

GAaDO

Kostra grafu

Technická univerzita v Košiciach



2025–2026

Definícia

- *Kostra*

Definícia

- **Kostra** súvislého grafu $G = (V, H)$ je taký jeho faktorový podgraf, ktorý je strom.
- Nech $G = (V, H, c)$ je hranovo ohodnotený graf a K kostra grafu G . **Cena kostry** K , ozn. $c(K)$,

Definícia

- **Kostra** súvislého grafu $G = (V, H)$ je taký jeho faktorový podgraf, ktorý je strom.
- Nech $G = (V, H, c)$ je hranovo ohodnotený graf a K kostra grafu G . **Cena kostry** K , ozn. $c(K)$, je súčet ohodnotení jej hrán.
- **Najlacnejšia kostra** v grafe G

Definícia

- **Kostra** súvislého grafu $G = (V, H)$ je taký jeho faktorový podgraf, ktorý je strom.
- Nech $G = (V, H, c)$ je hranovo ohodnotený graf a K kostra grafu G . **Cena kostry** K , ozn. $c(K)$, je súčet ohodnotení jej hrán.
- **Najlacnejšia kostra** v grafe G je kostra s najmenšou cenou.
- **Najdrahšia kostra** (najcennejšia kostra) v grafe G

Definícia

- **Kostra** súvislého grafu $G = (V, H)$ je taký jeho faktorový podgraf, ktorý je strom.
- Nech $G = (V, H, c)$ je hranovo ohodnotený graf a K kostra grafu G . **Cena kostry** K , ozn. $c(K)$, je súčet ohodnotení jej hrán.
- **Najlacnejšia kostra** v grafe G je kostra s najmenšou cenou.
- **Najdrahšia kostra** (najcennejšia kostra) v grafe G je kostra s najväčšou cenou.

Definition

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

Definition

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

Definition

Dve kostry sa nazývajú **susedné**,

Definition

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

Definition

Dve kostry sa nazývajú **susedné**, ak jednu možno získať z druhej výmenou jednej hrany.

Definition

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

Definition

Dve kostry sa nazývajú **susedné**, ak jednu možno získať z druhej výmenou jednej hrany.

Veta

Kostra K grafu $G = (V, H, c)$ je najlacnejšia práve vtedy, keď k nej neexistuje lacnejšia susedná kostra.

Kruskalov algoritmus

- **Krok 1.** Zorad' hrany podľa ohodnotenia **vzostupne/zostupne** do postupnosti P .

Kruskalov algoritmus

- **Krok 1.** Zorad' hrany podľa ohodnotenia **vzostupne/zostupne** do postupnosti P .
- **Krok 2.** Pre každý vrchol $v_i \in V$ polož $k(i) = i$.

Kruskalov algoritmus

- **Krok 1.** Zorad' hrany podľa ohodnotenia **vzostupne/zostupne** do postupnosti P .
- **Krok 2.** Pre každý vrchol $v_i \in V$ polož $k(i) = i$.
- **Krok 3.** Nech prvá hrana v postupnosti P je hrana uv . Vylúč hranu uv z postupnosti P . Ak $k(u) \neq k(v)$, zarad' hranu uv do kostry a $\forall i \in V$, pre ktoré $k(i) = k(v)$, polož $k(i) := k(u)$.

Kruskalov algoritmus

- **Krok 1.** Zorad' hrany podľa ohodnotenia vzostupne/zostupne do postupnosti P .
- **Krok 2.** Pre každý vrchol $v_i \in V$ polož $k(i) = i$.
- **Krok 3.** Nech prvá hrana v postupnosti P je hrana uv . Vylúč hranu uv z postupnosti P . Ak $k(u) \neq k(v)$, zarad' hranu uv do kostry a $\forall i \in V$, pre ktoré $k(i) = k(v)$, polož $k(i) := k(u)$.
- **Krok 4.** Ak je počet vybraných hrán rovný $|V| - 1$, alebo ak je postupnosť P prázdna, STOP. Inak choď na krok 3.

Kruskalov algoritmus

Zložitosť Kruskalovho algoritmu:

- Krok 1 vyžaduje $O(m \log(m))$ operácií,

Zložitosť Kruskalovho algoritmu:

- Krok 1 vyžaduje $O(m \log(m))$ operácií,
- krok 2 $O(n)$.

Zložitosť Kruskalovho algoritmu:

- Krok 1 vyžaduje $O(m \log(m))$ operácií,
- krok 2 $O(n)$.
- Krok 3 sa opakuje najviac m -krát. Ak sa hrana nezaradí do P , bola s ňou urobená len jedna operácia. Ak hrana bola zaradená do kostry, čo nastane presne $(n - 1)$ -krát, je potrebné preznačkováť značky vrcholov, čo vyžaduje $O(n)$ operácií pri jednom zaradení, t.j. $O(n(n - 1))$ operácií celkom.

Zložitosť Kruskalovho algoritmu:

- Krok 1 vyžaduje $O(m \log(m))$ operácií,
- krok 2 $O(n)$.
- Krok 3 sa opakuje najviac m -krát. Ak sa hrana nezaradí do P , bola s ňou urobená len jedna operácia. Ak hrana bola zaradená do kostry, čo nastane presne $(n - 1)$ -krát, je potrebné preznačovať značky vrcholov, čo vyžaduje $O(n)$ operácií pri jednom zaradení, t.j. $O(n(n - 1))$ operácií celkom.
- celková zložitosť - $O(m \log(m) + n^2)$.

Použitím vhodných dátových štruktúr možno dosiahnuť $O(m \log(n))$.

Primov algoritmus

- **Krok 1.** Vyberieme ľubovoľný vrchol grafu a zarad' ho do kostry K .

Primov algoritmus

- **Krok 1.** Vyberieme ľubovoľný vrchol grafu a zarad' ho do kostry K .
- **Krok 2.** Ku kostre K pridaj hraničnú hranu grafu s najnižšou cenou.

Primov algoritmus

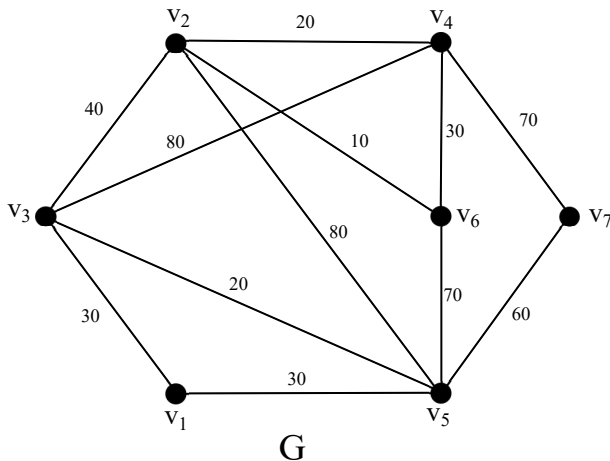
- **Krok 1.** Vyberieme ľubovoľný vrchol grafu a zarad' ho do kostry K .
- **Krok 2.** Ku kostre K pridaj hraničnú hranu grafu s najnižšou cenou.
- **Krok 3.** Ak strom už obsahuje $n - 1$ hrán, STOP , inak pokračuj na Krok 2.

Primov algoritmus

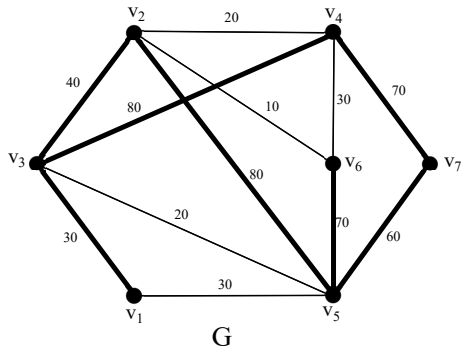
Zložitosť Primovho algoritmu: Podobne ako v predchádzajúcom Kruskalovom algoritme dostávame časovú zložitosť algoritmu rovnú $O(m \log(n))$.

Príklad

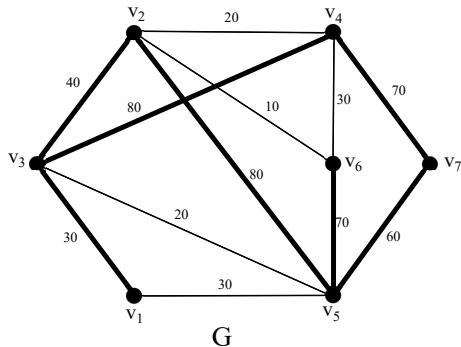
Nájdite maximálnu kostru grafu G na obrázku. Použite Kruskalov aj Primov algoritmus. Napíšte v akom poradí pribúdali hrany do kostry



Príklad

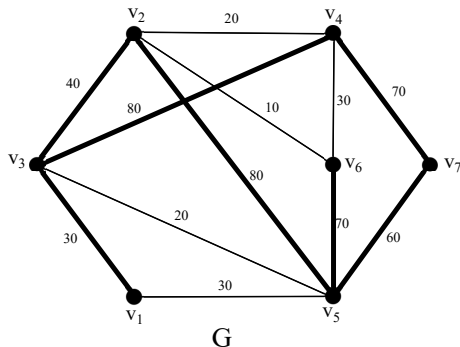


Príklad



Kruskalov algoritmus: $v_2 v_5$, $v_3 v_4$, $v_4 v_7$, $v_5 v_6$, $v_5 v_7$, $v_1 v_3$

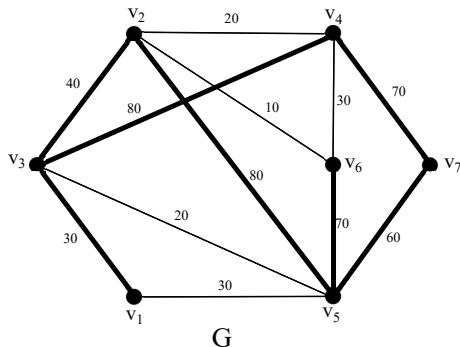
Príklad



Kruskalov algoritmus: $v_2 v_5$, $v_3 v_4$, $v_4 v_7$, $v_5 v_6$, $v_5 v_7$, $v_1 v_3$

Primov algoritmus: $v_1 v_3$, $v_3 v_4$, $v_4 v_7$, $v_7 v_5$, $v_5 v_2$, $v_5 v_6$

Príklad



Kruskalov algoritmus: $v_2 v_5$, $v_3 v_4$, $v_4 v_7$, $v_5 v_6$, $v_5 v_7$, $v_1 v_3$

Primov algoritmus: $v_1 v_3$, $v_3 v_4$, $v_4 v_7$, $v_7 v_5$, $v_5 v_2$, $v_5 v_6$

Cena kostry: 400

Cesta maximálnej priepustnosti

Budeme sa zaoberať hranovo ohodnoteným grafmi

$G = (V, H, c)$, v ktorých $c(h) > 0$, $h \in H$ znamená priepustnosť (kapacitu) hrany h .

Priepustnosť $c(h)$ hrany h môže znamenať:

- maximálny počet vozidiel, ktoré prejdú hranou za jednotku času,
- maximálnu hmotnosť (výšku, šírku) vozidla, ktoré prejde hranou,
- maximálny počet bitov za sekundu prenesiteľných úsekom počítačovej siete,
- veľkosť maximálneho prúdu, ktorý môže tiecť segmentom elektrickej siete.

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** $c(P)$, uv -cesty (sledu, ťahu, dráhy), definujeme ako $c(P) = \min\{c(h) : h \in H(P)\}$.*

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) uv -cesta v K uv -cestou maximálnej priepustnosti v G .

Algoritmus na nájdenie najkratšej cesty s maximálnou priepustnosťou:

- **Krok 1.** V grafe G zostroj najdrahšiu kostru K vzhľadom na ohodnotenie hrán c .

Algoritmus na nájdenie najkratšej cesty s maximálnou priepustnosťou:

- **Krok 1.** V grafe G zostroj najdrahšiu kostru K vzhľadom na ohodnotenie hrán c .
- **Krok 2.** V kostre K nájdi (jedinú) uv -cestu. Táto cesta P v kostre K je uv -cestou maximálnej priepustnosti v grafe G . Nech C je priepustnosť cesty P .

Algoritmus na nájdenie najkratšej cesty s maximálnou priepustnosťou:

- **Krok 1.** V grafe G zostroj najdrahšiu kostru K vzhľadom na ohodnotenie hrán c .
- **Krok 2.** V kostre K nájdí (jedinú) uv -cestu. Táto cesta P v kostre K je uv -cestou maximálnej priepustnosti v grafe G . Nech C je priepustnosť cesty P .
- **Krok 3.** Vytvor graf $G' = (V, H', d)$, kde $H' = \{h : h \in H, c(h) \geq C\}$.

Algoritmus na nájdenie najkratšej cesty s maximálnou priepustnosťou:

- **Krok 1.** V grafe G zostroj najdrahšiu kostru K vzhľadom na ohodnotenie hrán c .
- **Krok 2.** V kostre K nájdí (jedinú) uv -cestu. Táto cesta P v kostre K je uv -cestou maximálnej priepustnosti v grafe G . Nech C je priepustnosť cesty P .
- **Krok 3.** Vytvor graf $G' = (V, H', d)$, kde $H' = \{h : h \in H, c(h) \geq C\}$.
- **Krok 4.** V grafe G' nájdí najkratšiu uv -cestu vzhľadom na ohodnotenie hrán d .

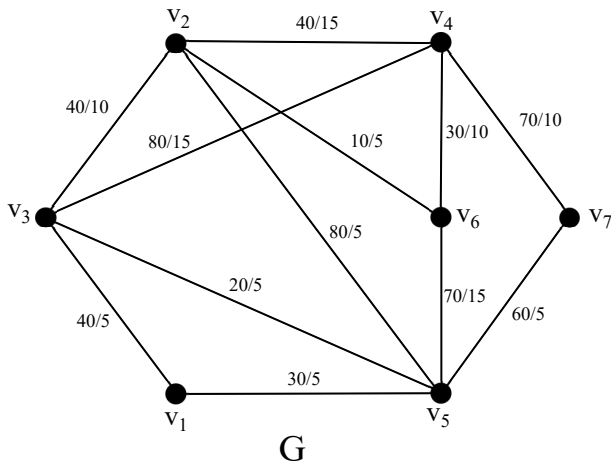
Cesta maximálnej priepustnosti

Zložitosť algoritmu:

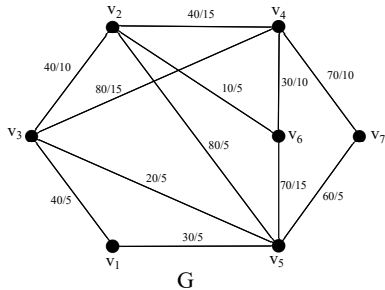
Zložitosť algoritmu priamo úmerne závisí od zložitosti algoritmu na hľadanie kostier a od zložitosti algoritmu na nájdenie najkratšej cesty. Vzhľadom na to, že v tomto prípade nie je dôležité poznať vzdialenosť z ľubovoľných vrcholov, môžeme povedať, že zložitosť algoritmu je rovná $O(m \log(n))$.

Príklad

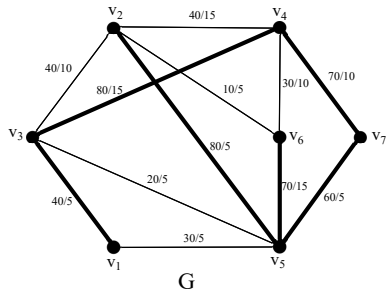
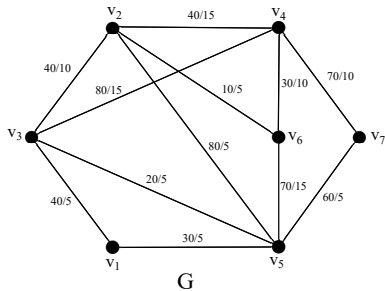
Nájdite najkratšiu cestu maximálnej priepustnosti medzi vrcholmi v_7 a v_1 v grafe G .



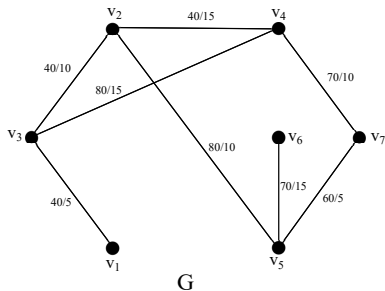
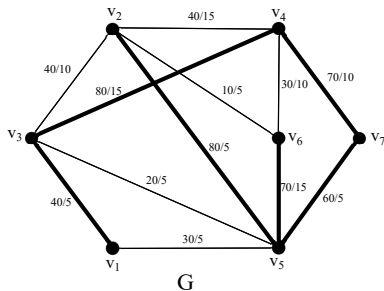
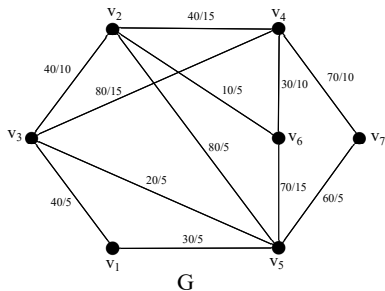
Príklad - riešenie



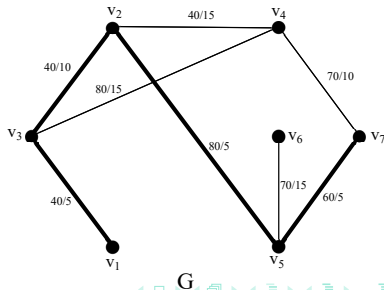
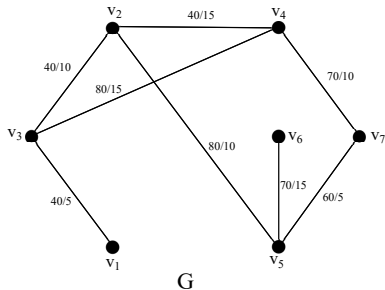
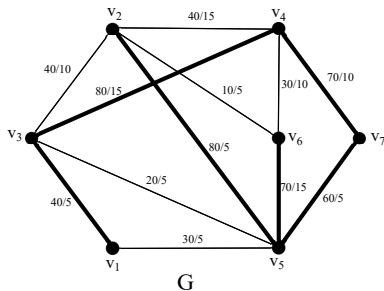
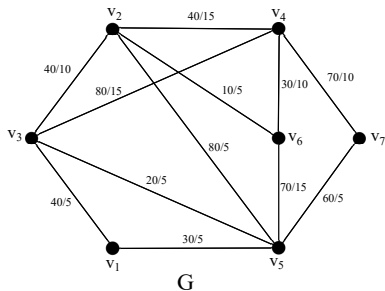
Príklad - riešenie



Príklad - riešenie



Príklad - riešenie



- Prefixové kódovanie.

- Prefixové kódovanie.
- Štruktúra adresárov na disku.

- Prefixové kódovanie.
- Štruktúra adresárov na disku.
- Úloha o elektrifikácii.

- Prefixové kódovanie.
- Štruktúra adresárov na disku.
- Úloha o elektrifikácii.
- Plánovanie prepravy nadrozmerných nákladov.

- Prefixové kódovanie.
- Štruktúra adresárov na disku.
- Úloha o elektrifikácii.
- Plánovanie prepravy nadrozmerných nákladov.