

Borůvkův algoritmus

Richard Klem

17. dubna 2023

- Definice problému
- Trocha z historie
- Pseudokód
- Složitost algoritmu
- Příklad

Definice problému

Definice pojmů

Kostra grafu

Nechť $G = (V_G, E_G)$ je graf definovaný množinou vrcholů V_G a množinou hran E_G .

Pak podgraf $F = (V_F, E_F)$, takový, že $V_F = V_G$, $E_F \subseteq E_G$ a F je stromem, je kostrou grafu G .

Minimální kostra

Minimální kostra F grafu $G = (V_G, E_G)$, je taková kostra, kde hrany $e \in E_G$ jsou ohodnoceny nějakým číslem a součet hodnotí hran kostry

$$\sum_{e \in E_F} w(e)$$

, kde $w(e)$ je hodnota hrany, je minimální.

Definice problému

Borůvkův algoritmus

Borůvkův algoritmus řeší problém hledání nejmenší kostry grafu. Algoritmus pracuje s grafy, které mají ohodnocení hran pomocí nezáporných čísel.

BORŮVKŮV ALGORITMUS

Input: Graf $G = (V_G, E_G)$ s různě a nezáporně ohodnocenými hranami.

$C_{old} = V_G$, $C_{new} = \{\}$, $E_F = \{\}$.

Output: Minimální kostra $F = (V_G, E_F)$.

```
1 while  $|C_{old}| > 1$  do
2     foreach komponenta  $v$   $C_{old}$  do
3         Najdi hranu vedoucí z komponenty s nejmenším ohodnocením.
4         Zjisti do které komponenty hrana vede.
5         Z těchto dvou komponent vytvoř jednu novou.
6         if Nová komponenta ještě v  $C_{new}$  není then
7             Do  $C_{new}$  vlož tuto nově vytvořenou komponentu.
8             Do  $E_F$  vlož vybranou hranu.
9         end if
10    end foreach
11     $C_{old} = C_{new}$ 
12 end while
```

Asymptotická složitost

Asymptotická časová složitost algoritmu vyjadřuje vztah mezi velikostí vstupních dat a potřebným časem pro zpracování tohoto vstupu.

Asymptotická složitost

Asymptotická časová složitost algoritmu vyjadřuje vztah mezi velikostí vstupních dat a potřebným časem pro zpracování tohoto vstupu.

- Borůvkův algoritmus s každou iterací sníží počet komponent minimálně na polovinu.

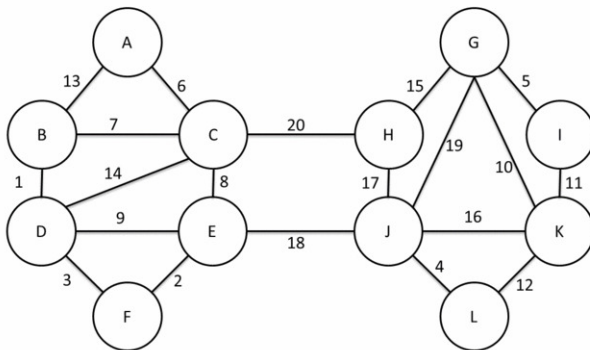
Asymptotická složitost

Asymptotická časová složitost algoritmu vyjadřuje vztah mezi velikostí vstupních dat a potřebným časem pro zpracování tohoto vstupu.

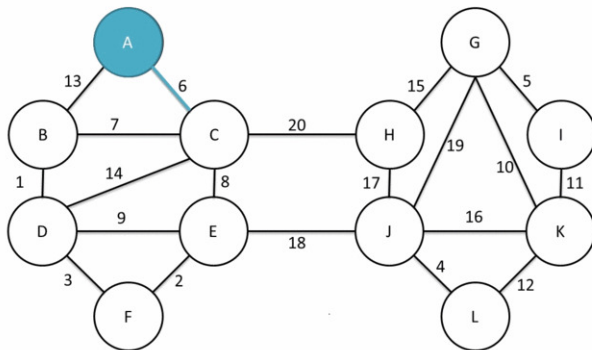
- Borůvkův algoritmus s každou iterací sníží počet komponent minimálně na polovinu.
- Algoritmus má tedy logaritmickou složitost

$$\mathcal{O}(\log(|V|))$$

, kde $|V|$ je počet vrcholů grafu.

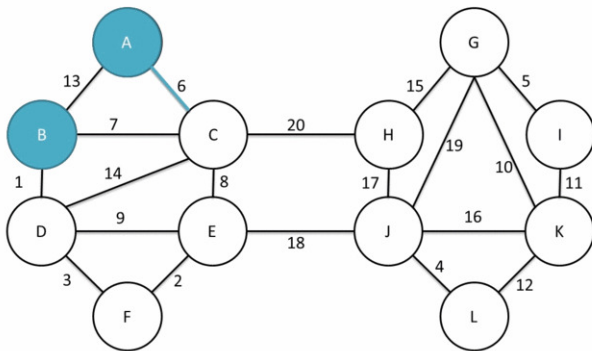


Obrázek: Průběh hledání minimální kostry grafu.[1]

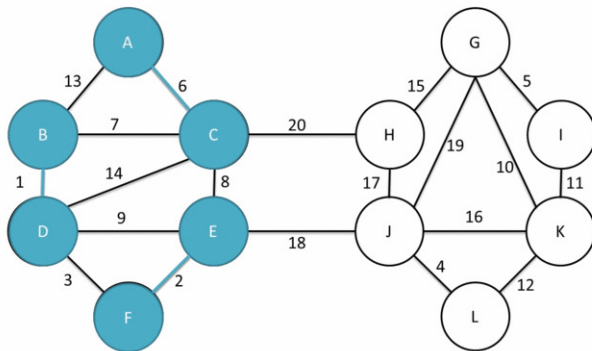


Obrázek: Průběh hledání minimální kostry grafu.[1]

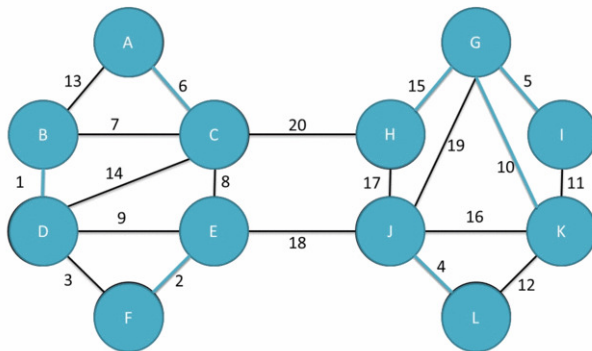
Příklad



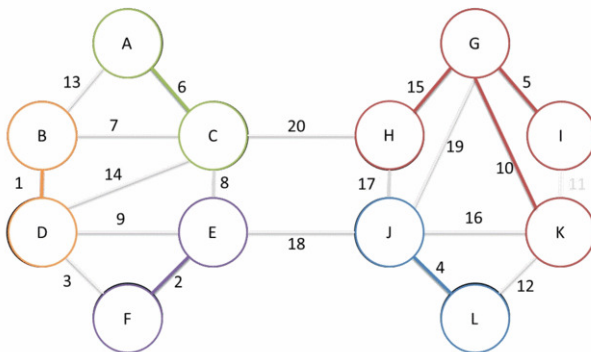
Obrázek: Průběh hledání minimální kostry grafu.[1]



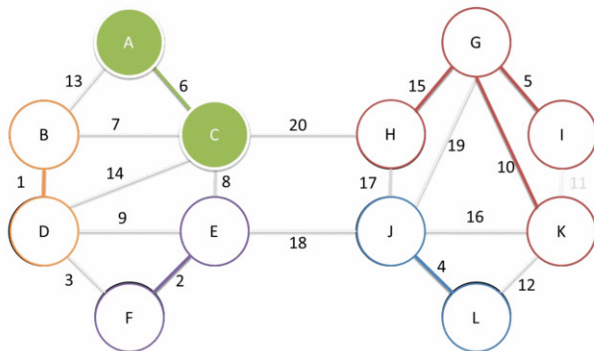
Obrázek: Průběh hledání minimální kostry grafu.[1]



Obrázek: Průběh hledání minimální kostry grafu.[1]

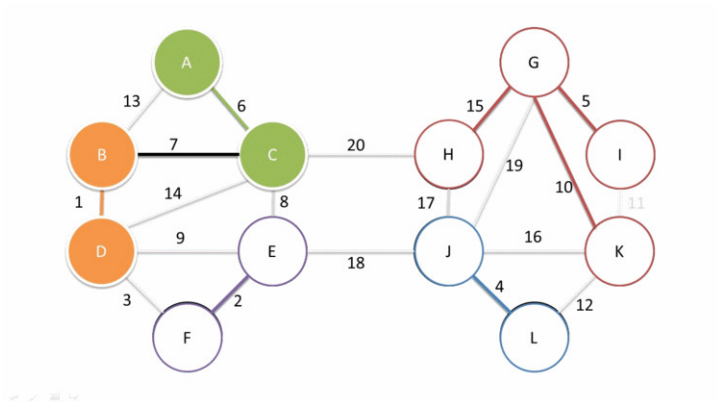


Obrázek: Průběh hledání minimální kostry grafu.[1]



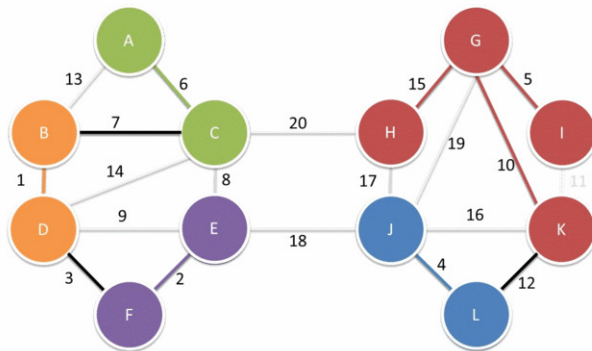
Obrázek: Průběh hledání minimální kostry grafu.[1]

Příklad

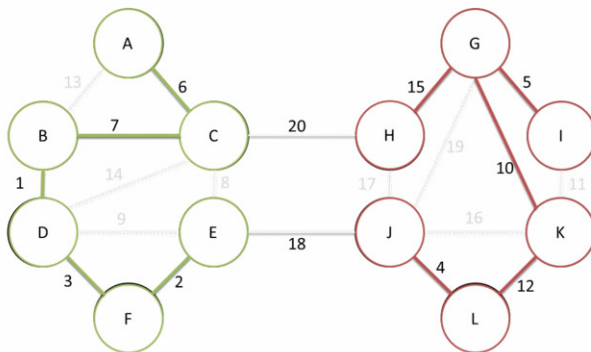


Obrázek: Průběh hledání minimální kostry grafu.[1]

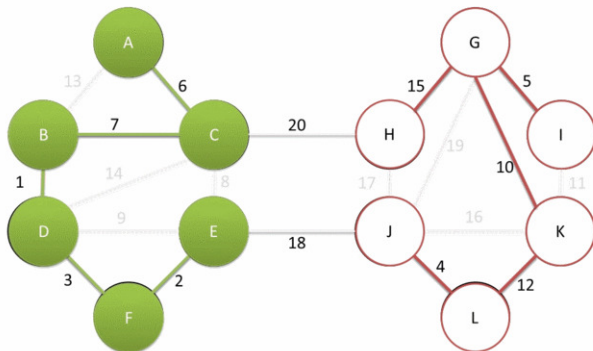
Příklad



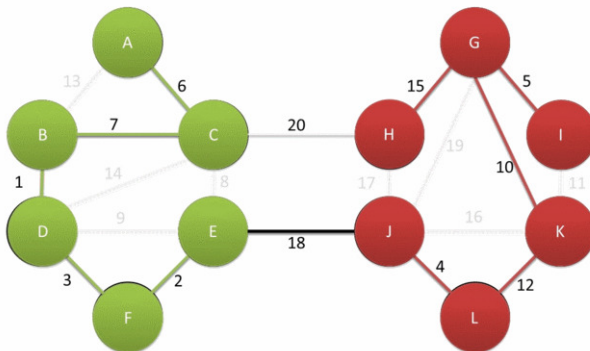
Obrázek: Průběh hledání minimální kostry grafu.[1]



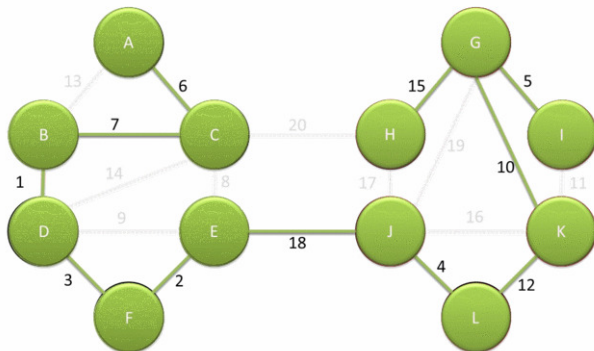
Obrázek: Průběh hledání minimální kostry grafu.[1]



Obrázek: Průběh hledání minimální kostry grafu.[1]



Obrázek: Průběh hledání minimální kostry grafu.[1]



Obrázek: Průběh hledání minimální kostry grafu.[1]



Wikimedia Commons. *An animation, describing Boruvka's (Sollin's) algorithm.* Navštíveno 10.4.2020. 2012. URL:
[https://commons.wikimedia.org/wiki/File:
Boruvka%27s_algorithm_\(Sollin%27s_algorithm\)_Anim.gif.](https://commons.wikimedia.org/wiki/File:Boruvka%27s_algorithm_(Sollin%27s_algorithm)_Anim.gif)