, že do súčasného cyklu vkladná taký vrchol, ktorým ho najmenej predĺži. Je to typická vytvárajúca heuristika.

Vkladacia heuristika pre TSP



$$z(h) = \qquad c\{6,5\} + c\{5,3\} - c\{6,3\}$$

$$z(h) = \min\{z(h),\ c\{6,4\} + c\{4,3\} - c\{6,3\}\}$$

$$z(h) = \min\{z(h),\ c\{6,2\} + c\{2,3\} - c\{6,3\}\}$$

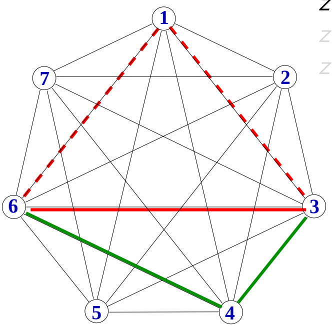
$$z(h) = \min\{z(h),\ c\{6,7\} + c\{7,3\} - c\{6,3\}\}$$

$$z(h) = \min\begin{Bmatrix} c\{6,5\} + c\{5,3\} - c\{6,3\}, \\ c\{6,4\} + c\{4,3\} - c\{6,3\}, \\ c\{6,2\} + c\{2,3\} - c\{6,3\}, \\ c\{6,7\} + c\{7,3\} - c\{6,3\} \end{Bmatrix}$$

Poznámka

Algoritmus vkladacia heuristika vytvára postupne cykly tak,
že do súčasného cyklu vkladná taký vrchol, ktorým ho najmenej predĺži.
Je to typická vytvárajúca heuristika.

Vkladacia heuristika pre TSP



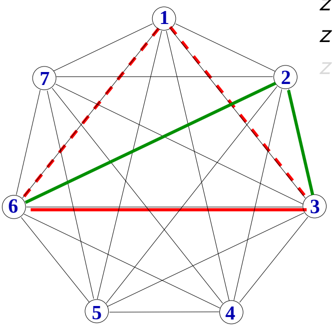
$$z(h) = c\{6,5\} + c\{5,3\} - c\{6,3\}$$
$$z(h) = \min\{z(h),\ c\{6,4\} + c\{4,3\} - c\{6,3\}\}$$
$$z(h) = \min\{z(h),\ c\{6,2\} + c\{2,3\} - c\{6,3\}\}$$
$$z(h) = \min\{z(h),\ c\{6,7\} + c\{7,3\} - c\{6,3\}\}$$

$$z(h) = \min\left\{\begin{array}{l} c\{6,5\} + c\{5,3\} - c\{6,3\}, \\ c\{6,4\} + c\{4,3\} - c\{6,3\}, \\ c\{6,2\} + c\{2,3\} - c\{6,3\}, \\ c\{6,7\} + c\{7,3\} - c\{6,3\} \end{array}\right\}$$

Poznámka

Algoritmus vkladacia heuristika vytvára postupne cykly tak,
že do súčasného cyklu vkladná taký vrchol, ktorým ho najmenej predĺži.
Je to typická vytvárajúca heuristika.

Vkladacia heuristika pre TSP



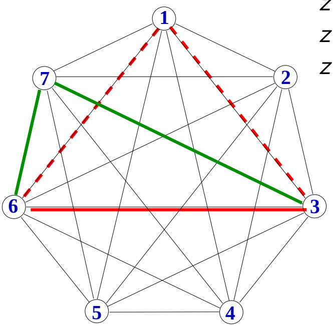
$$\begin{aligned}
z(h) &= && c\{6,5\}+c\{5,3\}-c\{6,3\}\\
z(h) &= && \min\{z(h),\ c\{6,4\}+c\{4,3\}-c\{6,3\}\}\\
z(h) &= && \min\{z(h),\ c\{6,2\}+c\{2,3\}-c\{6,3\}\}\\
z(h) &= && \min\{z(h),\ c\{6,7\}+c\{7,3\}-c\{6,3\}\}
\end{aligned}$$

$$z(h)=\min\left\{\begin{array}{l} c\{6,5\}+c\{5,3\}-c\{6,3\},\\ c\{6,4\}+c\{4,3\}-c\{6,3\},\\ c\{6,2\}+c\{2,3\}-c\{6,3\},\\ c\{6,7\}+c\{7,3\}-c\{6,3\} \end{array}\right\}$$

Poznámka

Algoritmus vkladacia heuristika vytvára postupne cykly tak,
že do súčasného cyklu vkladná taký vrchol, ktorým ho najmenej predĺži.
Je to typická vytvárajúca heuristika.

Vkladacia heuristika pre TSP



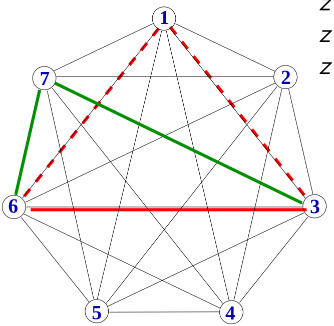
$$z(h) = c\{6,5\} + c\{5,3\} - c\{6,3\}$$
$$z(h) = \min\big\{z(h),\ c\{6,4\} + c\{4,3\} - c\{6,3\}\big\}$$
$$z(h) = \min\big\{z(h),\ c\{6,2\} + c\{2,3\} - c\{6,3\}\big\}$$
$$z(h) = \min\big\{z(h),\ c\{6,7\} + c\{7,3\} - c\{6,3\}\big\}$$

$$z(h) = \min \left\{ \begin{array}{l} c\{6,5\} + c\{5,3\} - c\{6,3\}, \\ c\{6,4\} + c\{4,3\} - c\{6,3\}, \\ c\{6,2\} + c\{2,3\} - c\{6,3\}, \\ c\{6,7\} + c\{7,3\} - c\{6,3\} \end{array} \right\}$$

Poznámka

Algoritmus vkladacia heuristika vytvára postupne cykly tak, že do súčasného cyklu vkladná taký vrchol, ktorým ho najmenej predĺži. Je to typická vytvárajúca heuristika.

Vkladacia heuristika pre TSP



$$
\begin{aligned}
z(h) &= & c\{6,5\} + c\{5,3\} - c\{6,3\} \\
z(h) &= & \min\{z(h),\ c\{6,4\} + c\{4,3\} - c\{6,3\}\} \\
z(h) &= & \min\{z(h),\ c\{6,2\} + c\{2,3\} - c\{6,3\}\} \\
z(h) &= & \min\{z(h),\ c\{6,7\} + c\{7,3\} - c\{6,3\}\}
\end{aligned}
$$

$$
z(h) = \min \left\{ \begin{array}{l} c\{6,5\} + c\{5,3\} - c\{6,3\}, \\ c\{6,4\} + c\{4,3\} - c\{6,3\}, \\ c\{6,2\} + c\{2,3\} - c\{6,3\}, \\ c\{6,7\} + c\{7,3\} - c\{6,3\} \end{array} \right\}
$$

Poznámka

Algoritmus vkladacia heuristika vytvára postupne cykly tak,
že do súčasného cyklu vkladná taký vrchol, ktorým ho najmenej predĺži.
Je to typická vytvárajúca heuristika.

Algoritmus prehľadávania okolí

Algoritmus

Algoritmus prehľadávania okolí.
Ku každému riešeniu – hamiltonovskému cyklu C – definujeme jeho okolie $\mathcal{O}(C)$ ako množinu hamiltonovských cyklov, ktorú z cyklu C dostaneme nejakými operáciami.
Označme $c(C)$ cenu hamiltonovského cyklu C.

- **Krok 1.** *Za počiatočný hamiltonovský cyklus C vezmi ľubovoľný hamiltonovský cyklus (dostať ho môžeš náhodným generátorom alebo ako výsledok niektorej vytvárajúcej heuristiky).*
- **Krok 2.** *Hľadaj $C' \in \mathcal{O}(C)$ také, že $c(C') < c(C)$.*

  Ak pre všetky $C' \in \mathcal{O}(C)$ $c(C') \geq c(C)$, STOP, C je suboptimálny hamiltonovský cyklus.

  Inak pokračuj krokom 3.
- **Krok 3.** *Vezmi $C' \in \mathcal{O}(C)$ také, že $c(C') < c(C)$ a polož $C := C'$. Goto Krok 2.*

♣

Algoritmus prehľadávania okolí

Algoritmus

Algoritmus prehľadávania okolí.
Ku každému riešeniu – hamiltonovskému cyklu C *– definujeme jeho okolie* $\mathcal{O}(C)$ *ako množinu hamiltonovských cyklov, ktorú z cyklu* C *dostaneme nejakými operáciami.*
Označme $c(C)$ *cenu hamiltonovského cyklu* C.

- **Krok 1.** *Za počiatočný hamiltonovský cyklus* C *vezmi ľubovoľný hamiltonovský cyklus (dostať ho môžeš náhodným generátorom alebo ako výsledok niektorej vytvárajúcej heuristiky).*
- **Krok 2.** *Hľadaj* $C' \in \mathcal{O}(C)$ *také, že* $c(C') < c(C)$.

  Ak pre všetky $C' \in \mathcal{O}(C)$ $c(C') \geq c(C)$, *STOP,* C *je suboptimálny hamiltonovský cyklus.*

  Inak pokračuj krokom 3.
- **Krok 3.** *Vezmi* $C' \in \mathcal{O}(C)$ *také, že* $c(C') < c(C)$ *a polož* $C := C'$. *Goto Krok 2.*

♣

Algoritmus prehľadávania okolí

Algoritmus

Algoritmus prehľadávania okolí.

Ku každému riešeniu – hamiltonovskému cyklu C – definujeme jeho okolie $\mathcal{O}(C)$ ako množinu hamiltonovských cyklov, ktorú z cyklu C dostaneme nejakými operáciami.

Označme $c(C)$ cenu hamiltonovského cyklu C.

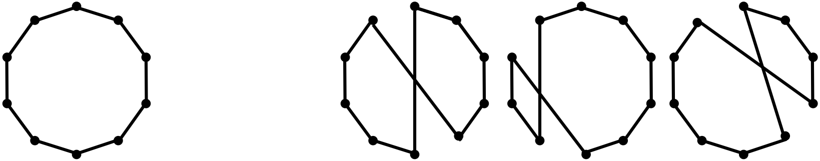
- **Krok 1.** *Za počiatočný hamiltonovský cyklus C vezmi ľubovoľný hamiltonovský cyklus (dostať ho môžeš náhodným generátorom alebo ako výsledok niektorej vytvárajúcej heuristiky).*
- **Krok 2.** *Hľadaj $C' \in \mathcal{O}(C)$ také, že $c(C') < c(C)$.*

  Ak pre všetky $C' \in \mathcal{O}(C)$ $c(C') \geq c(C)$, STOP, C je suboptimálny hamiltonovský cyklus.

  Inak pokračuj krokom 3.
- **Krok 3.** *Vezmi $C' \in \mathcal{O}(C)$ také, že $c(C') < c(C)$ a polož $C := C'$. Goto Krok 2.*

♣

Algoritmus prehľadávania okolí

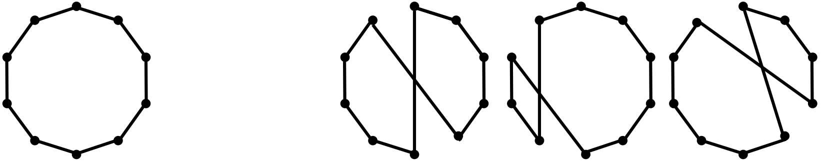


Cyklus C a niekoľko prvkov jeho okolia.

Nebezpečenstvo algoritmu prehľadávania okolí:
Algoritmus uviazne v takom zlom riešení, v okolí ktorého niet lepšieho riešenia.

Riešenie:
Viacnásobné spustenie algoritmu s rôznymi štartovacími riešeniami.

Algoritmus prehľadávania okolí



Cyklus C a niekoľko prvkov jeho okolia.

Nebezpečenstvo algoritmu prehľadávania okolí:
Algoritmus uviazne v takom zlom riešení, v okolí ktorého niet lepšieho riešenia.

Riešenie:
Viacnásobné spustenie algoritmu s rôznymi štartovacími riešeniami.