

# Grafové algoritmy

Erik Belko  
([xbelko02@stud.fit.vutbr.cz](mailto:xbelko02@stud.fit.vutbr.cz))

Vysoké učení technické v Brně  
Fakulta informačních technologií

3. května 2020

# Přehled

## 1. Definice grafu

## 2. Primuv algoritmus

Pseudokód a Ukázka  
Složitost

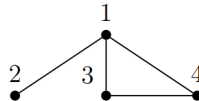
## 3. Použité zdroje

# Graf

**Graf** (jednoduchý neorientovaný) je uspořádaná dvojice  $G = (V, E)$ , kde  $V$  (verticies) je množina **vrcholů** a  $E$  (edges) je množina **hran**, množina vybraných dvouprvkových množin vrcholů.

$$V = \{1, 2, 3, 4\}$$

$$E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 4\}\}$$



# Primův algoritmus

- ▶ **Primův algoritmus** (Jarníkův, Primův-Jarníkův nebo i DJP alg.) je algoritmus hledající minimální kostru souvislého ohodnoceného grafu.



# Primův algoritmus

- ▶ **Primův algoritmus** (Jarníkův, Primův-Jarníkův nebo i DJP alg.) je algoritmus hledající minimální kostru souvislého ohodnoceného grafu.
- ▶ Co je to kostra?
  - ▶ **Kostra grafu**  $G$  je jeho podgraf  $T$ , který je stromem, a propojuje všechny vrcholy, tedy  $V(T) = V(G)$ .



# Primův algoritmus

- ▶ **Primův algoritmus** (Jarníkův, Primův-Jarníkův nebo i DJP alg.) je algoritmus hledající minimální kostru souvislého ohodnoceného grafu.
- ▶ Co je to kostra?
  - ▶ **Kostra grafu**  $G$  je jeho podgraf  $T$ , který je stromem, a propojuje všechny vrcholy, tedy  $V(T) = V(G)$ .
- ▶ Najde tedy takovou podmnožinu hran grafu, která tvoří strom obsahující všechny vrcholy původního grafu a **součet ohodnocení hran** z této množiny je minimální.

# Pseudokód

Algoritmus začíná s jedním vrcholem a postupně přidává další, čímž zvětšuje velikost stromu, dokud neobsahuje všechny vrcholy.

---

## Algoritmus 1: PRIMŮV ALGORITHMUS

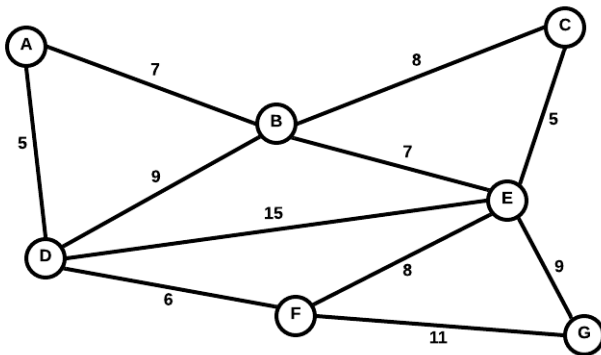
---

**Input:** souvislý ohodnocený graf  $G(V, E)$

**Output:**  $T(V', E')$  je minimální kostra grafu

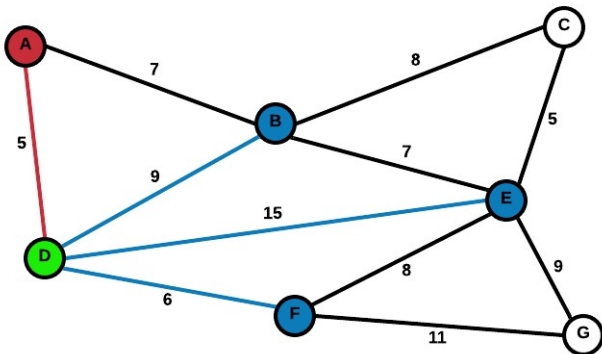
- 1:  $V' = \{x\}$ , kde  $x$  je libovolný vrchol z  $V$ ,  $E' = \{\}$
  - 2: **while**  $V' \neq V$  **do**
  - 3:   Vyber hranu  $(u, v)$  z  $E$  s minimální cenou tak, že  $u$  patří  $V'$   
      a  $v$  nepatří  $V'$
  - 4:   Přidej  $v$  do  $V'$ , přidej  $(u, v)$  do  $E'$
  - 5: **end while**
  - 6: **return**  $T(V', E')$
-

# Ukázka Primůvho algoritmu

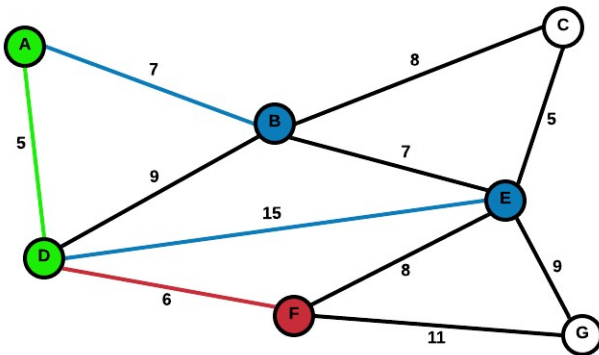




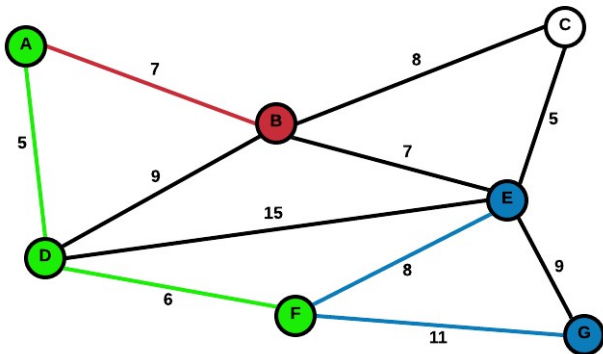
# Ukázka Primůvho algoritmu



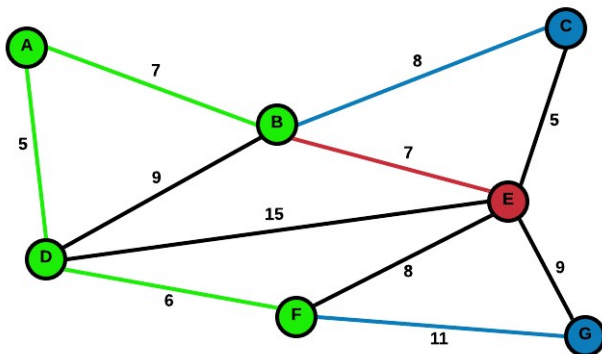
# Ukázka Primůvho algoritmu



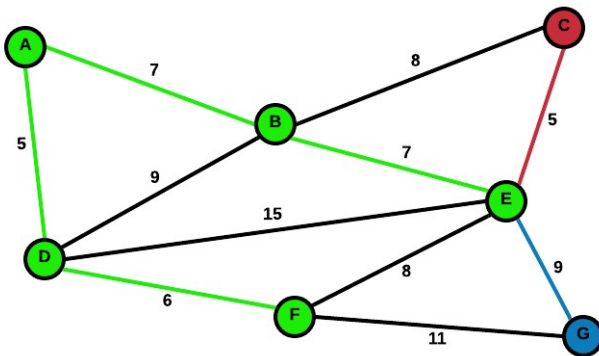
# Ukázka Primůvho algoritmu



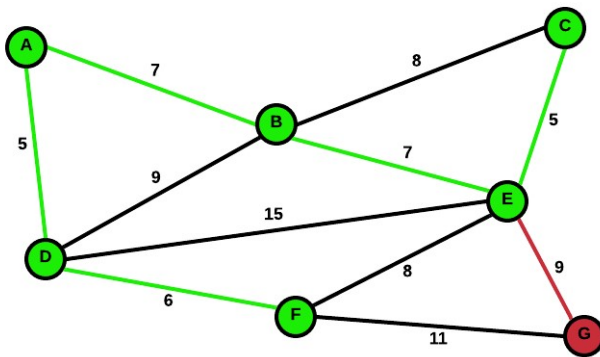
# Ukázka Primůvho algoritmu



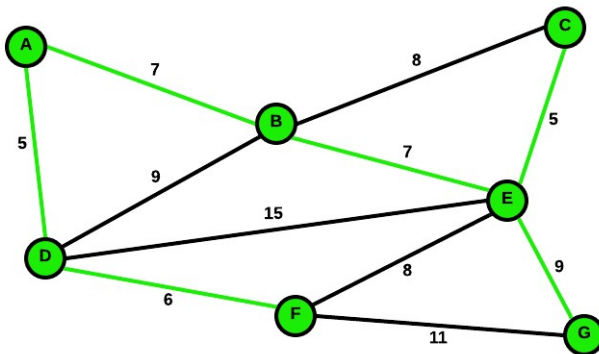
# Ukázka Primůvho algoritmu



# Ukázka Primůvho algoritmu



# Ukázka Primůvho algoritmu



# Složitost

- ▶ Každá hrana má dva konce, proto je testována dvakrát
- ▶ Potřebný čas je maximálně přímo úměrný  $|E| \log_2 |V|$
- ▶  $|V|$  je počet vrcholů
- ▶  $|E|$  je počet hran



# Použité zdroje

## ► Teorie grafů

- <http://www.fit.vutbr.cz/~lengal/idm/grafy-algoritmy.pdf>

- <http://www.fit.vutbr.cz/~lengal/idm/grafy-zaklady.pdf>

## ► Primův algoritmus wiki

[https://sk.wikipedia.org/wiki/Primov\\_algoritmus](https://sk.wikipedia.org/wiki/Primov_algoritmus)

## ► Primův algoritmus video

[https://www.youtube.com/watch?v=5pE\\_0Git5AM](https://www.youtube.com/watch?v=5pE_0Git5AM)