Acyklické grafy, stromy a kostry

Stanislav Palúch

Fakulta riadenia a informatiky, Žilinská univerzita

18. mája 2020

Opakovanie – Cyklus, kružnica

Definícia

Cyklus (orientovaný cyklus, polocyklus) *je netriviálny uzavretý ťah (orientovaný ťah, poloťah), v ktorom sa okrem prvého a posledného vrchola žiaden vrchol nevyskytuje viac než raz.*

Definícia

Kružnica *je súvislý pravidelný graf 2. stupňa. Kružnicu o n vrcholoch budeme označovať C_n.*

Poznámka

Všetky vrcholy a hrany kružnice C_n možno usporiadať do cyklu

$$(v_1, \{v_1, v_2\}, v_2, \dots, \{v_{n-1}, v_n\}, v_n, \{v_n, v_1\}, v_1\ )$$

a naopak,
všetky vrcholy a hrany cyklu tvoria graf, ktorý je kružnicou.

Opakovanie – Cyklus, kružnica

Definícia

Cyklus (orientovaný cyklus, polocyklus) *je netriviálny uzavretý ťah (orientovaný ťah, poloťah), v ktorom sa okrem prvého a posledného vrchola žiaden vrchol nevyskytuje viac než raz.*

Definícia

Kružnica *je súvislý pravidelný graf 2. stupňa. Kružnicu o n vrcholoch budeme označovať C_n.*

Poznámka

Všetky vrcholy a hrany kružnice C_n možno usporiadať do cyklu

$$(v_1, \{v_1, v_2\}, v_2, \dots, \{v_{n-1}, v_n\}, v_n, \{v_n, v_1\}, v_1\ )$$

a naopak,
všetky vrcholy a hrany cyklu tvoria graf, ktorý je kružnicou.

Opakovanie – Cyklus, kružnica

Definícia

Cyklus (orientovaný cyklus, polocyklus) *je* *netriviálny* *uzavretý ťah (orientovaný ťah, poloťah), v ktorom sa okrem prvého a posledného vrchola žiaden vrchol nevyskytuje viac než raz.*

Definícia

Kružnica *je súvislý pravidelný graf 2. stupňa. Kružnicu o n vrcholoch budeme označovať* C_n.

Poznámka

Všetky vrcholy a hrany kružnice C_n *možno usporiadať do cyklu*

$$(v_1, \{v_1, v_2\}, v_2, \ldots, \{v_{n-1}, v_n\}, v_n, \{v_n, v_1\}, v_1\ )$$

a naopak,
všetky vrcholy a hrany cyklu tvoria graf, ktorý je kružnicou.

Definícia acyklického grafu a stromu

Definícia

Acyklický graf *je taký graf, ktorý neobsahuje ako podgraf kružnicu.*

Definícia

Strom *je súvislý acyklický graf.*

Poznámka

Triviálny graf je stromom.

Poznámka

Pretože každý komponent acyklického grafu je stromom (je súvislý a neobsahuje kružnicu), možno sa na acyklický graf pozerať ako na zjednotenie stromov. Od toho je odvodený pojem les, ktorý sa používa ako synonymum pre acyklické grafy.

Definícia acyklického grafu a stromu

Definícia

Acyklický graf *je taký graf, ktorý neobsahuje ako podgraf kružnicu.*

Definícia

Strom *je súvislý acyklický graf.*

Poznámka

Triviálny graf je stromom.

Poznámka

Pretože každý komponent acyklického grafu je stromom (je súvislý a neobsahuje kružnicu), možno sa na acyklický graf pozerať ako na zjednotenie stromov. Od toho je odvodený pojem les, ktorý sa používa ako synonymum pre acyklické grafy.

Definícia acyklického grafu a stromu

Definícia

Acyklický graf *je taký graf, ktorý neobsahuje ako podgraf kružnicu.*

Definícia

Strom *je súvislý acyklický graf.*

Poznámka

Triviálny graf je stromom.

Poznámka

Pretože každý komponent acyklického grafu je stromom (je súvislý a neobsahuje kružnicu), možno sa na acyklický graf pozerať ako na zjednotenie stromov. Od toho je odvodený pojem les, ktorý sa používa ako synonymum pre acyklické grafy.

Definícia acyklického grafu a stromu

Definícia

Acyklický graf *je taký graf, ktorý neobsahuje ako podgraf kružnicu.*

Definícia

Strom *je súvislý acyklický graf.*

Poznámka

Triviálny graf je stromom.

Poznámka

Pretože každý komponent acyklického grafu je stromom (je súvislý a neobsahuje kružnicu), možno sa na acyklický graf pozerať ako na zjednotenie stromov. Od toho je odvodený pojem les, ktorý sa používa ako synonymum pre acyklické grafy.

V netriviálnom strome existujú aspoň dva vrcholy stupňa 1

Veta

Nech $G = (V, H)$ je strom, ktorý má aspoň dva vrcholy.
Potom V obsahuje aspoň dva vrcholy stupňa 1.

DÔKAZ.
Nech

$$(v_1, \{v_1, v_2\}, v_2, \dots, \{v_{k-1}, v_k\}, v_k) \tag{1}$$

je cesta v strome G s najväčším počtom hrán. Ukážeme, že $\deg(v_k) = 1$.

Obr.: Keby $\deg(v_k) > 1$,

existovala by aspoň jedna hrana (čiarkovane) incidentná s v_k,
vytvárajúca jednu zo situácií a) alebo b).

V netriviálnom strome existujú aspoň dva vrcholy stupňa 1

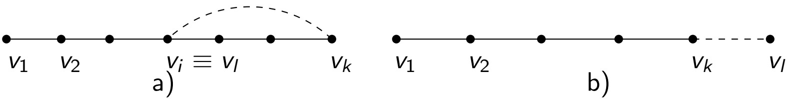
Veta

Nech $G = (V, H)$ je strom, ktorý má aspoň dva vrcholy.
Potom V obsahuje aspoň dva vrcholy stupňa 1.

DÔKAZ.
Nech

$$(v_1, \{v_1, v_2\}, v_2, \dots, \{v_{k-1}, v_k\}, v_k) \tag{1}$$

je cesta v strome G s najväčším počtom hrán. Ukážeme, že $\deg(v_k) = 1$.



Obr.: Keby $\deg(v_k) > 1$,

existovala by aspoň jedna hrana (čiarkovane) incidentná s v_k,
vytvárajúca jednu zo situácií a) alebo b).

Vlastnosti stromov

Veta

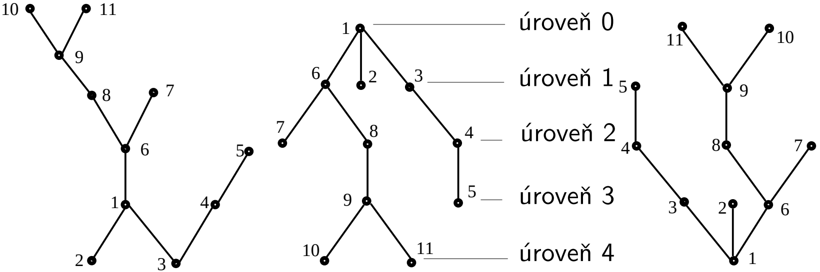
Nasledujúce tvrdenia sú ekvivalentné:

- *a)* $G = (V, H)$ *je strom.*
- *b) V grafe* $G = (V, H)$ *existuje pre každé* $u, v \in V$ *jediná* u–v *cesta.*
- *c) Graf* $G = (V, H)$ *je súvislý a každá hrana množiny* H *je mostom.*
- *d) Graf* $G = (V, H)$ *je súvislý a* $|H| = |V| - 1$.
- *e) V grafe* $G = (V, H)$ *platí* $|H| = |V| - 1$ *a* G *je acyklický.*

Koreňový strom

Definícia

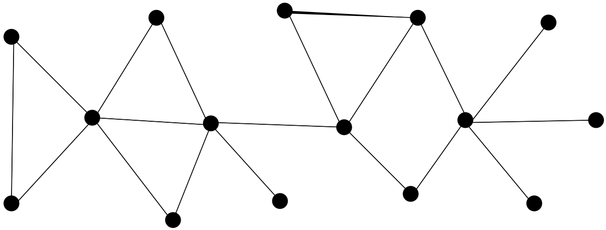
Koreňový strom *je strom* $G = (V, H)$ *s pevne vybraným vrcholom* $k \in V$*, ktorý nazývame* **koreň**. *Koreňový strom budeme značiť* $G = (V, H, k)$.
Úroveň vrchola u *v koreňovom strome* $G = (V, H, k)$ *je dĺžka – počet hrán – (jedinej)* k*–*u *cesty.*
Výška koreňového stromu $G = (V, H, k)$ *je maximum z úrovní všetkých vrcholov koreňového stromu* G.



Obr.: Spôsoby kreslenia diagramu koreňového stromu s koreňom 1.

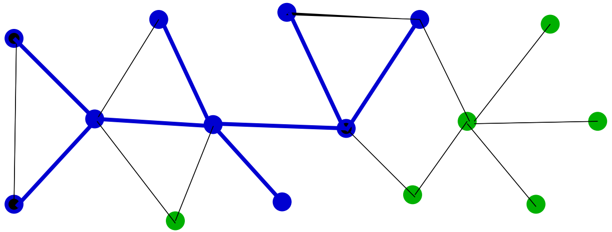
Definícia

Nech strom $T = (V_T, H_T)$ je podgrafom grafu $G = (V, H)$. Hovoríme, že hrana $h = \{u, v\} \in H$ je **hraničnou hranou***, ak $u \in V_T$ a $v \notin V_T$.*
Nech $h = \{u, v\}$ je hraničná hrana, $u \in V_T$, $v \notin V_T$. Povieme, že u je **zaradený vrchol***, v je* **voľný vrchol** *hraničnej hrany h.*



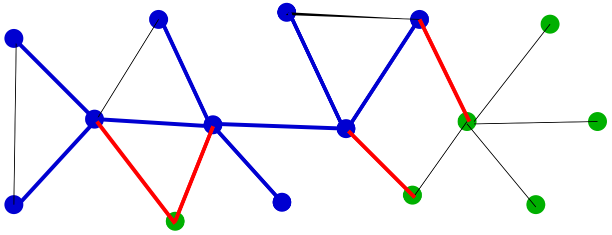
Definícia

Nech strom $T = (V_T, H_T)$ je podgrafom grafu $G = (V, H)$. Hovoríme, že hrana $h = \{u, v\} \in H$ je **hraničnou hranou**, *ak $u \in V_T$ a $v \notin V_T$. Nech $h = \{u, v\}$ je hraničná hrana, $u \in V_T$, $v \notin V_T$. Povieme, že u je* **zaradený vrchol**, *v je* **voľný vrchol** *hraničnej hrany h.*



Definícia

Nech strom $T = (V_T, H_T)$ je podgrafom grafu $G = (V, H)$. Hovoríme, že hrana $h = \{u, v\} \in H$ je **hraničnou hranou**, *ak $u \in V_T$ a $v \notin V_T$. Nech $h = \{u, v\}$ je hraničná hrana, $u \in V_T$, $v \notin V_T$. Povieme, že u je* **zaradený vrchol**, *v je* **voľný vrchol** *hraničnej hrany h.*



Algoritmus

Prehľadávanie grafu $G = (V, H)$ do hĺbky. (Depth-First Search)

- **Krok 1. Inicializácia.**
  Nech strom T je triviálny strom obsahujúci jediný vrchol $v \in V$.
  Polož $p(v) := 1$, $k := 1$.
- **Krok 2.** *Ak T ešte neobsahuje všetky vrcholy grafu, GOTO Krok 3. Inak STOP.*
- **Krok 3.** *V grafe G so stromom T nájdi hraničnú hranu $h = \{u, v\}$* **s maximálnou značkou** *$p(u)$ zaradeného vrchola u.*
- **Krok 4.** *Polož $T := T \cup \{h\} \cup \{v\}$, $\quad k := k + 1$, $\quad p(v) := k$.*
  GOTO Krok 2.

♣

Algoritmus

Prehľadávanie grafu $G = (V, H)$ do hĺbky. (Depth-First Search)

- **Krok 1. Inicializácia.**
  Nech strom T je triviálny strom obsahujúci jediný vrchol $v \in V$.
  Polož $p(v) := 1$, $k := 1$.
- **Krok 2.** *Ak T ešte neobsahuje všetky vrcholy grafu, GOTO Krok 3.*
  Inak STOP.
- **Krok 3.** *V grafe G so stromom T nájdi hraničnú hranu $h = \{u, v\}$* **s maximálnou značkou** *$p(u)$ zaradeného vrchola u.*
- **Krok 4.** *Polož $T := T \cup \{h\} \cup \{v\}$, $\quad k := k + 1$, $\quad p(v) := k$.*
  GOTO Krok 2.

♣

Algoritmus

Prehľadávanie grafu $G = (V, H)$ do hĺbky. (Depth-First Search)

- **Krok 1. Inicializácia.**
  Nech strom T je triviálny strom obsahujúci jediný vrchol $v \in V$.
  Polož $p(v) := 1$, $k := 1$.
- **Krok 2.** *Ak T ešte neobsahuje všetky vrcholy grafu, GOTO Krok 3. Inak STOP.*
- **Krok 3.** *V grafe G so stromom T nájdi hraničnú hranu $h = \{u, v\}$* **s maximálnou značkou** *$p(u)$ zaradeného vrchola u.*
- **Krok 4.** *Polož $T := T \cup \{h\} \cup \{v\}$, $k := k + 1$, $p(v) := k$.*
  GOTO Krok 2.

♣

Algoritmus

Prehľadávanie grafu $G = (V, H)$ **do hĺbky. (Depth-First Search)**

- **Krok 1. Inicializácia.**
  Nech strom T *je triviálny strom obsahujúci jediný vrchol* $v \in V$.
  Polož $p(v) := 1$, $k := 1$.
- **Krok 2.** *Ak* T *ešte neobsahuje všetky vrcholy grafu, GOTO Krok 3. Inak STOP.*
- **Krok 3.** *V grafe* G *so stromom* T *nájdi hraničnú hranu* $h = \{u, v\}$ **s maximálnou značkou** $p(u)$ *zaradeného vrchola* u.
- **Krok 4.** *Polož* $T := T \cup \{h\} \cup \{v\}$, $\quad k := k + 1$, $\quad p(v) := k$.
  GOTO Krok 2.

♣

Prehľadávanie grafu do šírky – Breadth-First Search

Algoritmus

Prehľadávanie grafu $G = (V, H)$ do šírky. (Breadth-First Search.)

- **Krok 1.** *Inicializácia. Nech strom T je triviálny strom obsahujúci jediný vrchol $v \in V$. Polož $p(v) := 1$, $k := 1$.*
- **Krok 2.** *Ak T ešte neobsahuje všetky vrcholy grafu, GOTO Krok 3. Inak STOP.*
- **Krok 3.** *V grafe G so stromom T nájdi hraničnú hranu $h = \{u, v\}$* **s minimálnou značkou** *$p(u)$ zaradeného vrchola u.*
- **Krok 4.** *Polož $T := T \cup \{h\} \cup \{v\}$, $\quad k := k + 1$, $\quad p(v) := k$. GOTO Krok 2.*

♣

Prehľadávanie grafu do šírky – Breadth-First Search

Algoritmus

Prehľadávanie grafu $G = (V, H)$ do šírky. (Breadth-First Search.)

- **Krok 1.** *Inicializácia. Nech strom T je triviálny strom obsahujúci jediný vrchol $v \in V$. Polož $p(v) := 1$, $k := 1$.*
- **Krok 2.** *Ak T ešte neobsahuje všetky vrcholy grafu, GOTO Krok 3. Inak STOP.*
- **Krok 3.** *V grafe G so stromom T nájdi hraničnú hranu $h = \{u, v\}$* **s minimálnou značkou** *$p(u)$ zaradeného vrchola u.*
- **Krok 4.** *Polož $T := T \cup \{h\} \cup \{v\}$, $\quad k := k + 1$, $\quad p(v) := k$. GOTO Krok 2.*

♣

Prehľadávanie grafu do šírky – Breadth-First Search

Algoritmus

Prehľadávanie grafu $G = (V, H)$ **do šírky. (Breadth-First Search.)**

- **Krok 1.** *Inicializácia. Nech strom* T *je triviálny strom obsahujúci jediný vrchol* $v \in V$*. Polož* $p(v) := 1$, $k := 1$.
- **Krok 2.** *Ak* T *ešte neobsahuje všetky vrcholy grafu, GOTO Krok 3. Inak STOP.*
- **Krok 3.** *V grafe* G *so stromom* T *nájdi hraničnú hranu* $h = \{u, v\}$ **s minimálnou značkou** $p(u)$ *zaradeného vrchola* u.
- **Krok 4.** *Polož* $T := T \cup \{h\} \cup \{v\}$, $k := k + 1$, $p(v) := k$. *GOTO Krok 2.*

♣

Prehľadávanie grafu do šírky – Breadth-First Search

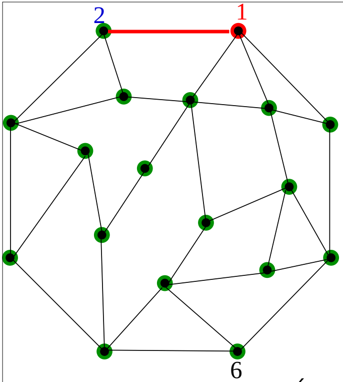
Algoritmus

Prehľadávanie grafu $G = (V, H)$ do šírky. (Breadth-First Search.)

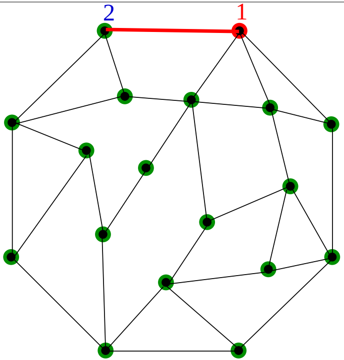
- **Krok 1.** *Inicializácia. Nech strom T je triviálny strom obsahujúci jediný vrchol $v \in V$. Polož $p(v) := 1$, $k := 1$.*
- **Krok 2.** *Ak T ešte neobsahuje všetky vrcholy grafu, GOTO Krok 3. Inak STOP.*
- **Krok 3.** *V grafe G so stromom T nájdi hraničnú hranu $h = \{u, v\}$* **s minimálnou značkou** *$p(u)$ zaradeného vrchola u.*
- **Krok 4.** *Polož $T := T \cup \{h\} \cup \{v\}$, $\quad k := k + 1$, $\quad p(v) := k$. GOTO Krok 2.*

♣

Prehľadávanie grafu do hĺbky a do šírky

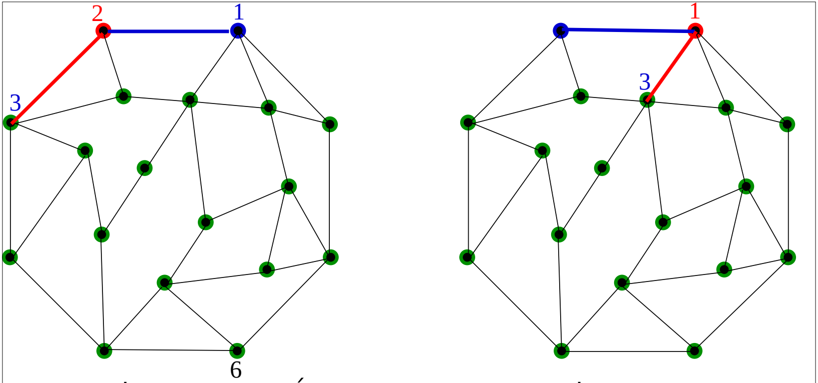


Prehľadávanie do hĺbky



Prehľadávanie do šírky

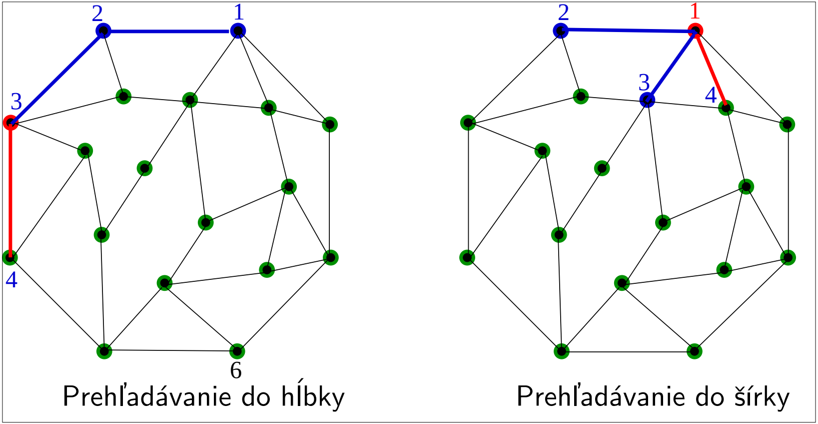
Prehľadávanie grafu do hĺbky a do šírky



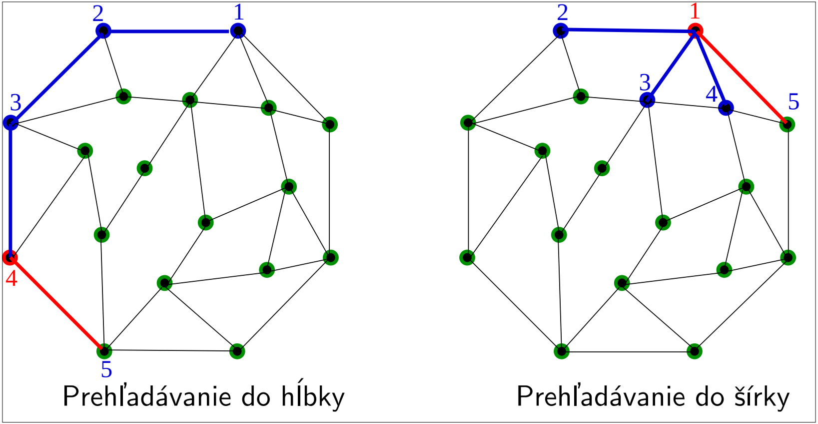
Prehľadávanie do hĺbky

Prehľadávanie do šírky

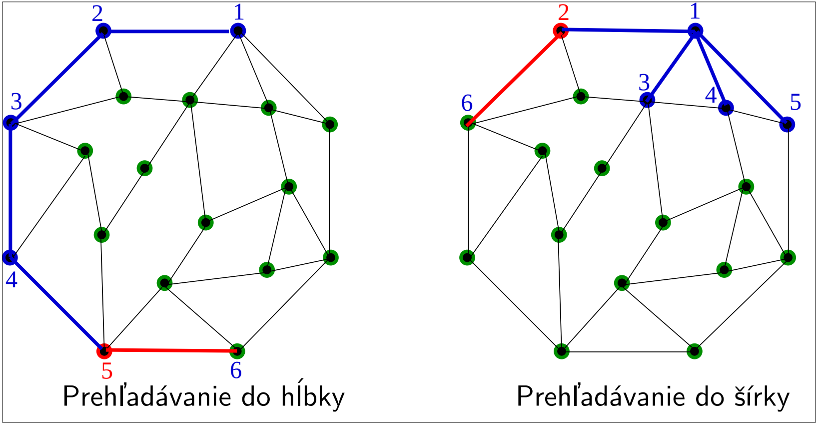
Prehľadávanie grafu do hĺbky a do šírky



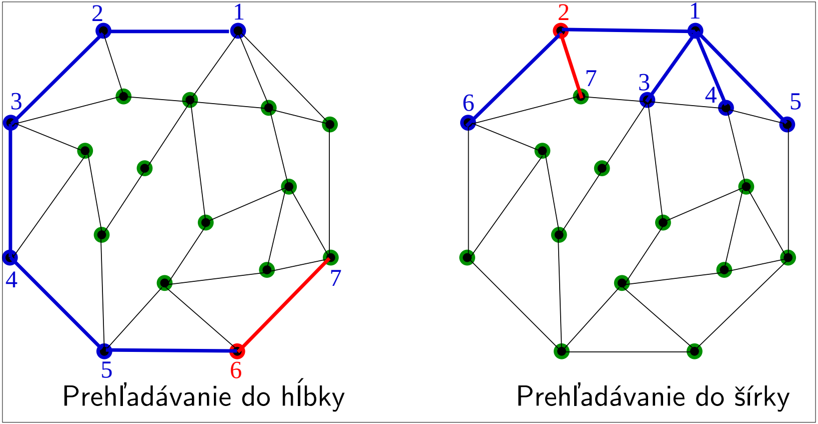
Prehľadávanie grafu do hĺbky a do šírky



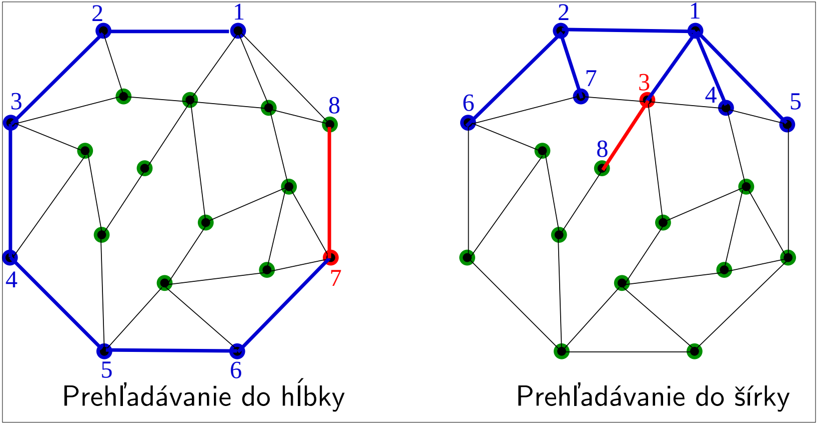
Prehľadávanie grafu do hĺbky a do šírky



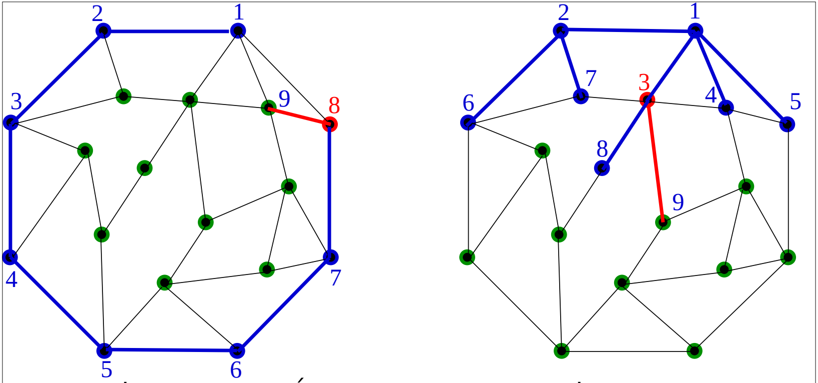
Prehľadávanie grafu do hĺbky a do šírky



Prehľadávanie grafu do hĺbky a do šírky

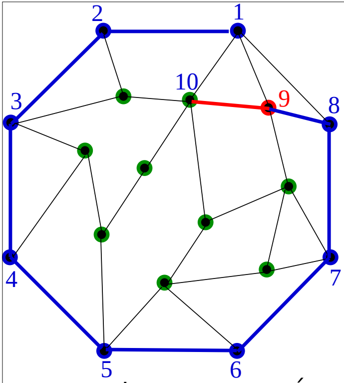


Prehľadávanie grafu do hĺbky a do šírky

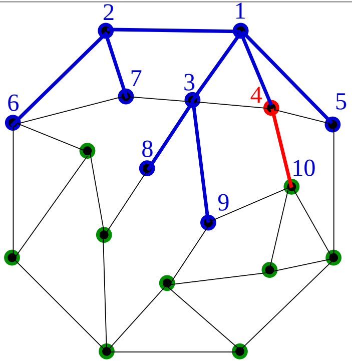


Prehľadávanie do hĺbky

Prehľadávanie do šírky

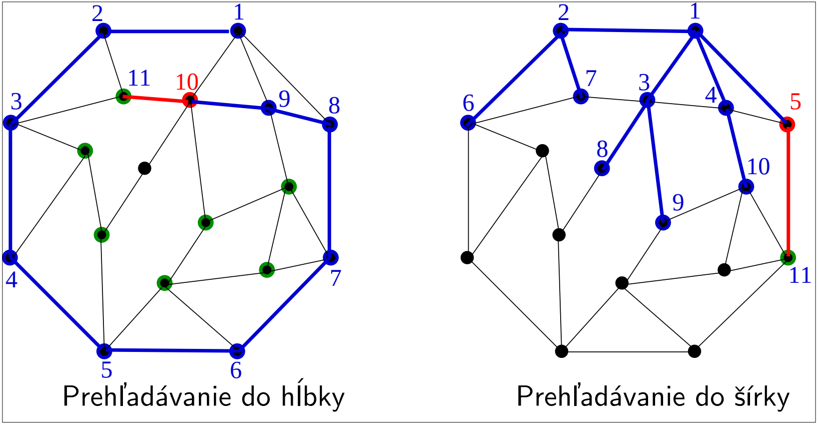


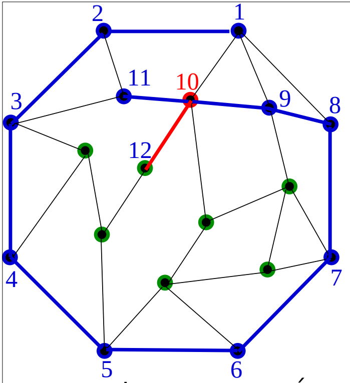
Prehľadávanie do hĺbky



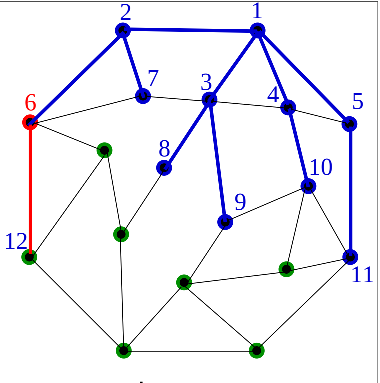
Prehľadávanie do šírky

Prehľadávanie grafu do hĺbky a do šírky



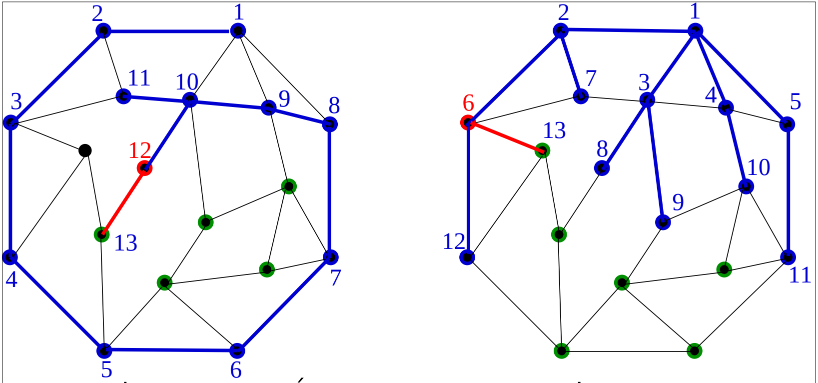


Prehľadávanie do hĺbky



Prehľadávanie do šírky

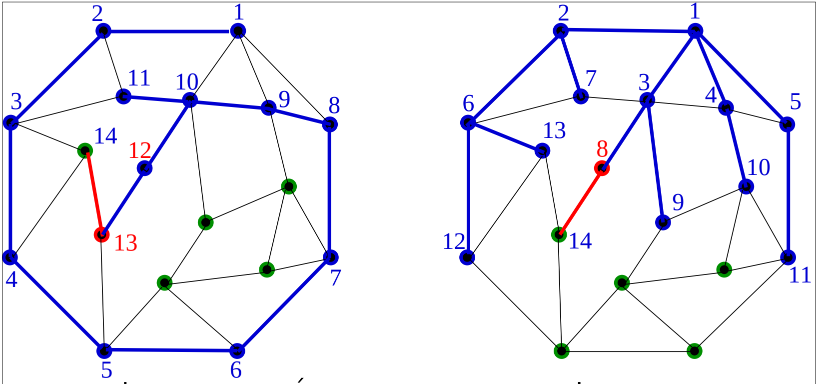
Prehľadávanie grafu do hĺbky a do šírky



Prehľadávanie do hĺbky

Prehľadávanie do šírky

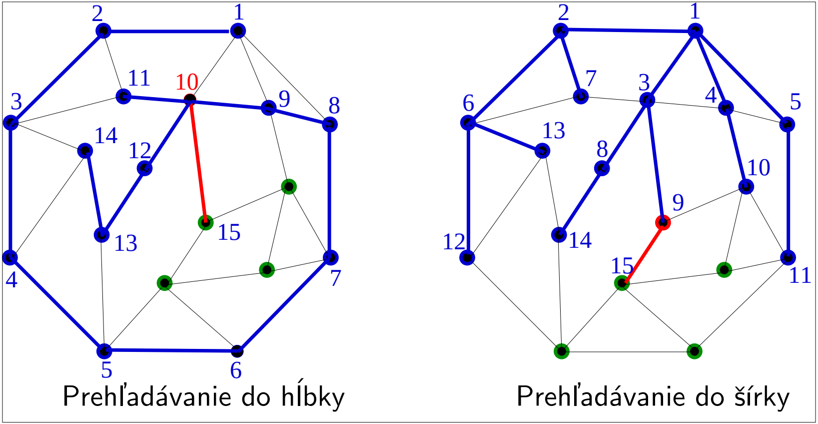
Prehľadávanie grafu do hĺbky a do šírky

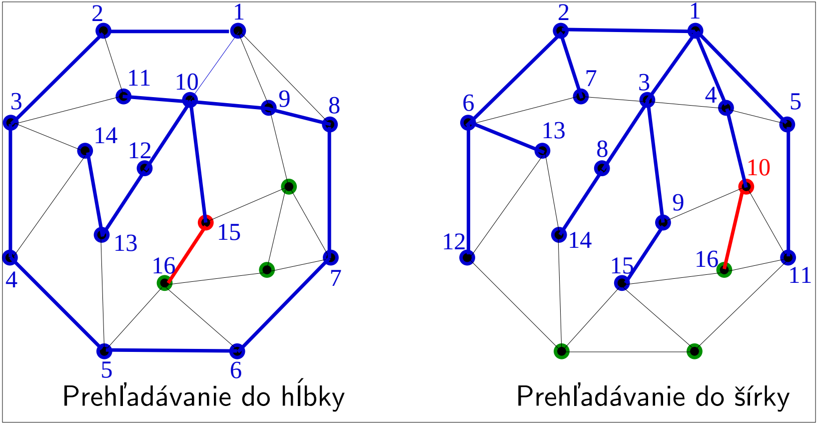


Prehľadávanie do hĺbky

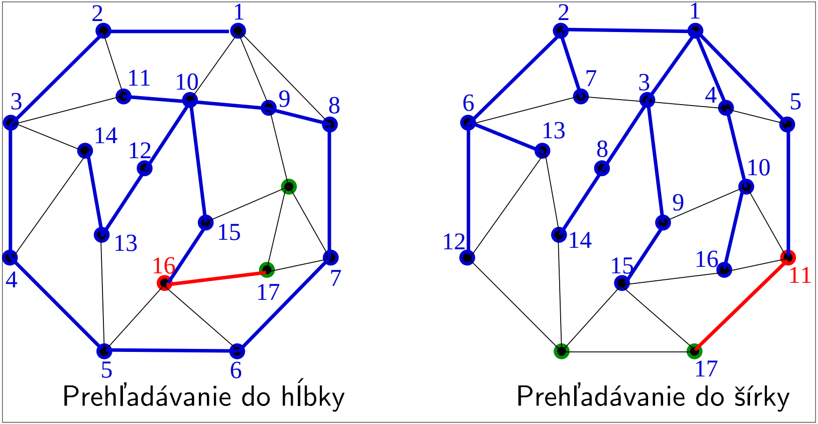
Prehľadávanie do šírky

Prehľadávanie grafu do hĺbky a do šírky

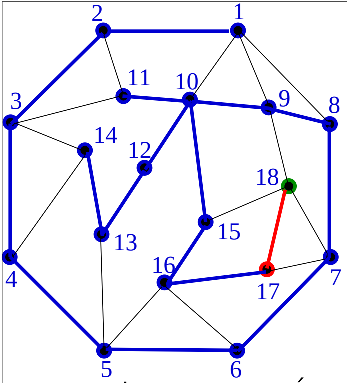



2
1
11
10
9
8
3
14
12
15
13
16
4
7
5
6
Prehľadávanie do hĺbky
2
1
7
3
4
5
6
13
8
10
9
12
14
16
15
11
Prehľadávanie do šírky

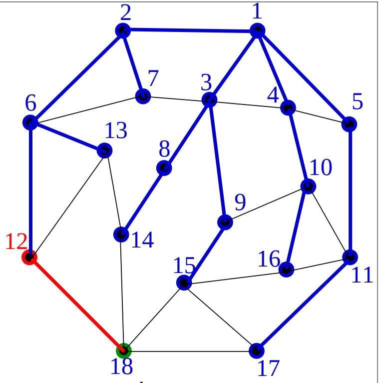
Prehľadávanie grafu do hĺbky a do šírky



Prehľadávanie grafu do hĺbky a do šírky



Prehľadávanie do hĺbky



Prehľadávanie do šírky

Najlacnejšia a najdrahšia kostra

Definícia

Kostra *súvislého grafu* $G = (V, H)$ *je taký jeho faktorový podgraf, ktorý je stromom.*

Nech $G = (V, H, c)$ *je hranovo ohodnotený graf,* K *kostra grafu* G.
Cena $c(K)$ **kostry** K *je súčet ohodnotení jej hrán.*

Najlacnejšia kostra *v grafe* G *je kostra s najmenšou cenou.*

Najdrahšia kostra *v grafe* G *je kostra s najväčšou cenou.*

Najlacnejšia a najdrahšia kostra

Definícia

Kostra *súvislého grafu* $G = (V, H)$ *je taký jeho faktorový podgraf, ktorý je stromom.*

Nech $G = (V, H, c)$ *je hranovo ohodnotený graf,* K *kostra grafu* G.
Cena $c(K)$ **kostry** K *je súčet ohodnotení jej hrán.*

Najlacnejšia kostra *v grafe* G *je kostra s najmenšou cenou.*

Najdrahšia kostra *v grafe* G *je kostra s najväčšou cenou.*

Kruskalov algoritmus I.

Algoritmus

Kruskalov algoritmus I. *na hľadanie najlacnejšej (najdrahšej) kostry súvislého hranovo ohodnoteného grafu* $G = (V, H, c)$.

- **Krok 1.** *Zoraď hrany podľa ich ohodnotenia vzostupne (zostupne) do postupnosti* $\mathcal{P}$.
- **Krok 2.** *Nech prvá hrana v postupnosti* $\mathcal{P}$ *je hrana* $\{u, v\}$. *Vylúč hranu* $\{u, v\}$ *z postupnosti* $\mathcal{P}$ *a ak s už vybranými hranami nevytvára cyklus, zaraď ju do kostry.*
- **Krok 3.** *Ak je počet vybraných hrán rovný* $|V| - 1$ *alebo ak je postupnosť* $\mathcal{P}$ *prázdna, STOP. Inak GOTO Krok 2.*

♣

Kruskalov algoritmus I.

Algoritmus

Kruskalov algoritmus I. *na hľadanie najlacnejšej (najdrahšej) kostry súvislého hranovo ohodnoteného grafu* $G = (V, H, c)$.

- **Krok 1.** *Zoraď hrany podľa ich ohodnotenia vzostupne (zostupne) do postupnosti* $\mathcal{P}$.
- **Krok 2.** *Nech prvá hrana v postupnosti* $\mathcal{P}$ *je hrana* $\{u, v\}$. *Vylúč hranu* $\{u, v\}$ *z postupnosti* $\mathcal{P}$ *a ak s už vybranými hranami nevytvára cyklus, zaraď ju do kostry.*
- **Krok 3.** *Ak je počet vybraných hrán rovný* $|V| - 1$ *alebo ak je postupnosť* $\mathcal{P}$ *prázdna, STOP. Inak GOTO Krok 2.*

♣

Kruskalov algoritmus I.

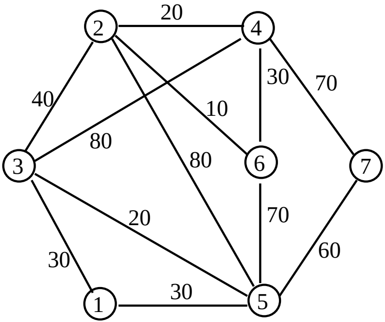
Algoritmus

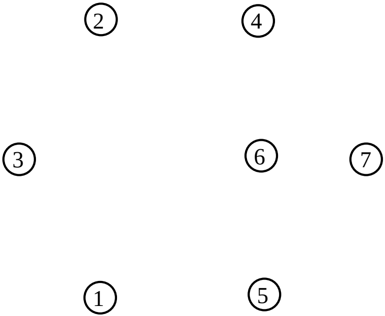
Kruskalov algoritmus I. *na hľadanie najlacnejšej (najdrahšej) kostry súvislého hranovo ohodnoteného grafu $G = (V, H, c)$.*

- **Krok 1.** *Zoraď hrany podľa ich ohodnotenia vzostupne (zostupne) do postupnosti $\mathcal{P}$.*
- **Krok 2.** *Nech prvá hrana v postupnosti $\mathcal{P}$ je hrana $\{u, v\}$. Vylúč hranu $\{u, v\}$ z postupnosti $\mathcal{P}$ a ak s už vybranými hranami nevytvára cyklus, zaraď ju do kostry.*
- **Krok 3.** *Ak je počet vybraných hrán rovný $|V| - 1$ alebo ak je postupnosť $\mathcal{P}$ prázdna, STOP. Inak GOTO Krok 2.*

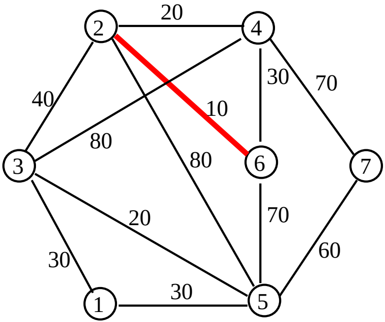
♣

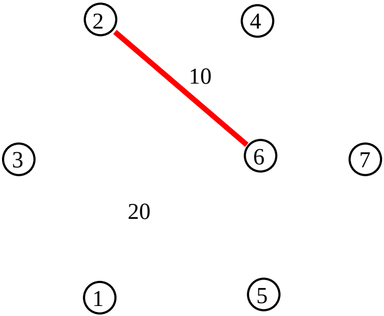
Príklad



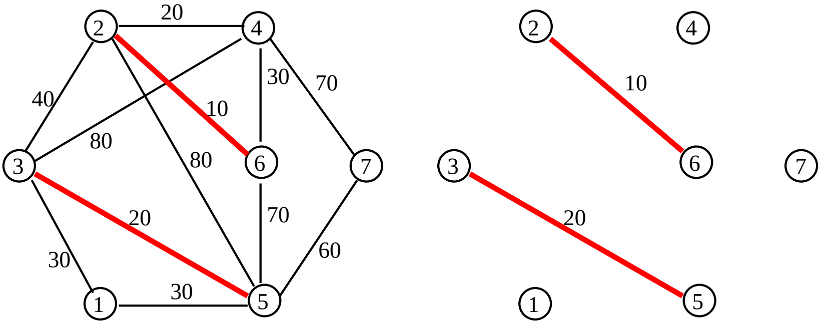


Príklad

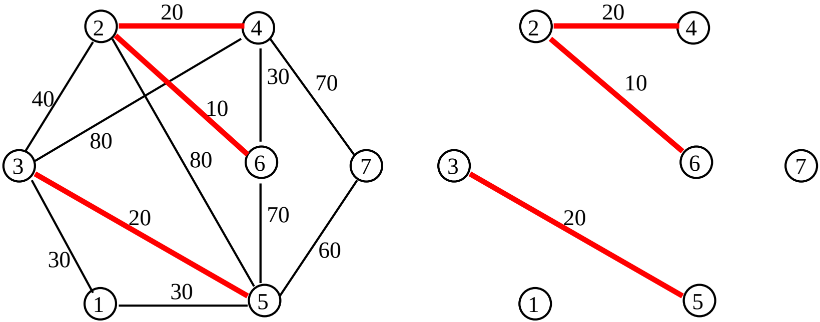




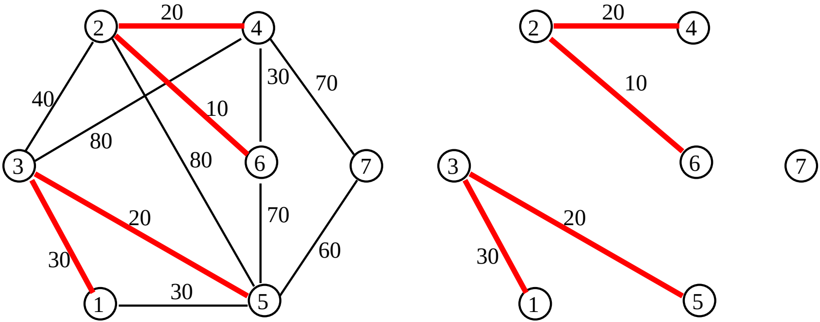
Príklad



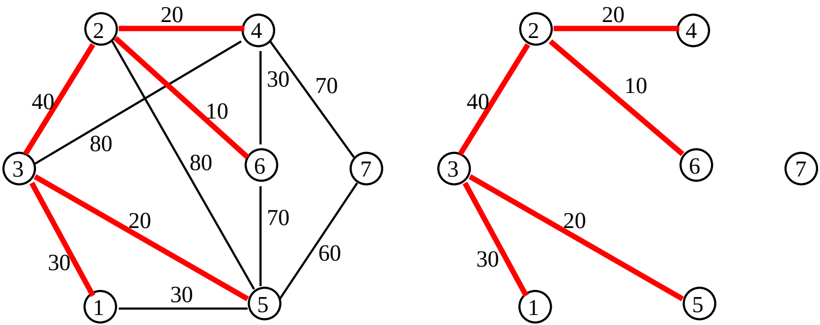
Príklad



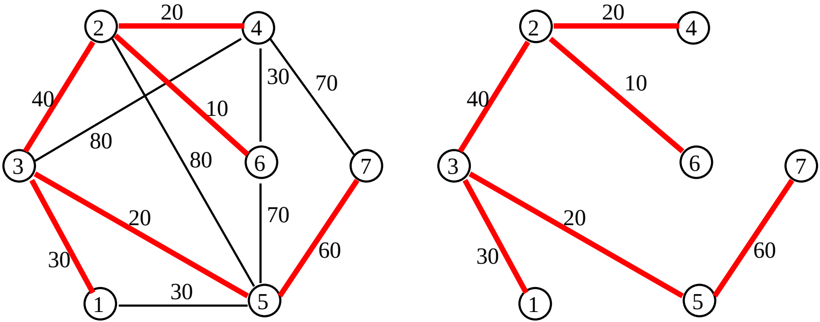
Príklad



Príklad



Príklad



20
40
10
30
70
80
80
20
70
30
60
30
1
2
3
4
5
6
7
20
40
10
20
30
60
1
2
3
4
5
6
7

Algoritmus

Kruskalov algoritmus II. *na hľadanie najlacnejšej (najdrahšej) kostry súvislého hranovo ohodnoteného grafu* $G = (V, H, c)$.

- **Krok 1.** *Zoraď hrany podľa ich ohodnotenia vzostupne (zostupne) do postupnosti* $\mathcal{P}$.
- **Krok 2.** *Pre každý vrchol* $i \in V$ *polož* $k(i) = i$.
- **Krok 3.** *Nech prvá hrana v postupnosti* $\mathcal{P}$ *je hrana* $\{u, v\}$. *Vylúč hranu* $\{u, v\}$ *z postupnosti* $\mathcal{P}$.
  Ak $k(u) \neq k(v)$, *zaraď hranu* $\{u, v\}$ *do kostry,*
  vypočítaj $kmin = \min(k(u), k(v))$, $kmax = \max(k(u), k(v))$,
  a $\forall i \in V$, *pre ktoré* $k(i) = kmax$, *polož* $k(i) := kmin$.
- **Krok 4.** *Ak je počet vybraných hrán rovný* $|V| - 1$ *alebo ak je postupnosť* $\mathcal{P}$ *prázdna, STOP. Inak GOTO krok 3.*

Kruskalov algoritmus II.

Algoritmus

Kruskalov algoritmus II. *na hľadanie najlacnejšej (najdrahšej) kostry súvislého hranovo ohodnoteného grafu* $G = (V, H, c)$.

- **Krok 1.** *Zoraď hrany podľa ich ohodnotenia vzostupne (zostupne) do postupnosti* $\mathcal{P}$.
- **Krok 2.** *Pre každý vrchol* $i \in V$ *polož* $k(i) = i$.
- **Krok 3.** *Nech prvá hrana v postupnosti* $\mathcal{P}$ *je hrana* $\{u, v\}$. *Vylúč hranu* $\{u, v\}$ *z postupnosti* $\mathcal{P}$.
  Ak $k(u) \neq k(v)$, *zaraď hranu* $\{u, v\}$ *do kostry,*
  vypočítaj $kmin = \min(k(u), k(v))$, $kmax = \max(k(u), k(v))$,
  a $\forall i \in V$, *pre ktoré* $k(i) = kmax$, *polož* $k(i) := kmin$.
- **Krok 4.** *Ak je počet vybraných hrán rovný* $|V| - 1$ *alebo ak je postupnosť* $\mathcal{P}$ *prázdna, STOP. Inak GOTO krok 3.*

♣

Algoritmus

Kruskalov algoritmus II. *na hľadanie najlacnejšej (najdrahšej) kostry súvislého hranovo ohodnoteného grafu* $G = (V, H, c)$.

- **Krok 1.** *Zoraď hrany podľa ich ohodnotenia vzostupne (zostupne) do postupnosti* $\mathcal{P}$.
- **Krok 2.** *Pre každý vrchol* $i \in V$ *polož* $k(i) = i$.
- **Krok 3.** *Nech prvá hrana v postupnosti* $\mathcal{P}$ *je hrana* $\{u, v\}$. *Vylúč hranu* $\{u, v\}$ *z postupnosti* $\mathcal{P}$.

  Ak $k(u) \neq k(v)$, *zaraď hranu* $\{u, v\}$ *do kostry,*

  vypočítaj $kmin = \min(k(u), k(v)), \quad kmax = \max(k(u), k(v))$,

  a $\forall i \in V$, *pre ktoré* $k(i) = kmax$, *polož* $k(i) := kmin$.
- **Krok 4.** *Ak je počet vybraných hrán rovný* $|V| - 1$ *alebo ak je postupnosť* $\mathcal{P}$ *prázdna, STOP. Inak GOTO krok 3.*

♣

Algoritmus

Kruskalov algoritmus II. *na hľadanie najlacnejšej (najdrahšej) kostry súvislého hranovo ohodnoteného grafu* $G = (V, H, c)$.

- **Krok 1.** *Zoraď hrany podľa ich ohodnotenia vzostupne (zostupne) do postupnosti* $\mathcal{P}$.
- **Krok 2.** *Pre každý vrchol* $i \in V$ *polož* $k(i) = i$.
- **Krok 3.** *Nech prvá hrana v postupnosti* $\mathcal{P}$ *je hrana* $\{u, v\}$. *Vylúč hranu* $\{u, v\}$ *z postupnosti* $\mathcal{P}$.

  Ak $k(u) \neq k(v)$, *zaraď hranu* $\{u, v\}$ *do kostry,*

  vypočítaj $kmin = \min(k(u), k(v)), \quad kmax = \max(k(u), k(v))$,

  a $\forall i \in V$, *pre ktoré* $k(i) = kmax$, *polož* $k(i) := kmin$.
- **Krok 4.** *Ak je počet vybraných hrán rovný* $|V| - 1$ *alebo ak je postupnosť* $\mathcal{P}$ *prázdna, STOP. Inak GOTO krok 3.*

♣

Kruskalov algoritmus II.

Hrany grafu G usporiadame neklesajúco podľa ich ohodnotenia do nasledujúcej tabuľky:

{2,6}	{2,4}	{3,5}	{1,3}	{1,5}	{4,6}	{2,3}	{5,7}	{4,7}	{5,6}	{2,5}	{3,4}
10	20	20	30	30	30	40	60	70	70	80	80

Hrana do kostry	1	2	3	4	5	6	7
				$k(v)$			
-	1	2	3	4	5	6	7
{2,6}	1	2	3	4	5	2	7
{2,4}	1	2	3	2	5	2	7
{3,5}	1	2	3	2	3	2	7
{1,3}	1	2	1	2	1	2	7
{2,3}	1	1	1	1	1	1	7
{5,7}	1	1	1	1	1	1	1

Kruskalov algoritmus II.

Hrany grafu G usporiadame neklesajúco podľa ich ohodnotenia do nasledujúcej tabuľky:

{2,6}	{2,4}	{3,5}	{1,3}	{1,5}	{4,6}	{2,3}	{5,7}	{4,7}	{5,6}	{2,5}	{3,4}
10	20	20	30	30	30	40	60	70	70	80	80

Hrana $\{u, v\} = \{2, 6\}$
$k(2) = 2,\ k(6) = 6$

$k(2) \neq k(6) \Rightarrow$
zaraď $\{2, 6\}$ do kostry

Hrana do kostry	1	2	3	4	5	6	7
	$k(v)$						
-	1	2	3	4	5	6	7
{2,6}	1	2	3	4	5	2	7
{2,4}	1	2	3	2	5	2	7
{3,5}	1	2	3	2	3	2	7
{1,3}	1	2	1	2	1	2	7
{2,3}	1	1	1	1	1	1	7
{5,7}	1	1	1	1	1	1	1

Kruskalov algoritmus II.

Hrany grafu G usporiadame neklesajúco podľa ich ohodnotenia do nasledujúcej tabuľky:

{2,6}	{2,4}	{3,5}	{1,3}	{1,5}	{4,6}	{2,3}	{5,7}	{4,7}	{5,6}	{2,5}	{3,4}
10	20	20	30	30	30	40	60	70	70	80	80

Hrana $\{u, v\} = \{2, 4\}$
$k(2) = 2,\ k(4) = 4$

$k(2) \neq k(4) \Rightarrow$
zaraď $\{2, 4\}$ do kostry

Hrana do kostry	1	2	3	4	5	6	7
				$k(v)$			
-	1	2	3	4	5	6	7
{2,6}	1	2	3	4	5	2	7
{2,4}	1	2	3	2	5	2	7
{3,5}	1	2	3	2	3	2	7
{1,3}	1	2	1	2	1	2	7
{2,3}	1	1	1	1	1	1	7
{5,7}	1	1	1	1	1	1	1

Kruskalov algoritmus II.

Hrany grafu G usporiadame neklesajúco podľa ich ohodnotenia do nasledujúcej tabuľky:

{2,6}	{2,4}	{3,5}	{1,3}	{1,5}	{4,6}	{2,3}	{5,7}	{4,7}	{5,6}	{2,5}	{3,4}
10	20	20	30	30	30	40	60	70	70	80	80

Hrana $\{u, v\} = \{3, 5\}$
$k(3) = 3,\ k(5) = 5$

$k(3) \neq k(5) \Rightarrow$
zaraď $\{3, 5\}$ do kostry

Hrana do kostry	1	2	3	4	5	6	7
	$k(v)$						
-	1	2	3	4	5	6	7
{2,6}	1	2	3	4	5	2	7
{2,4}	1	2	3	2	5	2	7
{3,5}	1	2	3	2	3	2	7
{1,3}	1	2	1	2	1	2	7
{2,3}	1	1	1	1	1	1	7
{5,7}	1	1	1	1	1	1	1

Kruskalov algoritmus II.

Hrany grafu G usporiadame neklesajúco podľa ich ohodnotenia do nasledujúcej tabuľky:

{2,6}	{2,4}	{3,5}	{1,3}	{1,5}	{4,6}	{2,3}	{5,7}	{4,7}	{5,6}	{2,5}	{3,4}
10	20	20	30	30	30	40	60	70	70	80	80

Hrana $\{u, v\} = \{1, 3\}$
$k(1) = 1,\ k(3) = 3$

$k(1) \neq k(3) \Rightarrow$
zaraď $\{1, 3\}$ do kostry

Hrana do kostry	1	2	3	4	5	6	7
	$k(v)$						
-	1	2	3	4	5	6	7
{2,6}	1	2	3	4	5	2	7
{2,4}	1	2	3	2	5	2	7
{3,5}	1	2	3	2	3	2	7
{1,3}	1	2	1	2	1	2	7
{2,3}	1	1	1	1	1	1	7
{5,7}	1	1	1	1	1	1	1

Kruskalov algoritmus II.

Hrany grafu G usporiadame neklesajúco podľa ich ohodnotenia do nasledujúcej tabuľky:

{2,6}	{2,4}	{3,5}	{1,3}	{1,5}	{4,6}	{2,3}	{5,7}	{4,7}	{5,6}	{2,5}	{3,4}
10	20	20	30	30	30	40	60	70	70	80	80

Hrana $\{u, v\} = \{1, 5\}$
$k(1) = 1,\ k(5) = 1$

$k(1) = k(5) \Rightarrow$
vyhoď $\{1, 5\}$

Hrana do kostry	1	2	3	4	5	6	7
	$k(v)$						
-	1	2	3	4	5	6	7
{2,6}	1	2	3	4	5	2	7
{2,4}	1	2	3	2	5	2	7
{3,5}	1	2	3	2	3	2	7
{1,3}	1	2	1	2	1	2	7
{2,3}	1	1	1	1	1	1	7
{5,7}	1	1	1	1	1	1	1

Kruskalov algoritmus II.

Hrany grafu G usporiadame neklesajúco podľa ich ohodnotenia do nasledujúcej tabuľky:

{2,6}	{2,4}	{3,5}	{1,3}	{1,5}	{4,6}	{2,3}	{5,7}	{4,7}	{5,6}	{2,5}	{3,4}
10	20	20	30	30	30	40	60	70	70	80	80

Hrana $\{u, v\} = \{4, 6\}$
$k(4) = 2,\ k(6) = 2$

$k(4) = k(6) \Rightarrow$
vyhoď $\{4, 6\}$

Hrana do kostry	1	2	3	4	5	6	7
				$k(v)$			
-	1	2	3	4	5	6	7
{2,6}	1	2	3	4	5	2	7
{2,4}	1	2	3	2	5	2	7
{3,5}	1	2	3	2	3	2	7
{1,3}	1	2	1	2	1	2	7
{2,3}	1	1	1	1	1	1	7
{5,7}	1	1	1	1	1	1	1

Kruskalov algoritmus II.

Hrany grafu G usporiadame neklesajúco podľa ich ohodnotenia do nasledujúcej tabuľky:

{2,6}	{2,4}	{3,5}	{1,3}	{1,5}	{4,6}	{2,3}	{5,7}	{4,7}	{5,6}	{2,5}	{3,4}
10	20	20	30	30	30	40	60	70	70	80	80

Hrana $\{u, v\} = \{2, 3\}$
$k(2) = 2,\ k(3) = 1$

$k(2) \neq k(3) \Rightarrow$
zaraď $\{2, 3\}$ do kostry

Hrana do kostry	1	2	3	4	5	6	7
				$k(v)$			
-	1	2	3	4	5	6	7
{2,6}	1	2	3	4	5	2	7
{2,4}	1	2	3	2	5	2	7
{3,5}	1	2	3	2	3	2	7
{1,3}	1	2	1	2	1	2	7
{2,3}	1	1	1	1	1	1	7
{5,7}	1	1	1	1	1	1	1

Kruskalov algoritmus II.

Hrany grafu G usporiadame neklesajúco podľa ich ohodnotenia do nasledujúcej tabuľky:

{2,6}	{2,4}	{3,5}	{1,3}	{1,5}	{4,6}	{2,3}	{5,7}	{4,7}	{5,6}	{2,5}	{3,4}
10	20	20	30	30	30	40	60	70	70	80	80

Hrana $\{u, v\} = \{5, 7\}$
$k(5) = 1,\ k(7) = 7$

$k(5) \neq k(7) \Rightarrow$
zaraď $\{5, 7\}$ do kostry

Hrana do kostry	1	2	3	4	5	6	7
				$k(v)$			
-	1	2	3	4	5	6	7
{2,6}	1	2	3	4	5	2	7
{2,4}	1	2	3	2	5	2	7
{3,5}	1	2	3	2	3	2	7
{1,3}	1	2	1	2	1	2	7
{2,3}	1	1	1	1	1	1	7
{5,7}	1	1	1	1	1	1	1

Cesta maximálnej priepustnosti

Definícia

Nech $G = (V, H, c)$ je hranovo ohodnotený graf, v ktorom cena hrany $h \in H$ $c(h) > 0$ znamená jej priepustnosť.

Priepustnosť $c(\boldsymbol{\mu}(u, v))$ u–v **cesty (sledu, polosledu, atď.)** $\boldsymbol{\mu}(u, v)$ *definujeme ako*

$$c(\boldsymbol{\mu}(u, v)) = \min\{c(h) \mid h \in \boldsymbol{\mu}(u, v)\}.$$

Definícia

Hovoríme, že u–v cesta $\boldsymbol{\mu}(u, v)$ v grafe $G = (V, H, c)$ je u–v **cesta maximálnej priepustnosti**, *má najväčšiu priepustnosť zo všetkých u–v ciest v G.*

Cesta maximálnej priepustnosti

Definícia

Nech $G = (V, H, c)$ je hranovo ohodnotený graf, v ktorom cena hrany $h \in H$ $c(h) > 0$ znamená jej priepustnosť.

Priepustnosť $c(\boldsymbol{\mu}(u, v))$ u–v **cesty (sledu, polosledu, atď.)** $\boldsymbol{\mu}(u, v)$ *definujeme ako*

$$c(\boldsymbol{\mu}(u, v)) = \min\{c(h) \mid h \in \boldsymbol{\mu}(u, v)\}.$$

Definícia

Hovoríme, že u–v cesta $\boldsymbol{\mu}(u, v)$ v grafe $G = (V, H, c)$ je u–v **cesta maximálnej priepustnosti**, *má najväčšiu priepustnosť zo všetkých u–v ciest v G.*

Cesta maximálnej priepustnosti

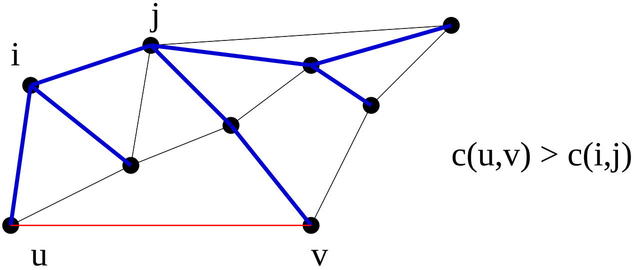
Veta

Nech K je najdrahšia kostra v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$, nech $\{u, v\} \in H$ je taká hrana grafu G, ktorá nepatrí k hranovej množine kostry K.
Nech $\boldsymbol{\mu}(u, v)$ je (jediná) u-v cesta v kostre K.
Potom je priepustnosť cesty $\boldsymbol{\mu}(u, v)$ väčšia alebo rovná ako priepustnosť hrany $\{u, v\}$, t. j.

$$c(\boldsymbol{\mu}(u, v)) \geq c(u, v).$$

Cesta maximálnej priepustnosti

DÔKAZ.
Majme najdrahšiu kostru $\mathcal{K}$ a nech existuje hrana $\{u, v\}$ taká ,že priepustnosť u-v cesty po hranách kostry je menšia než $c(u, v)$).



Kostra $\mathcal{K}$ modro, hrana $h = \{u, v\}$ (červeno)

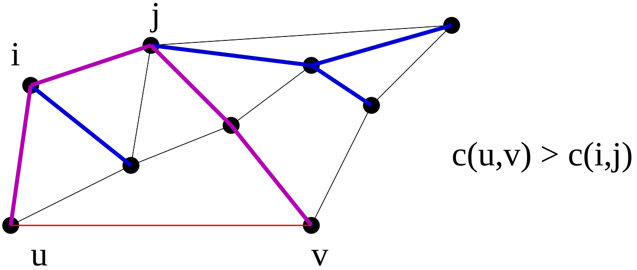
u-v cesta po hranách kostry (fialovo) s menšou priepustnosťou než $c(u, v)$

Musí existovať hrana $\{i, j\}$ tejto cesty taká že $c(u, v) > c(i, j)$

Nahradením hrany $\{i, j\}$ hranou $\{u, v\}$ vznikne kostra s väčšou cenou – spor s tým, že $\mathcal{K}$ bola najdrahšia kostra.

Cesta maximálnej priepustnosti

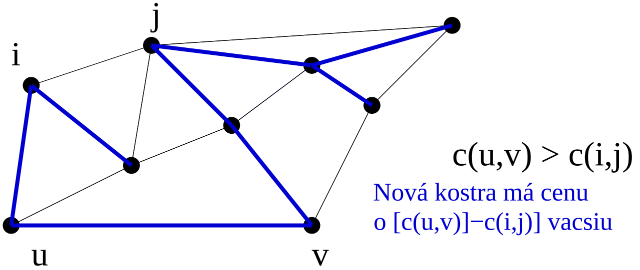
DÔKAZ.
Majme najdrahšiu kostru $\mathcal{K}$ a nech existuje hrana $\{u, v\}$ taká ,že priepustnosť u-v cesty po hranách kostry je menšia než $c(u, v)$).



Kostra $\mathcal{K}$ modro, hrana $h = \{u, v\}$ (červeno)
u-v cesta po hranách kostry (fialovo) s menšou priepustnosťou než $c(u, v)$
Musí existovať hrana $\{i, j\}$ tejto cesty taká že $c(u, v) > c(i, j)$
Nahradením hrany $\{i, j\}$ hranou $\{u, v\}$ vznikne kostra s väčšou cenou – spor s tým, že $\mathcal{K}$ bola najdrahšia kostra.

DÔKAZ.

Majme najdrahšiu kostru $\mathcal{K}$ a nech existuje hrana $\{u, v\}$ taká ,že priepustnosť u-v cesty po hranách kostry je menšia než $c(u, v)$).



Kostra $\mathcal{K}$ modro, hrana $h = \{u, v\}$ (červeno)

u-v cesta po hranách kostry (fialovo) s menšou priepustnosťou než $c(u, v)$

Musí existovať hrana $\{i, j\}$ tejto cesty taká že $c(u, v) > c(i, j)$

Nahradením hrany $\{i, j\}$ hranou $\{u, v\}$ vznikne kostra s väčšou cenou – spor s tým, že $\mathcal{K}$ bola najdrahšia kostra.

Veta

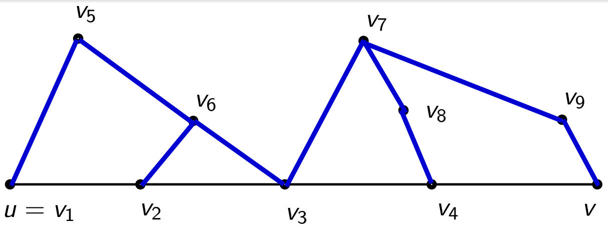
Nech K je najdrahšia kostra v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$. Potom pre ľubovoľné dva vrcholy $u, v \in V$ je (jediná) u–v cesta v K u–v cestou maximálnej priepustnosti v G.

Dôkaz.

Veta

Nech K je najdrahšia kostra v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$. Potom pre ľubovoľné dva vrcholy $u, v \in V$ je (jediná) u–v cesta v K u–v cestou maximálnej priepustnosti v G.

DÔKAZ.



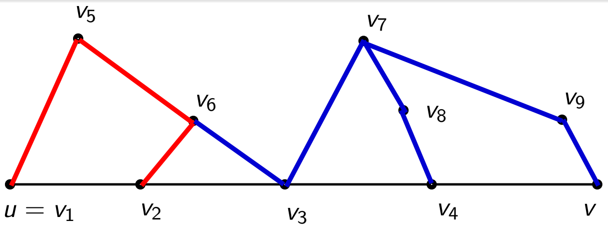
Cesta max. priepustnosti:

$$\mu(u, v) = (u, \{u \equiv v_1, v_2\}, v_2, \{v_2, v_3\}, v_3, \{v_3, v_4\}, v_4, \{v_4, v\}, v),$$

Veta

Nech K je najdrahšia kostra v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$. Potom pre ľubovoľné dva vrcholy $u, v \in V$ je (jediná) u–v cesta v K u–v cestou maximálnej priepustnosti v G.

DÔKAZ.

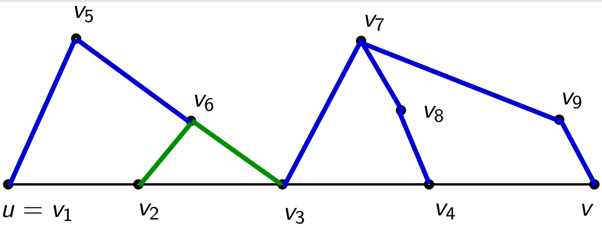


$$\mu(u, v_2) = (u, \{u, v_5\}, v_5, \{v_5, v_6\}, v_6, \{v_6, v_2\}, v_2),$$

Veta

Nech K je najdrahšia kostra v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$. Potom pre ľubovoľné dva vrcholy $u, v \in V$ je (jediná) u–v cesta v K u–v cestou maximálnej priepustnosti v G.

Dôkaz.

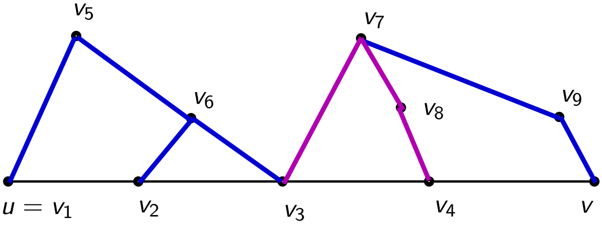


$$\mu(v_2, v_3) = (v_2, \{v_2, v_6\}, v_6, \{v_6, v_3\}, v_3),$$

Veta

Nech K je najdrahšia kostra v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$. Potom pre ľubovoľné dva vrcholy $u, v \in V$ je (jediná) u–v cesta v K u–v cestou maximálnej priepustnosti v G.

Dôkaz.



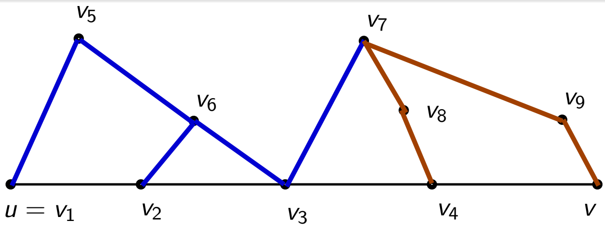
$$\mu(v_3, v_4) = (v_3, \{v_3, v_7\}, v_7, \{v_7, v_8\}, v_8, \{v_8, v_4\}, v_4),$$

Cesta maximálnej priepustnosti

Veta

Nech K je najdrahšia kostra v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$. Potom pre ľubovoľné dva vrcholy $u, v \in V$ je (jediná) u–v cesta v K u–v cestou maximálnej priepustnosti v G.

DÔKAZ.

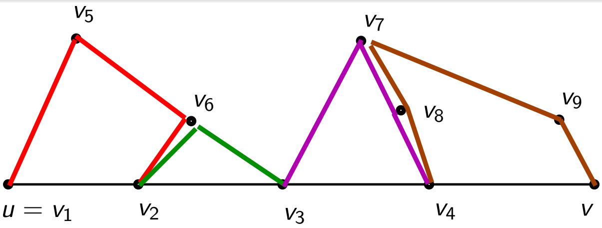


$$\mu(v_4, v) = (v_4, \{v_4, v_8\}, v_8, \{v_8, v_7\}, v_7, \{v_7, v_9\}, v_9, \{v_9, v\}, v).$$

Veta

Nech K je najdrahšia kostra v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$. Potom pre ľubovoľné dva vrcholy $u, v \in V$ je (jediná) u–v cesta v K u–v cestou maximálnej priepustnosti v G.

DÔKAZ.

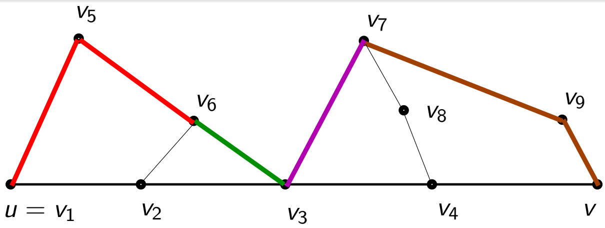


u-v sled po hranách kostry s priepustnosťou $\geq$ než priepustnnosť cesty $\mu(u, v)$

Veta

Nech K je najdrahšia kostra v súvislom hranovo ohodnotenom grafe $G = (V, H, c)$. Potom pre ľubovoľné dva vrcholy $u, v \in V$ je (jediná) u–v cesta v K u–v cestou maximálnej priepustnosti v G.

DÔKAZ.



Cesta max. priepustnosti:

$$\mu(u, v) = (u, \{u \equiv v_1, v_2\}, v_2, \{v_2, v_3\}, v_3, \{v_3, v_4\}, v_4, \{v_4, v\}, v),$$

Cesta max. priepustnosti po hranách kostry

$$u, \{u, v_5\}, v_5, \{v_5, v_6\}, v_6, \{v_6, v_3\}, v_3, \{v_3, v_7\}, v_7, \{v_7, v_9\}, v_9, \{v_9, v\}, v.$$

Algoritmus

Algoritmus na hľadanie u–v **cesty maximálnej priepustnosti v súvislom hranovo ohodnotenom grafe** $G = (V, H, c)$**.**

- **Krok 1.** *V grafe* G *zostroj najdrahšiu kostru* K.
- **Krok 2.** *V kostre* K *nájdi (jedinú)* u–v *cestu.*
  Táto (jediná) u–v *cesta v kostre* K *je* u–v *cestou maximálnej priepustnosti v grafe* G.

♣

Poznámka

Uvedený algoritmus síce nájde u–v cestu maximálnej priepustnosti, no táto nemusí byť – a spravidla ani nebýva – optimálnou z hľadiska prejdenej vzdialenosti.
Ak by sme chceli nájsť najkratšiu u–v cestu s maximálnou priepustnosťou, potrebujeme mať v príslušnom grafe okrem kapacitného ohodnotenia hrán aj ohodnotenie vyjadrujúce ich dĺžku.

Algoritmus

Algoritmus na hľadanie u–v **cesty maximálnej priepustnosti v súvislom hranovo ohodnotenom grafe** $G = (V, H, c)$**.**

- **Krok 1.** *V grafe* G *zostroj najdrahšiu kostru* K*.*
- **Krok 2.** *V kostre* K *nájdi (jedinú)* u–v *cestu.*

  Táto (jediná) u–v *cesta v kostre* K *je* u–v *cestou maximálnej priepustnosti v grafe* G*.* ♣

Poznámka

Uvedený algoritmus síce nájde u–v *cestu maximálnej priepustnosti, no táto nemusí byť – a spravidla ani nebýva – optimálnou z hľadiska prejdenej vzdialenosti.*
Ak by sme chceli nájsť najkratšiu u–v *cestu s maximálnou priepustnosťou, potrebujeme mať v príslušnom grafe okrem kapacitného ohodnotenia hrán aj ohodnotenie vyjadrujúce ich dĺžku.*

Algoritmus na hľadanie u–v cesty maximálnej priepustnosti

Algoritmus

Algoritmus na hľadanie u–v **cesty maximálnej priepustnosti v súvislom hranovo ohodnotenom grafe** $G = (V, H, c)$**.**

- **Krok 1.** *V grafe* G *zostroj najdrahšiu kostru* K.
- **Krok 2.** *V kostre* K *nájdi (jedinú)* u–v *cestu.*

  Táto (jediná) u–v *cesta v kostre* K *je* u–v *cestou maximálnej priepustnosti v grafe* G.

♣

Poznámka

Uvedený algoritmus síce nájde u–v cestu maximálnej priepustnosti, no táto nemusí byť – a spravidla ani nebýva – optimálnou z hľadiska prejdenej vzdialenosti.
Ak by sme chceli nájsť najkratšiu u–v cestu s maximálnou priepustnosťou, potrebujeme mať v príslušnom grafe okrem kapacitného ohodnotenia hrán aj ohodnotenie vyjadrujúce ich dĺžku.

Algoritmus

Algoritmus na hľadanie najkratšej u–v cesty s maximálnou priepustnosťou v súvislom hranovo ohodnotenom grafe $G = (V, H, c, d)$, kde $c(h)$ je priepustnosť a $d(h)$ je dĺžka hrany $h \in H$.

- **Krok 1.** *V grafe G nájdi cestu $\mu(u, v)$ maximálnej priepustnosti vzhľadom na ohodnotenie hrán c.*

  Nech C je priepustnosť cesty $\mu(u, v)$.

- **Krok 2.** *Vytvor graf $G' = (V, H', d)$, kde $H' = \{h | h \in H,\ c(h) \geq C\}$.*
  $\{H'$ obsahuje len tie hrany pôvodného grafu, ktoré majú priepustnosť väčšiu alebo rovnú než $C.\}$

- **Krok 3.** *V grafe G' nájdi najkratšiu u–v cestu vzhľadom na ohodnotenie hrán d.*

♣

Algoritmus

Algoritmus na hľadanie najkratšej u–v cesty s maximálnou priepustnosťou v súvislom hranovo ohodnotenom grafe $G = (V, H, c, d)$, kde $c(h)$ je priepustnosť a $d(h)$ je dĺžka hrany $h \in H$.

- **Krok 1.** *V grafe G nájdi cestu $\mu(u, v)$ maximálnej priepustnosti vzhľadom na ohodnotenie hrán c.*

  Nech C je priepustnosť cesty $\mu(u, v)$.

- **Krok 2.** *Vytvor graf $G' = (V, H', d)$, kde $H' = \{h | h \in H,\ c(h) \geq C\}$.*
  $\{H'$ obsahuje len tie hrany pôvodného grafu, ktoré majú priepustnosť väčšiu alebo rovnú než C.$\}$

- **Krok 3.** *V grafe G' nájdi najkratšiu u–v cestu vzhľadom na ohodnotenie hrán d.*

Algoritmus

Algoritmus na hľadanie najkratšej u–v cesty s maximálnou priepustnosťou v súvislom hranovo ohodnotenom grafe $G = (V, H, c, d)$, kde $c(h)$ je priepustnosť a $d(h)$ je dĺžka hrany $h \in H$.

- **Krok 1.** *V grafe G nájdi cestu $\boldsymbol{\mu}(u, v)$ maximálnej priepustnosti vzhľadom na ohodnotenie hrán c.*

  Nech C je priepustnosť cesty $\boldsymbol{\mu}(u, v)$.

- **Krok 2.** *Vytvor graf $G' = (V, H', d)$, kde $H' = \{h | h \in H,\ c(h) \geq C\}$.*
  $\{H'$ obsahuje len tie hrany pôvodného grafu, ktoré majú priepustnosť väčšiu alebo rovnú než C.$\}$

- **Krok 3.** *V grafe G' nájdi najkratšiu u–v cestu vzhľadom na ohodnotenie hrán d.*

♣