MA2BP_PDM1 Diskrétní matematika 1

3. Stromy, hledání minimální kostry, nejkratší cesta

Lukáš Másilko

Středisko pro pomoc studentům se specifickými nároky
Masarykova univerzita

17. 10. 2017

Program prezentace

1 Vlastnosti stromů

2 Kostra grafu

3 Algoritmy pro nalezení minimální kostry
- Kruskalův algoritmus
- Borůvkův algoritmus
- Jarníkův algoritmus

4 Hledání nejkratší cesty
- Dijkstrův algoritmus

5 Použité zdroje

Základní pojmy

Definice 4.1, 4.2, 4.3 (MILKOVÁ):

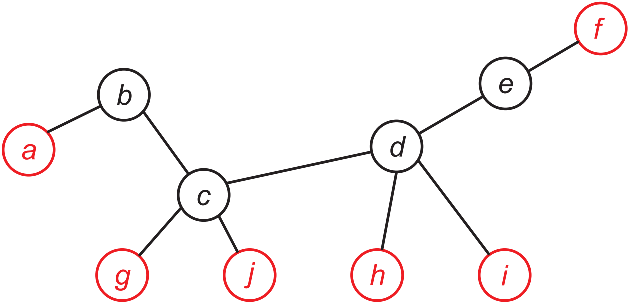
1. **Les** je graf, který neobsahuje kružnici.
2. **Strom** je souvislý graf, který neobsahuje kružnici.
3. Buď $T = (V, E)$ strom. Vrchol $u \in V$ stupně 1 nazýváme **listem**, vrchol $v \in V$ stupně většího než 1 nazývám **vnitřním vrcholem** stromu T.

Poznámka:

- Strom obsahující pouze jeden vrchol nazýváme **triviálním** stromem.
- Každá komponenta lesa je strom.

Ukázka stromu

Příklad stromu: viz následující obrázek. Vrcholy a, f, g, j, h, i jsou listy.



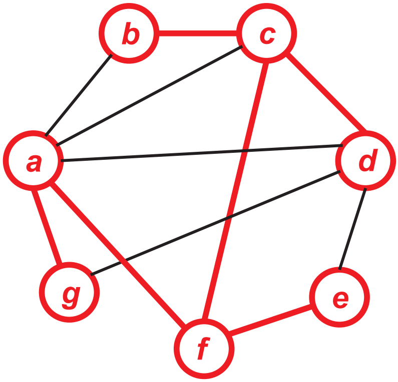
Charakteristika stromů

Věta 4.1 (MILKOVÁ): Pro každý graf $G = (V, E)$ jsou následující podmínky ekvivalentní:

1. Graf G je strom.
2. Pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y.
3. Graf G je souvislý a každá jeho hrana je most.
4. Graf G je souvislý a platí vztah $|V| = |E| + 1$.
5. Graf G neobsahuje kružnici a každý graf vzniklý z grafu G přidáním hrany, která spojuje libovolné dva vrcholy grafu G, které nejsou sousední, již kružnici obsahuje.

Definice 4.8 (MILKOVÁ): Libovolný strom $T = (V, E')$, který je podgrafem grafu $G = (V, E)$, nazýváme **kostra grafu** G.

Příklad kostry: viz barevně vyznačený podgraf na následujícím obrázku.



Souvislost grafu a kostra

Věta 4.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je souvislý právě tehdy, když obsahuje alespoň jednu kostru.

Důkaz: Tvrzení je ve tvaru ekvivalence, je tedy třeba dokázat oba směry implikace.

1. G je souvislý $\Rightarrow$ G obsahuje alespoň jednu kostru.
2. G obsahuje alespoň jednu kostru $\Rightarrow$ G je souvislý.

Souvislost grafu a kostra

Věta 4.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je souvislý právě tehdy, když obsahuje alespoň jednu kostru.

Důkaz: Tvrzení je ve tvaru ekvivalence, je tedy třeba dokázat oba směry implikace.

1. G je souvislý $\Rightarrow$ G obsahuje alespoň jednu kostru.
2. G obsahuje alespoň jednu kostru $\Rightarrow$ G je souvislý.

Proveďme důkaz obou implikací:

"$\Leftarrow$:" Předpokládejme, že graf G obsahuje alespoň jednu kostru.

Souvislost grafu a kostra

Věta 4.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je souvislý právě tehdy, když obsahuje alespoň jednu kostru.

Důkaz: Tvrzení je ve tvaru ekvivalence, je tedy třeba dokázat oba směry implikace.

1. G je souvislý $\Rightarrow$ G obsahuje alespoň jednu kostru.
2. G obsahuje alespoň jednu kostru $\Rightarrow$ G je souvislý.

Proveďme důkaz obou implikací:

“$\Leftarrow$:” Předpokládejme, že graf G obsahuje alespoň jednu kostru.

- Dle Definice 4.8 je kostra $K = (V, E')$ podgrafem obsahujícím všechny vrcholy původního grafu G.
-

Souvislost grafu a kostra

Věta 4.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je souvislý právě tehdy, když obsahuje alespoň jednu kostru.

Důkaz: Tvrzení je ve tvaru ekvivalence, je tedy třeba dokázat oba směry implikace.

1. G je souvislý $\Rightarrow$ G obsahuje alespoň jednu kostru.
2. G obsahuje alespoň jednu kostru $\Rightarrow$ G je souvislý.

Proveďme důkaz obou implikací:

"$\Leftarrow$:" Předpokládejme, že graf G obsahuje alespoň jednu kostru.

- Dle Definice 4.8 je kostra $K = (V, E')$ podgrafem obsahujícím všechny vrcholy původního grafu G.
- Navíc je K stromem, tudíž K je souvislý (dle definice stromu).

Souvislost grafu a kostra

Věta 4.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je souvislý právě tehdy, když obsahuje alespoň jednu kostru.

Důkaz: Tvrzení je ve tvaru ekvivalence, je tedy třeba dokázat oba směry implikace.

1. G je souvislý $\Rightarrow$ G obsahuje alespoň jednu kostru.
2. G obsahuje alespoň jednu kostru $\Rightarrow$ G je souvislý.

Proveďme důkaz obou implikací:

"$\Leftarrow$:" Předpokládejme, že graf G obsahuje alespoň jednu kostru.

- Dle Definice 4.8 je kostra $K = (V, E')$ podgrafem obsahujícím všechny vrcholy původního grafu G.
- Navíc je K stromem, tudíž K je souvislý (dle definice stromu).

Závěr: Samotný graf G je tedy souvislý.

Pokračování důkazu

"$\Rightarrow$:" Předpokládejme, že graf G je souvislý.

Důkaz faktu, že obsahuje alespoň jednu kostru, provedeme **matematickou indukcí** vzhledem k počtu hran $m \geq 0$.

-
-
- 1.
  2.

Pokračování důkazu

"$\Rightarrow$:" Předpokládejme, že graf G je souvislý.

Důkaz faktu, že obsahuje alespoň jednu kostru, provedeme **matematickou indukcí** vzhledem k počtu hran $m \geq 0$.

- **Báze:** $m = 0$, souvislý graf G má 0 hran $\Rightarrow$ jde o triviální strom s jedním izolovaným vrcholem. Kostrou grafu G je graf G samotný.
-
-
  1.
  2.

Pokračování důkazu

"$\Rightarrow$:" Předpokládejme, že graf G je souvislý.

Důkaz faktu, že obsahuje alespoň jednu kostru, provedeme **matematickou indukcí** vzhledem k počtu hran $m \geq 0$.

- **Báze:** $m = 0$, souvislý graf G má 0 hran $\Rightarrow$ jde o triviální strom s jedním izolovaným vrcholem. Kostrou grafu G je graf G samotný.
- **Indukční předpoklad:** nechť tvrzení platí pro souvislý graf G s m hranami, tj.
  (*) souvislý graf G o m hranách obsahuje alespoň jednu kostru.
- 1
  2

Pokračování důkazu

"$\Rightarrow$:" Předpokládejme, že graf G je souvislý.

Důkaz faktu, že obsahuje alespoň jednu kostru, provedeme **matematickou indukcí** vzhledem k počtu hran $m \geq 0$.

- **Báze:** $m = 0$, souvislý graf G má 0 hran $\Rightarrow$ jde o triviální strom s jedním izolovaným vrcholem. Kostrou grafu G je graf G samotný.
- **Indukční předpoklad:** nechť tvrzení platí pro souvislý graf G s m hranami, tj.
  (*) souvislý graf G o m hranách obsahuje alespoň jednu kostru.
- **Indukční krok:** Uvažujme souvislý graf G' o $m + 1$ hranách.
  1.
  2.

Pokračování důkazu

"$\Rightarrow$:" Předpokládejme, že graf G je souvislý.

Důkaz faktu, že obsahuje alespoň jednu kostru, provedeme **matematickou indukcí** vzhledem k počtu hran $m \geq 0$.

- **Báze:** $m = 0$, souvislý graf G má 0 hran $\Rightarrow$ jde o triviální strom s jedním izolovaným vrcholem. Kostrou grafu G je graf G samotný.
- **Indukční předpoklad:** nechť tvrzení platí pro souvislý graf G s m hranami, tj.
  (*) souvislý graf G o m hranách obsahuje alespoň jednu kostru.
- **Indukční krok:** Uvažujme souvislý graf G' o $m + 1$ hranách.
  1. G' neobsahuje kružnici $\Rightarrow$ G' je strom a jeho kostrou je G' samotný.
  2.

Pokračování důkazu

"$\Rightarrow$:" Předpokládejme, že graf G je souvislý.

Důkaz faktu, že obsahuje alespoň jednu kostru, provedeme **matematickou indukcí** vzhledem k počtu hran $m \geq 0$.

- **Báze:** $m = 0$, souvislý graf G má 0 hran $\Rightarrow$ jde o triviální strom s jedním izolovaným vrcholem. Kostrou grafu G je graf G samotný.
- **Indukční předpoklad:** nechť tvrzení platí pro souvislý graf G s m hranami, tj.
  (*) souvislý graf G o m hranách obsahuje alespoň jednu kostru.
- **Indukční krok:** Uvažujme souvislý graf G' o $m + 1$ hranách.
  1. G' neobsahuje kružnici $\Rightarrow$ G' je strom a jeho kostrou je G' samotný.
  2. G' obsahuje kružnici C
     $\Rightarrow$ odebráním libovolné hrany $e \in C$ dostáváme opět souvislý graf $G' - e$, který už má pouze m hran

Pokračování důkazu

"⇒:" Předpokládejme, že graf G je souvislý.

Důkaz faktu, že obsahuje alespoň jednu kostru, provedeme **matematickou indukcí** vzhledem k počtu hran $m \geq 0$.

- **Báze:** $m = 0$, souvislý graf G má 0 hran $\Rightarrow$ jde o triviální strom s jedním izolovaným vrcholem. Kostrou grafu G je graf G samotný.
- **Indukční předpoklad:** nechť tvrzení platí pro souvislý graf G s m hranami, tj.
  (*) souvislý graf G o m hranách obsahuje alespoň jednu kostru.
- **Indukční krok:** Uvažujme souvislý graf G' o $m + 1$ hranách.
  1. G' neobsahuje kružnici $\Rightarrow$ G' je strom a jeho kostrou je G' samotný.
  2. G' obsahuje kružnici C
     $\Rightarrow$ odebráním libovolné hrany $e \in C$ dostáváme opět souvislý graf $G' - e$, který už má pouze m hran
     $\Rightarrow$ dle indukčního předpokladu a tvrzení (*) obsahuje graf $G' - e$ alespoň jednu kostru. Ta je zároveň i kostrou grafu G'.

Pokračování důkazu

"$\Rightarrow$:" Předpokládejme, že graf G je souvislý.

Důkaz faktu, že obsahuje alespoň jednu kostru, provedeme **matematickou indukcí** vzhledem k počtu hran $m \geq 0$.

- **Báze:** $m = 0$, souvislý graf G má 0 hran $\Rightarrow$ jde o triviální strom s jedním izolovaným vrcholem. Kostrou grafu G je graf G samotný.
- **Indukční předpoklad:** nechť tvrzení platí pro souvislý graf G s m hranami, tj.
  (*) souvislý graf G o m hranách obsahuje alespoň jednu kostru.
- **Indukční krok:** Uvažujme souvislý graf G' o $m + 1$ hranách.
  1. G' neobsahuje kružnici $\Rightarrow$ G' je strom a jeho kostrou je G' samotný.
  2. G' obsahuje kružnici C
     $\Rightarrow$ odebráním libovolné hrany $e \in C$ dostáváme opět souvislý graf $G' - e$, který už má pouze m hran
     $\Rightarrow$ dle indukčního předpokladu a tvrzení (*) obsahuje graf $G' - e$ alespoň jednu kostru. Ta je zároveň i kostrou grafu G'.

Závěr: Graf G tedy obsahuje alespoň jednu kostru.

Podmínka pro souvislý graf

Důsledek 4.1 (MILKOVÁ): V souvislém grafu $G = (V, E)$ platí vztah $|E| \geq |V| - 1$.

Důkaz:

Podmínka pro souvislý graf

Důsledek 4.1 (MILKOVÁ): V souvislém grafu $G = (V, E)$ platí vztah $|E| \geq |V| - 1$.

Důkaz: Souvislý graf obsahuje alespoň jednu kostru $K = (V, E')$, kde $E' \subseteq E$ (dle Věty 4.2).

Podmínka pro souvislý graf

Důsledek 4.1 (MILKOVÁ): V souvislém grafu $G = (V, E)$ platí vztah $|E| \geq |V| - 1$.

Důkaz: Souvislý graf obsahuje alespoň jednu kostru $K = (V, E')$, kde $E' \subseteq E$ (dle Věty 4.2). Pro kostru K platí $|E'| = |V| - 1$.

Podmínka pro souvislý graf

Důsledek 4.1 (MILKOVÁ): V souvislém grafu $G = (V, E)$ platí vztah $|E| \geq |V| - 1$.

Důkaz: Souvislý graf obsahuje alespoň jednu kostru $K = (V, E')$, kde $E' \subseteq E$ (dle Věty 4.2). Pro kostru K platí $|E'| = |V| - 1$. Protože $E' \subseteq E$, platí $|E| \geq |V| - 1$.

Podmínka pro souvislý graf

Důsledek 4.1 (MILKOVÁ): V souvislém grafu $G = (V, E)$ platí vztah $|E| \geq |V| - 1$.

Důkaz: Souvislý graf obsahuje alespoň jednu kostru $K = (V, E')$, kde $E' \subseteq E$ (dle Věty 4.2). Pro kostru K platí $|E'| = |V| - 1$. Protože $E' \subseteq E$, platí $|E| \geq |V| - 1$.

Příklad: Je dáno skóre $(1, 1, 1, 1, 2, 2, 2)$ grafu G. Může být graf G souvislý?

Řešení:

Podmínka pro souvislý graf

Důsledek 4.1 (MILKOVÁ): V souvislém grafu $G = (V, E)$ platí vztah $|E| \geq |V| - 1$.

Důkaz: Souvislý graf obsahuje alespoň jednu kostru $K = (V, E')$, kde $E' \subseteq E$ (dle Věty 4.2). Pro kostru K platí $|E'| = |V| - 1$. Protože $E' \subseteq E$, platí $|E| \geq |V| - 1$.

Příklad: Je dáno skóre $(1, 1, 1, 1, 2, 2, 2)$ grafu G. Může být graf G souvislý?

Řešení: Počet vrcholů $|V| = 7$. Počet hran spočítáme: $|E| = (1 + 1 + 1 + 1 + 2 + 2 + 2)/2 = 5$.

Ověříme podmínku pro souvislý graf $|E| \geq |V| - 1$ z Věty 4.1. Není pravda, že $5 \geq 7 - 1$. Proto graf se zadaným skóre nemůže být souvislý.

Počet koster v souvislém grafu

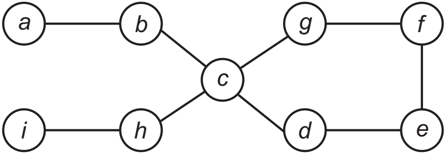
Platí zřejmě následující úvahy:

1. Obsahuje-li souvislý graf právě jednu kružnici délky k, pak obsahuje k koster.
2. Obsahuje-li souvislý graf p po dvou hranově disjunktních kružnic $C_1, C_2, \ldots, C_p$, jejichž délky jsou postupně $k_1, k_2, \ldots, k_p$, pak obsahuje $k_1 \cdot k_2 \cdot \cdots \cdot k_p$ koster.
3. Obsahuje-li graf G právě jednu kostru, pak je graf G stromem.

Věta 4.3 (Arthur Cayley): Počet koster úplného grafu K_n, $n \geq 2$, je n^{n-2}.

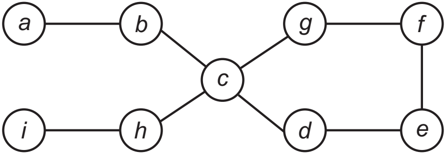
Příklad 1

Příklad 1: Určete počet koster v následujícím grafu.



Příklad 1

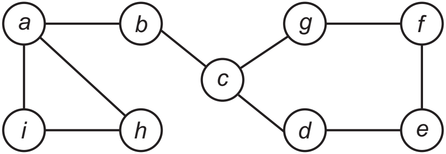
Příklad 1: Určete počet koster v následujícím grafu.



Řešení: V grafu je pouze jedna kružnice délky 5. Proto je počet koster roven 5.

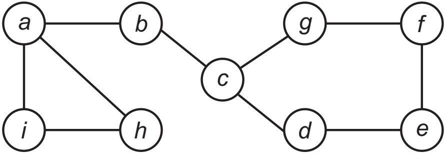
Příklad 2

Příklad 2: Určete počet koster v následujícím grafu.



Příklad 2

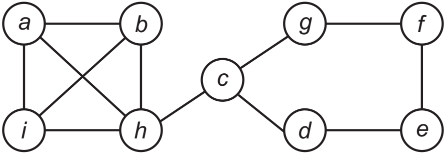
Příklad 2: Určete počet koster v následujícím grafu.



Řešení: V grafu jsou dvě po hranách disjunktní kružnice, jedna délky 3, druhá délky 5. Proto je počet koster roven $5 \cdot 3 = 15$.

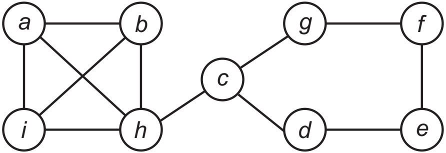
Příklad 3

Příklad 3: Určete počet koster v následujícím grafu.



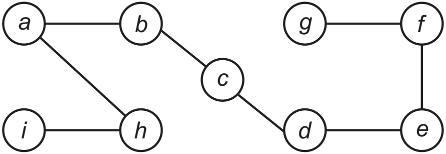
Příklad 3

Příklad 3: Určete počet koster v následujícím grafu.



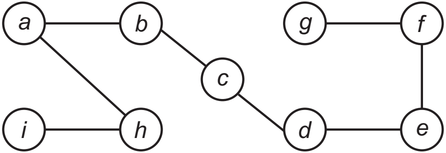
Řešení: V grafu jsou vidět dva zajímavé podgrafy: úplný graf K_4 a kružnice délky 5. Víme, že pro úplný graf K_4 platí, že počet koster je $4^2 = 16$. Obsahuje-li graf navíc ještě kružnici délky 5, pak je počet koster celého grafu roven $16 \cdot 5 = 80$.

Příklad 4: Určete počet koster v následujícím grafu.



Příklad 4

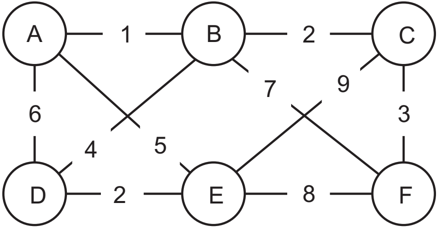
Příklad 4: Určete počet koster v následujícím grafu.



Řešení: Graf je stromem, počet koster je tedy roven 1.

Prohrnování silnic

V království je 6 měst – označme je A, B, C, D, E, F, a některá z nich jsou spojena přímou silnicí. Křižovatky jsou pouze ve městech. Mezi některými městy přímá silnice nevede, ale z každého města se dá po silnicích dostat do libovolného jiného. V noci se přihnala se vánice a zasněžila všechny silnice v celém království. Napadlo až metr sněhu. Odhrabovačů je málo a takovou sněhovou nadílku budou odklízet ještě hodně dlouho. Rozhodněte, které silnice se mají odhrabat jako první, aby mezi každýma dvěma městy vedla sjízdná silnice. Přikládáme plán království s délkou jednotlivých cest.



Problém minimální kostry

Mějme ohodnocený souvislý graf $G = (V, E)$. Každé hraně $e \in E$ je dáno reálné číslo $w(e)$, tzv. ohodnocení hrany.

Mezi všemi kostrami grafu G najděte kostru $H = (V, E')$ takovou, že součet ohodnocení jejích hran $w(H) = \sum_{e \in E'} w(e)$ nabývá minimální hodnoty.

Kostru H nazveme **minimální kostrou** grafu G a $w(H)$ **cenou kostry** H.

Poznámky:

1. Minimálních koster může být více.
2. V praxi se setkáváme většinou s tím, že ohodnocení hran je nezáporné.
3. Čeští matematici stáli u zrodu nejznámějších algoritmů pro nalezení minimální kostry. Prvním byl ve 20. letech 20. století brněnský matematik Otakar Borůvka (elektrifikace na jižní a západní Moravě).

Kruskalův algoritmus

Věta 4.18 (FUCHS): Buď (V, E, w) konečný souvislý graf s ohodnocením hran w. Všechny hrany $e_1, \ldots, e_n \in E$ sestavme do posloupnosti tak, aby $w(e_1) \leq w(e_2) \leq \cdots \leq w(e_n)$. Algoritmus pro nalezení minimální kostry grafu G je následující:

(1) Polož $E_0 = \emptyset, i = 1$.

(2) Obsahuje graf $(V, E_{i-1} \cup \{e_i\})$ kružnici? ANO: krok (4). NE: krok (3).

(3) Polož $E_i = E_{i-1} \cup \{e_i\}$. Jdi na krok (5).

(4) Polož $E_i = E_{i-1}$.

(5) Je $i = n$? ANO: Jdi na krok (6). NE: Jdi na krok (8).

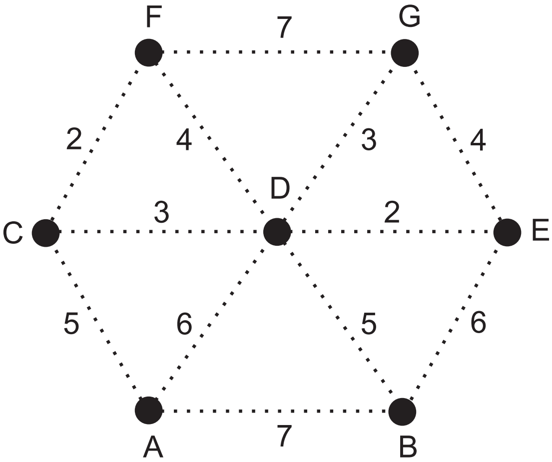
(6) Polož $E_i = K$.

(7) KONEC. (U, K) je minimální kostra.

(8) Zvětši hodnotu i o jedničku a jdi na krok (2).

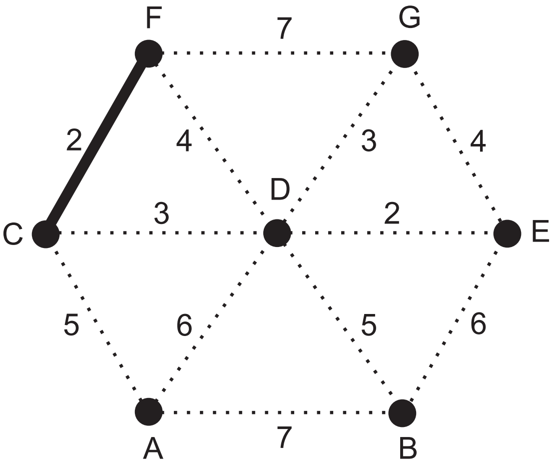
Příklad použití Kruskalova algoritmu

Počátek algoritmu – neorientovaný graf o sedmi uzlech A, B, C, D, E, F, G



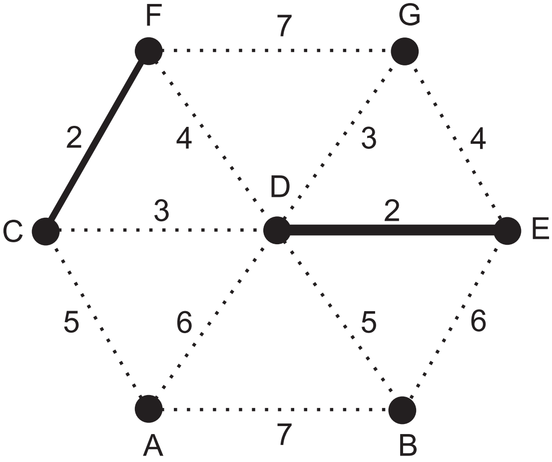
Příklad použití Kruskalova algoritmu

Aktuálně zpracovávaná hrana: C–2–F (patří do kostry)



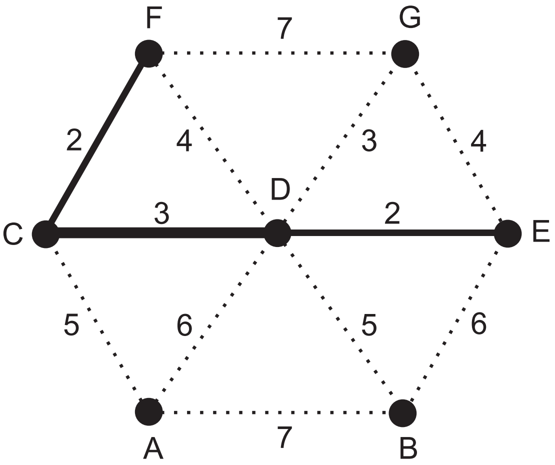
Příklad použití Kruskalova algoritmu

Aktuálně zpracovávaná hrana: D–2–E (patří do kostry)



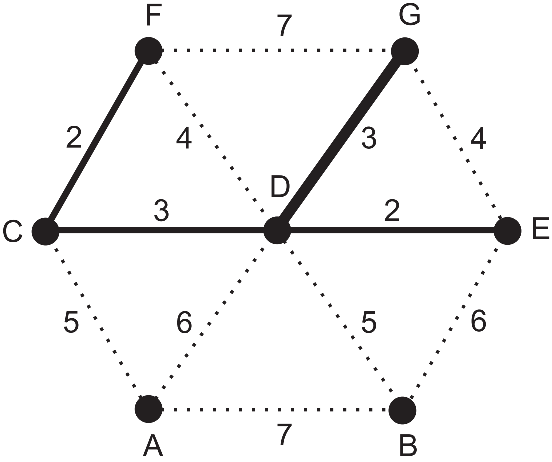
Příklad použití Kruskalova algoritmu

Aktuálně zpracovávaná hrana: C–3–D (patří do kostry)



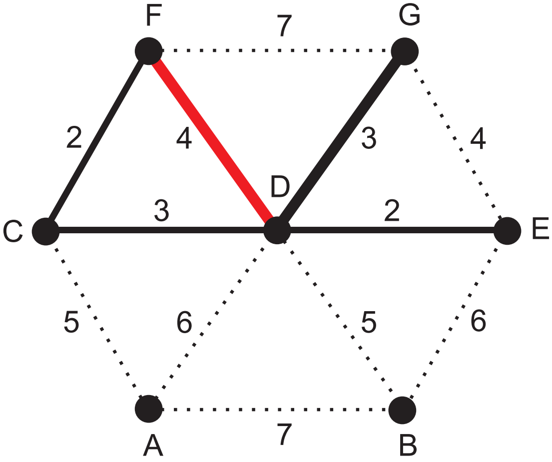
Příklad použití Kruskalova algoritmu

Aktuálně zpracovávaná hrana: D–3–G (patří do kostry)



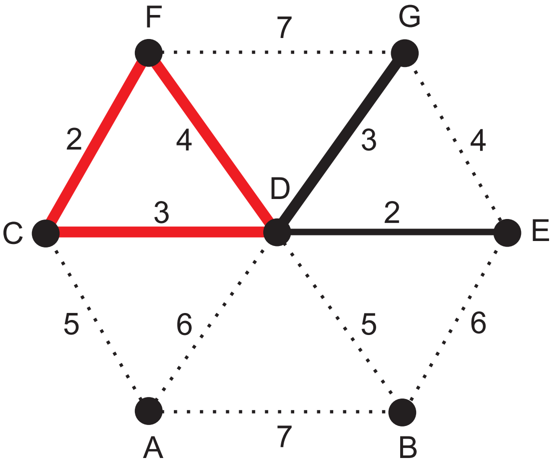
Příklad použití Kruskalova algoritmu

Aktuálně zpracovávaná hrana: D–4–F (nepatří do kostry)



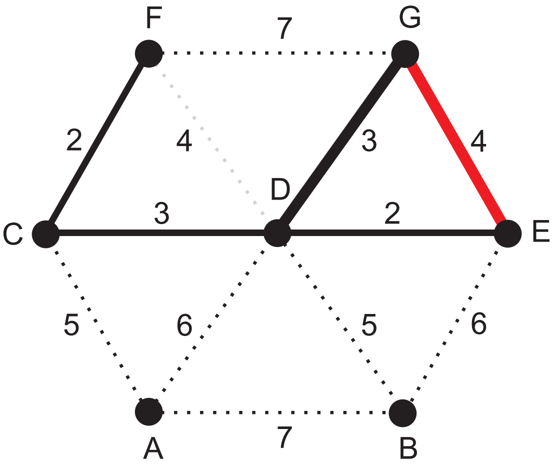
Příklad použití Kruskalova algoritmu

Aktuálně zpracovávaná hrana: D–4–F (přidáním hrany D–4–F by vznikla kružnice CDFC)



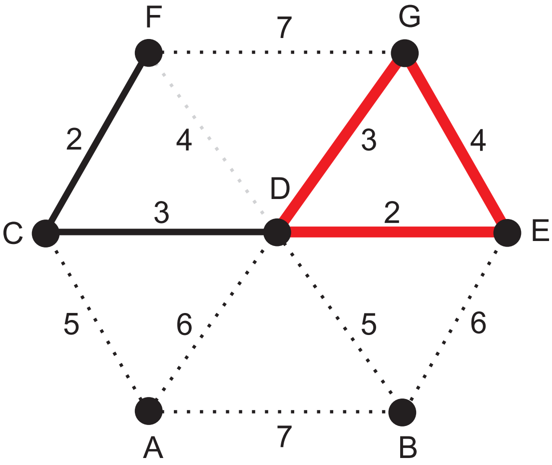
Příklad použití Kruskalova algoritmu

Aktuálně zpracovávaná hrana: E–4–G (nepatří do kostry)

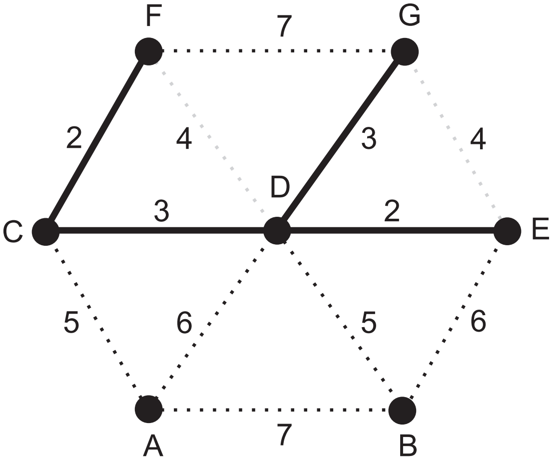


Příklad použití Kruskalova algoritmu

Aktuálně zpracovávaná hrana: E–4–G (přidáním hrany E–4–G by vznikla kružnice DEGD)

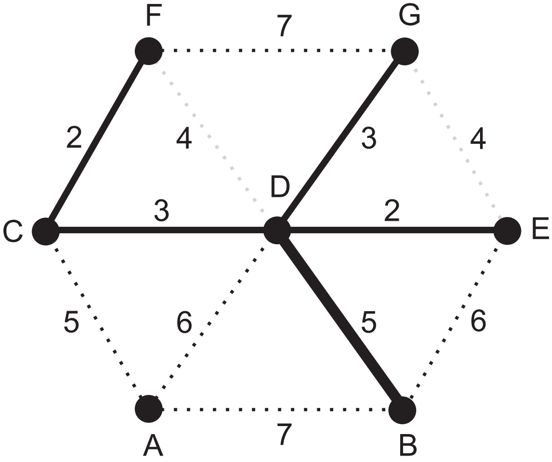


Příklad použití Kruskalova algoritmu



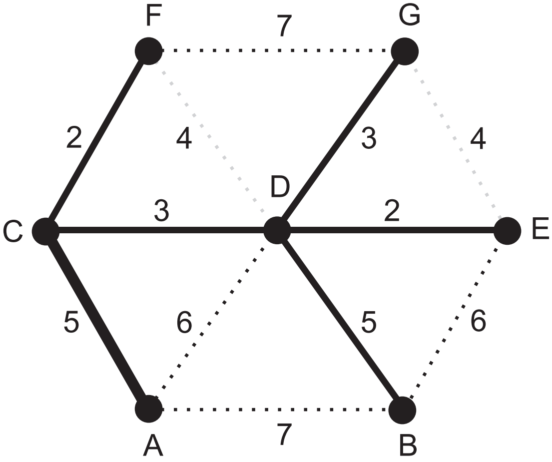
Příklad použití Kruskalova algoritmu

Aktuálně zpracovávaná hrana: B–5–D (patří do kostry)



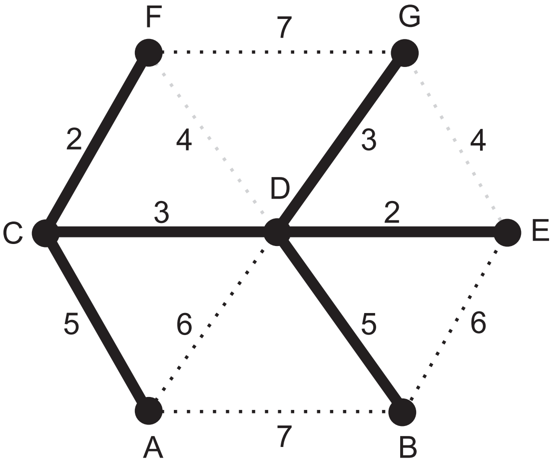
Příklad použití Kruskalova algoritmu

Aktuálně zpracovávaná hrana: A-5-C (patří do kostry)

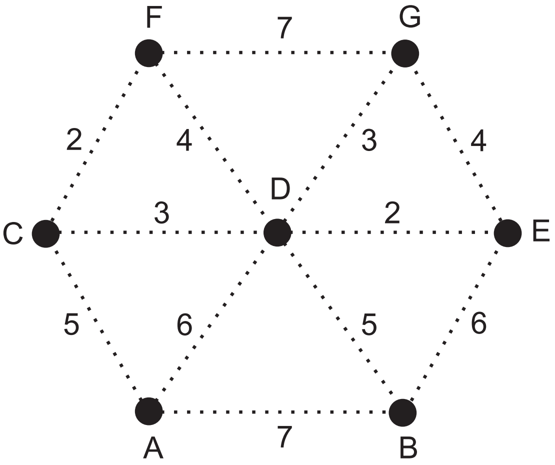


Příklad použití Kruskalova algoritmu

Konec algoritmu, kostra je úplná

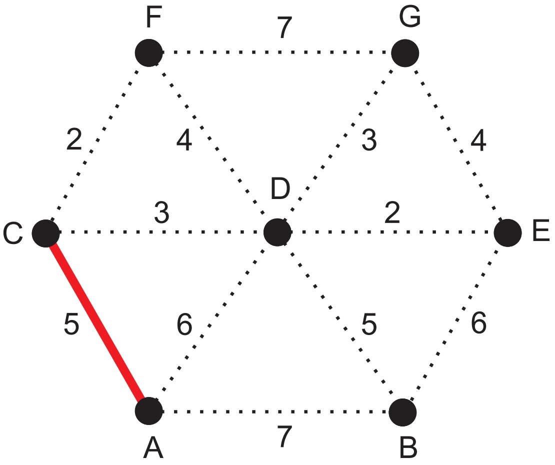


Počátek algoritmu – neorientovaný graf o sedmi uzlech (stromech) A, B, C, D, E, F, G



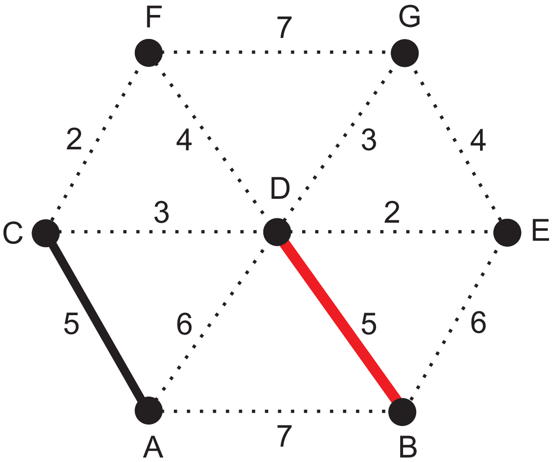
Příklad použití Borůvkova algoritmu

Aktuálně zpracovávaný "strom" $\{A\}$: nejkratší hrana A-5-C ke stromu $\{C\}$ $\Rightarrow$ nový strom $\{A, C\}$



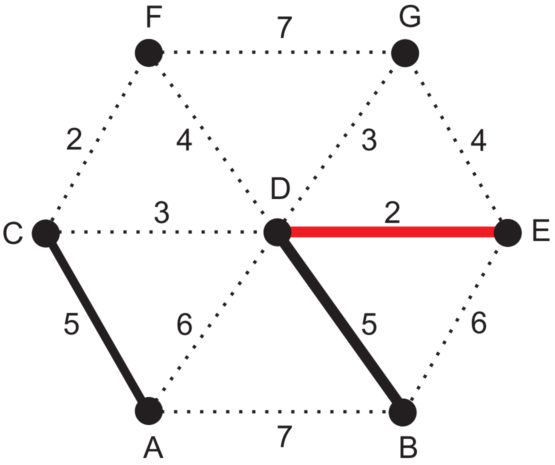
Příklad použití Borůvkova algoritmu

Aktuálně zpracovávaný "strom" {B}: nejkratší hrana B-5-D ke stromu {D} ⇒ nový strom {B, D}



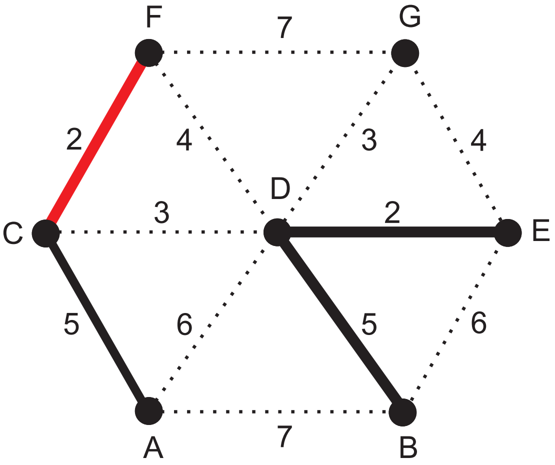
Příklad použití Borůvkova algoritmu

Aktuálně zpracovávaný "strom" {E}: nejkratší hrana E-2-D ke stromu {B, D} ⇒ nový strom {B, D, E}



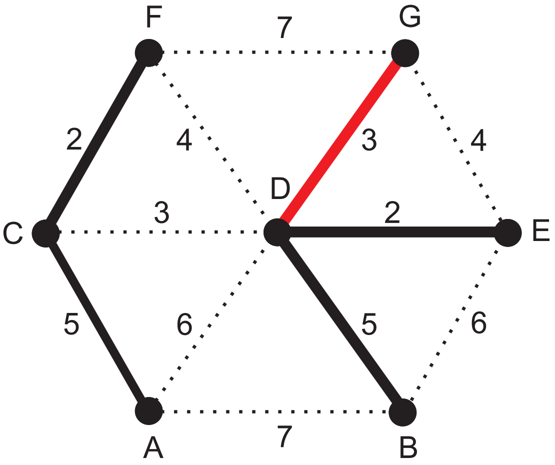
Příklad použití Borůvkova algoritmu

Aktuálně zpracovávaný "strom" $\{F\}$: nejkratší hrana F-2-C ke stromu $\{A, C\}$ $\Rightarrow$ nový strom $\{A, C, F\}$



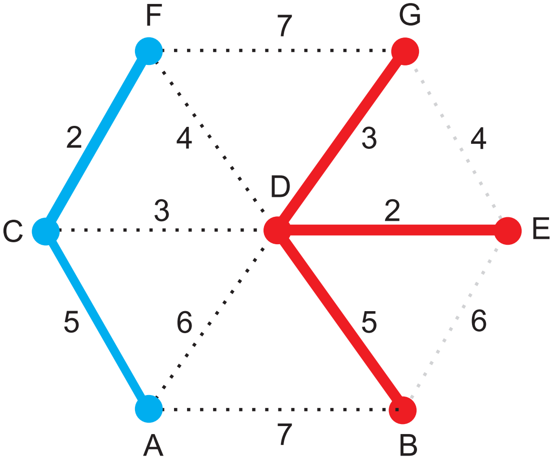
Příklad použití Borůvkova algoritmu

Aktuálně zpracovávaný "strom" {G}: nejkratší hrana G-3-D ke stromu {B, D, E} ⇒ nový strom {B, D, E, G}



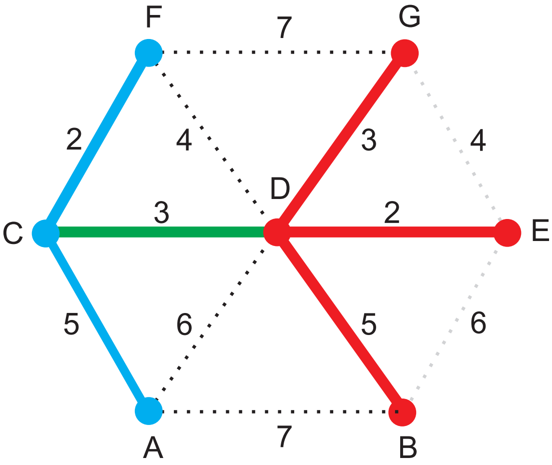
Příklad použití Borůvkova algoritmu

Sumarizace: dva stromy, modrý: {A, C, F}, červený: {B, D, E, G}



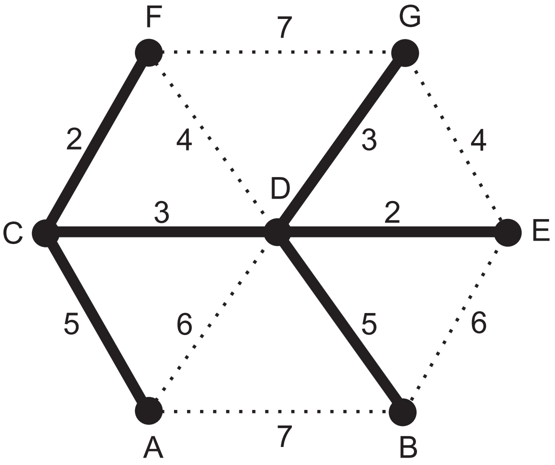
Příklad použití Borůvkova algoritmu

Aktuálně zpracovávaný "strom" {A, C, F}: nejkratší hrana C-3-D ke stromu {B, D, E, F}



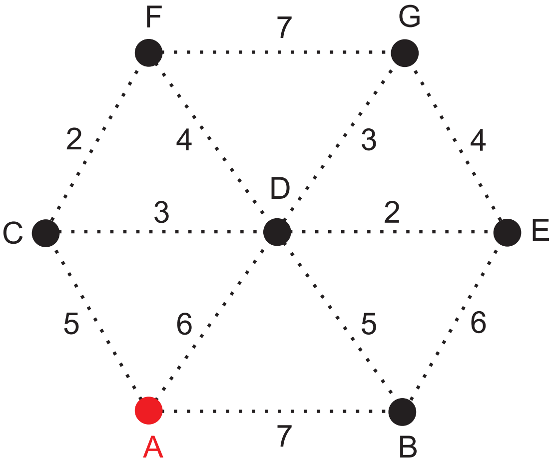
Příklad použití Borůvkova algoritmu

Konec algoritmu – spojení posledních dvou stromů ⇒ minimální kostra.



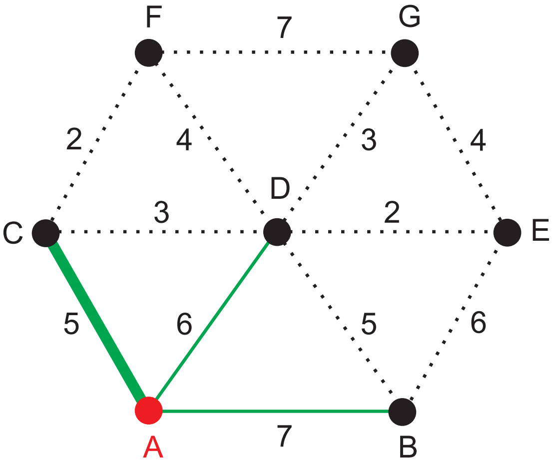
Příklad použití Jarníkova algoritmu

Počátek algoritmu – neorientovaný graf G o vrcholech A, B, C, D, E, F, G a vyznačeným stromem $S = \{A\}$



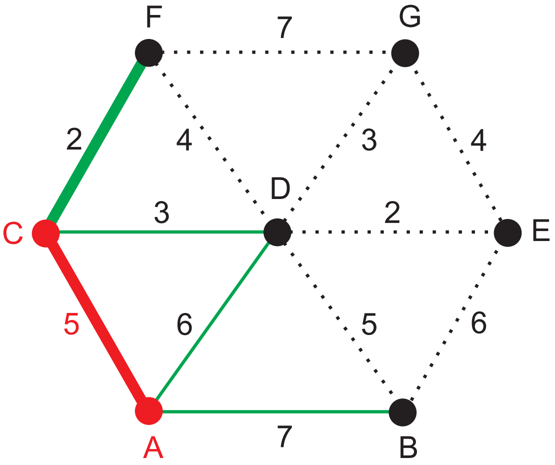
Příklad použití Jarníkova algoritmu

Aktuálně zpracovávaný "strom" $S = \{A\}$: nejkratší hrana od S do $G - S$ je hrana A-5-C $\Rightarrow$ nový strom $S = \{A, C\}$



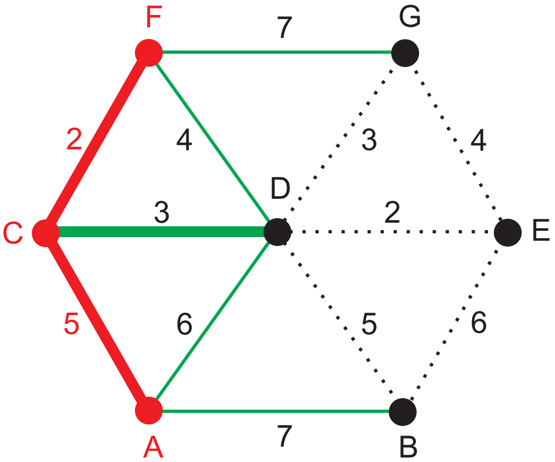
Příklad použití Jarníkova algoritmu

Aktuálně zpracovávaný "strom" $S = \{A, C\}$: nejkratší hrana od S do $G - S$ je hrana C-2-F $\Rightarrow$ nový strom $S = \{A, C, F\}$



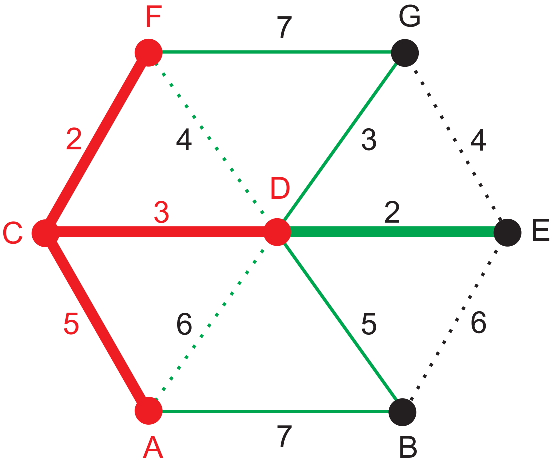
Příklad použití Jarníkova algoritmu

Aktuálně zpracovávaný "strom" $S = \{A, C, F\}$: nejkratší hrana od S do $G - S$ je hrana C-3-D $\Rightarrow$ nový strom $S = \{A, C, D, F\}$



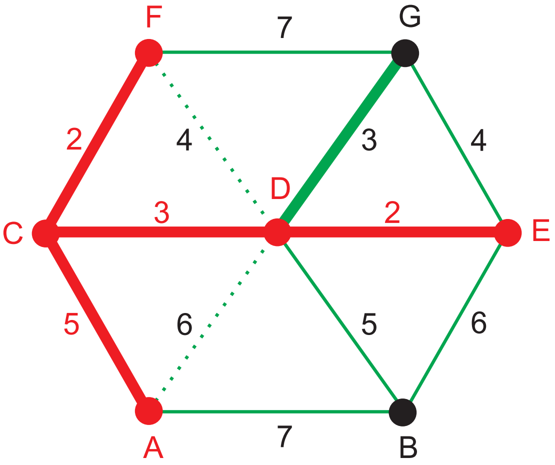
Příklad použití Jarníkova algoritmu

Aktuálně zpracovávaný "strom" $S = \{A, C, D, F\}$: nejkratší hrana od S do $G - S$ je hrana D-2-E $\Rightarrow$ nový strom $S = \{A, C, D, E, F\}$



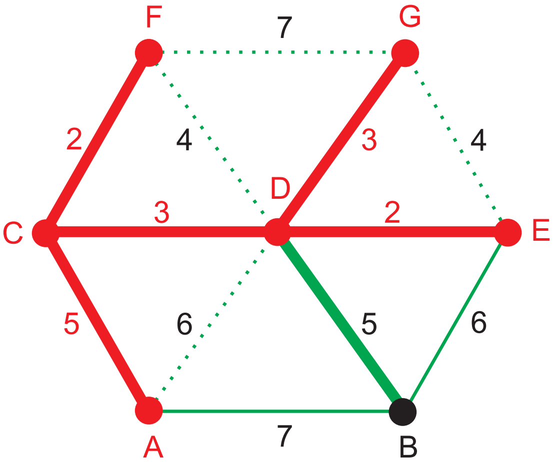
Příklad použití Jarníkova algoritmu

Aktuálně zpracovávaný "strom" $S = \{A, C, D, E, F\}$: nejkratší hrana od S do $G - S$ je hrana D-3-G $\Rightarrow$ nový strom $S = \{A, C, D, E, F, G\}$



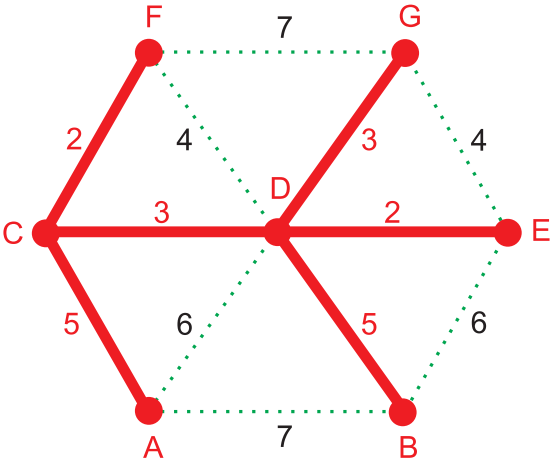
Příklad použití Jarníkova algoritmu

Aktuálně zpracovávaný "strom" $S = \{A, C, D, E, F, G\}$: nejkratší hrana od S do $G - S$ je hrana D-5-B $\Rightarrow$ nový strom $S = \{A, B, C, D, E, F, G\}$



Příklad použití Jarníkova algoritmu

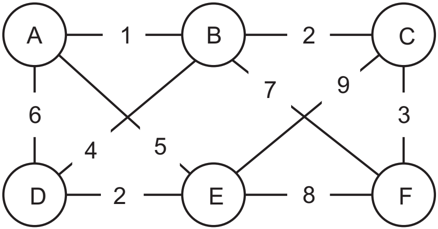
Konec algoritmu, ve stromu S jsou všechny vrcholy původního grafu G



Srovnání algoritmů

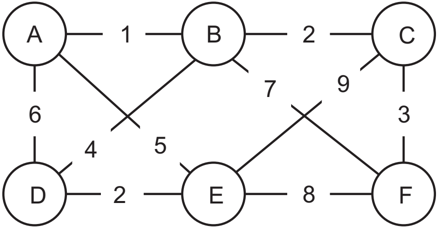
1. **Kruskalův algoritmus** nejprve uspořádá hrany podle ohodnocení, pak je prochází a když aktuální hrana nevytváří kružnici, přidá ji do minimální kostry. Pro grafy s menším počtem hran je nejefektivnější.
2. **Borůvkův algoritmus:** v každém kroku dochází ke spojení aktuálního stromu s nejbližším jiným stromem. S využitím speciálních datových struktur je pro grafy s velkým počtem hran velmi rychlý.
3. **Jarníkův algoritmus** je vlastně obdobou Borůvkova algoritmu s tím rozdílem, že v každém kroku bereme tentýž strom, který obohacujeme spojením s dalším stromem. Opět je výhodnější jej použít pro grafy s velkým počtem hran.

Prohrnování silnic – řešení

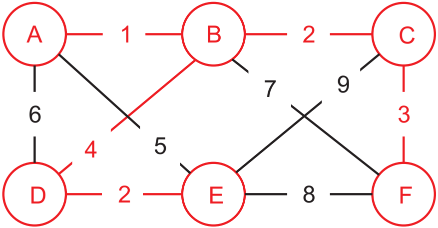


Řešení pomocí Kruskalova algoritmu:

Prohrnování silnic – řešení



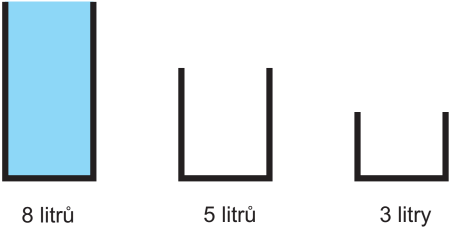
Řešení pomocí Kruskalova algoritmu:



Úloha o třech nádobách

Zadání: Mějme plnou nádobu vody o objemu 8 litrů a dvě prázdné nádoby s objemy 5 a 3 litry. Jakým způsobem musíme přelévat vodu tak, abychom nakonec dostali dvakrát 4 litry?

Poznámka: Ani jedna z nádob nemá měrnou stupnici.

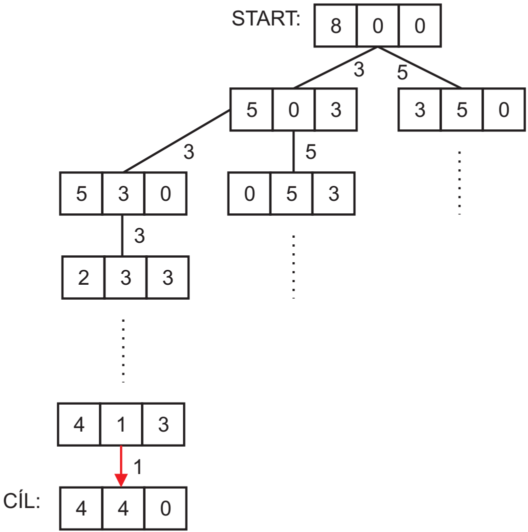


Autor prvního řešení z r. 1893: Georges Édouard Auguste Brunel (1856–1900)

Úloha o třech nádobách – jak řešit pomocí grafů

1. Objem tří nádob zapisovat do návěští vrcholů – trojice přirozených čísel (i, j, k) včetně nuly, přičemž $i \leq 8, j \leq 5, k \leq 3$.
2. Vrcholy jsou vlastně "stavy", v jakých mohou jednotlivé nádoby být. Je-li možné přejít z jednoho stavu do druhého, umístíme mezi ně hranu či šipku ("orientovanou hranu").
3. Existují hrany, u kterých záleží na pořadí počátečního a koncového vrcholu (viz červená šipka).
4. Řešením úlohy je nalezení cesty o co nejmenším počtu hran (šipek) ze stavu (8, 0, 0) do stavu (4, 4, 0).

Úloha o třech nádobách – stavové pole



Úloha o třech nádobách – řešení

Řešení:

$$
\begin{aligned}
(8,0,0) &\to_3 (5,0,3)\\
&\to_3 (5,3,0)\\
&\to_3 (2,3,3)\\
&\to_2 (2,5,1)\\
&\to_5 (7,0,1)\\
&\to_1 (7,1,0)\\
&\to_3 (4,1,3)\\
&\to_3 (4,4,0)
\end{aligned}
$$

Úloha Vlk, koza, zelí

Zadání: Vesničan se vrací z trhu domů. Má s sebou kozu, vlka a v ruce hlávku zelí. Najednou přišel k řece. Na břehu má přivázanou malou loď. Už chce nasednout, když tu ho náhle dobrá nálada opouští. Totiž, do té lodičky se všechno naráz nevejde. A když tu nechá vlka samotného, vlk sní kozu. Když tu nechá kozu, ta sní zelí. Jak dostane vlka, kozu i zelí na druhou stranu?

Do loďky se mu vejde jen jedna věc. A na žádném z břehů nesmí nechat samotného vlka s kozou nebo kozu a zelí.

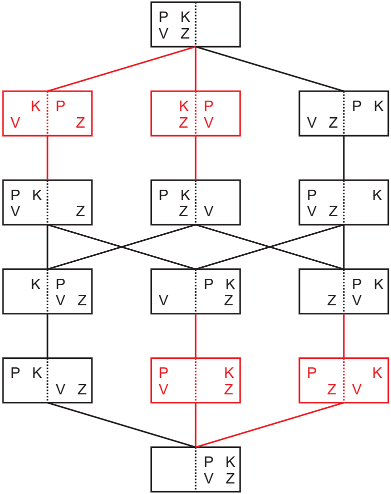
Úloha Vlk, koza, zelí

Zadání: Vesničan se vrací z trhu domů. Má s sebou kozu, vlka a v ruce hlávku zelí. Najednou přišel k řece. Na břehu má přivázanou malou loď. Už chce nasednout, když tu ho náhle dobrá nálada opouští. Totiž, do té lodičky se všechno naráz nevejde. A když tu nechá vlka samotného, vlk sní kozu. Když tu nechá kozu, ta sní zelí. Jak dostane vlka, kozu i zelí na druhou stranu?

Do loďky se mu vejde jen jedna věc. A na žádném z břehů nesmí nechat samotného vlka s kozou nebo kozu a zelí.

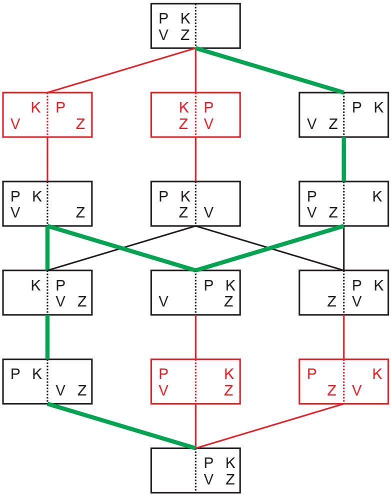
Řešení: Nejprve si označíme účastníky převozu. **V**lk, **K**oza, **Z**elí a **P**řevozník. Stavy grafu budou dvojice, které vzniknou rozkladem množiny $\{V, K, Z, P\}$ na dvě podmnožiny reprezentující objekty na levém a pravém břehu. Počátečním stavem je tedy dvojice $(\{V, K, Z, P\}, \emptyset)$, cílem je dojít k vrcholu $(\emptyset, \{V, K, Z, P\})$.

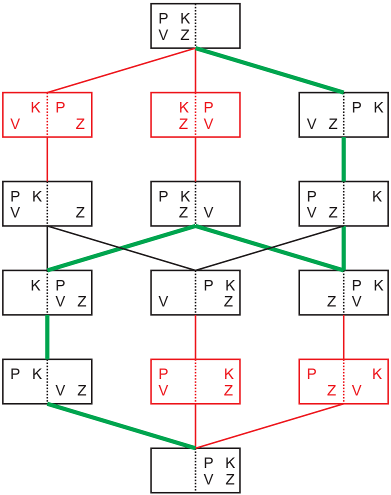
Úloha Vlk, koza, zelí – stavové pole



Červeně jsou označené nepovolené stavy a hrany s nimi incidentní.

Úloha Vlk, koza, zelí – řešení





Zeleně je vyznačena nejkratší cesta, která prochází povolenými stavy.

Hledání nejkratší cesty

Mějme ohodnocený (vážený) graf $G = (V, E)$. Každé hraně $e \in E$ je dáno reálné **nezáporné** číslo $w(e)$, tzv. ohodnocení (váha) hrany.

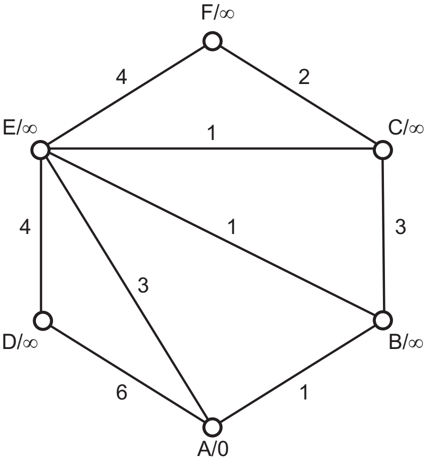
V teorii grafů se můžeme setkat s těmito velmi podobnými problémy:

1. **Shortest path:** Najděte nejkratší cestu z iniciálního vrcholu $s \in V$ do cílového vrcholu $t \in V$ v grafu G.
2. **Single source shortest path:** Najděte nejkratší cestu z iniciálního vrcholu $a \in V$ do všech ostatních uzlů grafu G.
3. **All pairs shortest path:** Najděte nejkratší cestu mezi každou dvojicí vrcholů grafu G.

Poznámka: K řešení prvních dvou problémů se používá *Dijkstrův algoritmus*. Je možné jej použít i pro řešení 3. problému (aplikujeme jej pro každý vrchol grafu zvlášť), efektivnější metodou je však *Floyd-Warshallův algoritmus*.

Dijkstrův algoritmus – inicializace

Na počátku algoritmu vložíme do návěští iniciálního uzlu A hodnotu $A/0$, do návěští ostatních uzlů X, jejichž vzdálenost od A zatím neznáme, zapíšeme X/∞, viz následující obrázek.



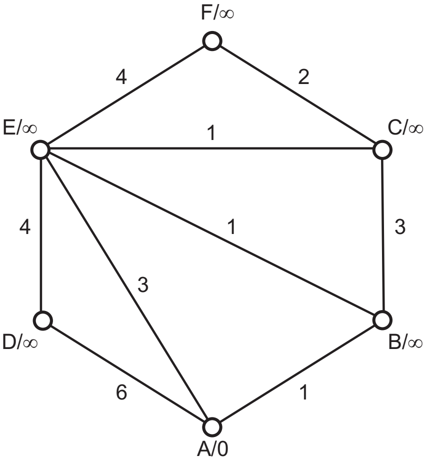
Dijkstrův algoritmus – procházení grafu

Opakovaně provádíme následující tři kroky, dokud nezpracujeme všechny vrcholy:

1. Mezi nezpracovanými vrcholy najdeme uzel X/n s nejmenší vzdáleností n od iniciálního vrcholu A.
2. Pro každou hranu e vedoucí z uzlu X/n do nezpracovaného vrcholu Y/m provedeme následující:
   - je-li $m > n + w(e)$, změníme aktuální vzdálenost uzlu Y od iniciálního uzlu A na $m = n + w(e)$,
   - v opačném případě ponecháme návěští uzlu Y beze změny.
3. Označíme uzel X jako zpracovaný.

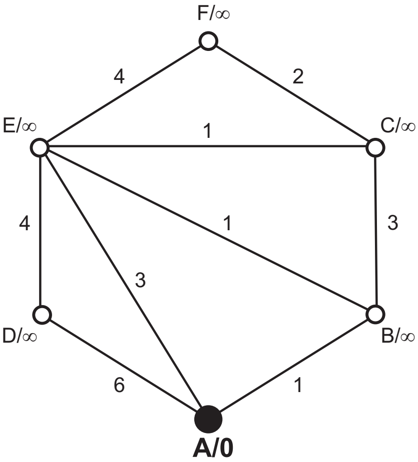
Příklad použití Dijkstrova algoritmu

Počátek algoritmu – neorientovaný graf o šesti uzlech A, B, C, D, E, F, přičemž vrchol A je iniciální.



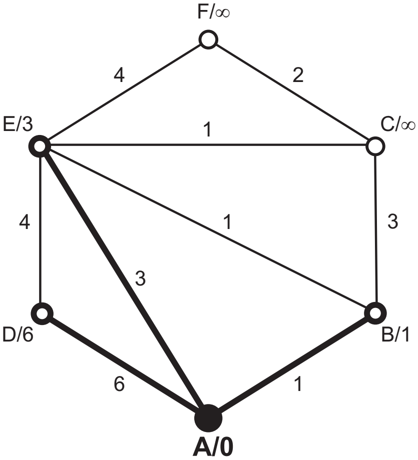
Příklad použití Dijkstrova algoritmu

Uzel A se aktuálně zpracovává



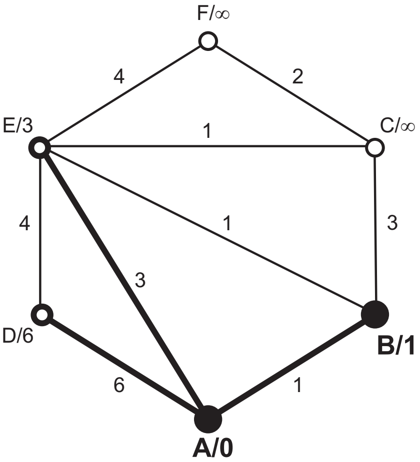
Příklad použití Dijkstrova algoritmu

Mění se návěští uzlů B, D, E (dosud nezpracovaní sousedé vrcholu A)



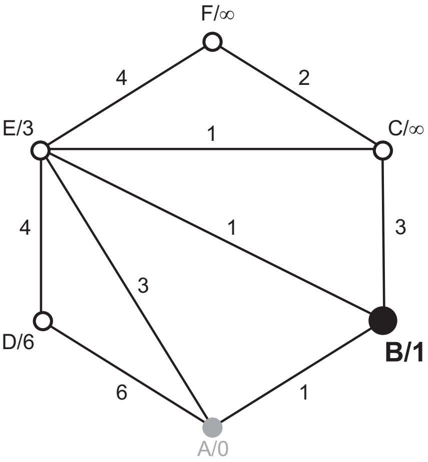
Příklad použití Dijkstrova algoritmu

Z nezpracovaných vrcholů má nejmenší vzdálenost uzel B



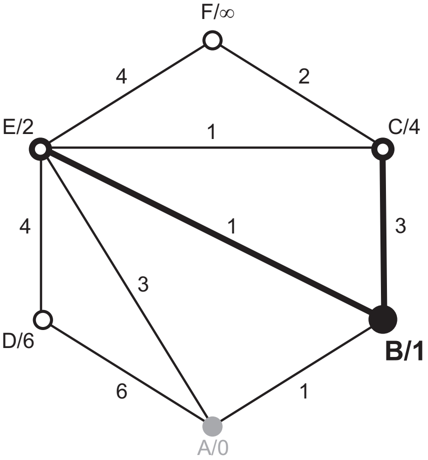
Příklad použití Dijkstrova algoritmu

Uzel B se aktuálně zpracovává, vrchol A je označen jako zpracovaný



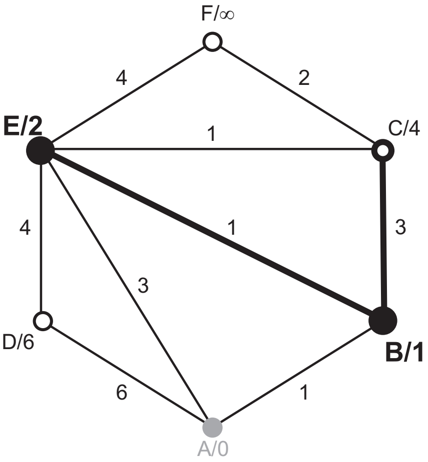
Příklad použití Dijkstrova algoritmu

Mění se návěští uzlů C, E (dosud nezpracovaní sousedé vrcholu B)



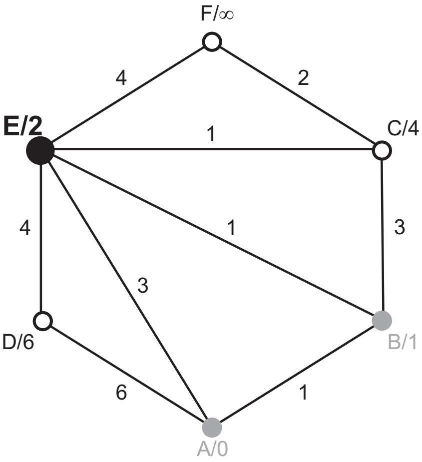
Příklad použití Dijkstrova algoritmu

Z nezpracovaných vrcholů má nejmenší vzdálenost uzel E



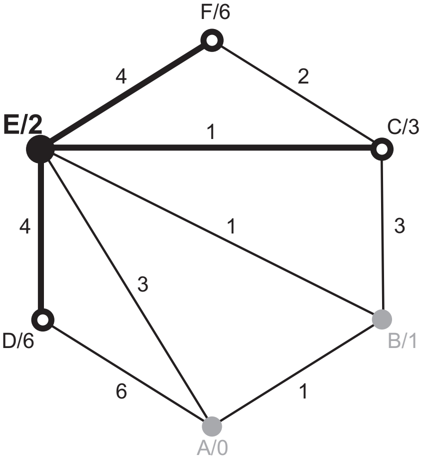
Příklad použití Dijkstrova algoritmu

Uzel E se aktuálně zpracovává, vrchol B je označen jako zpracovaný



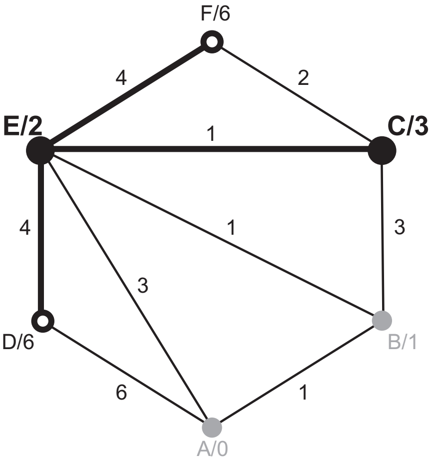
Příklad použití Dijkstrova algoritmu

Mění se návěští uzlů C, D, F (dosud nezpracovaní sousedé vrcholu E)



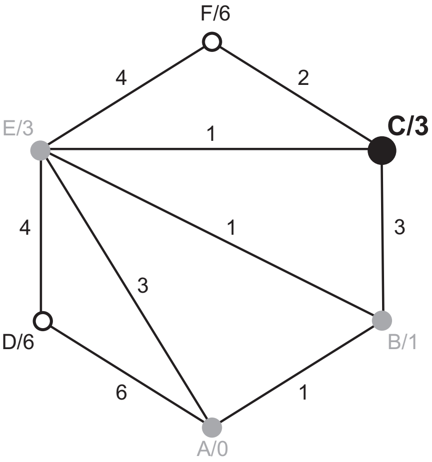
Příklad použití Dijkstrova algoritmu

Z nezpracovaných vrcholů má nejmenší vzdálenost uzel C



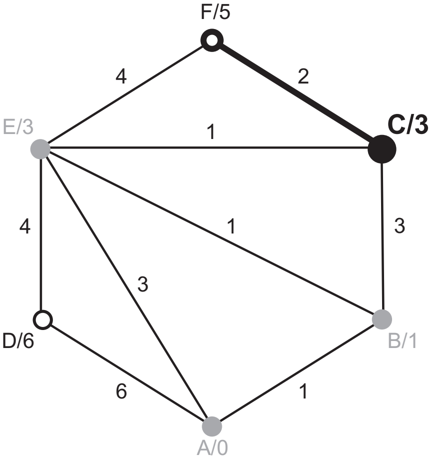
Příklad použití Dijkstrova algoritmu

Uzel C se aktuálně zpracovává, vrchol E je označen jako zpracovaný



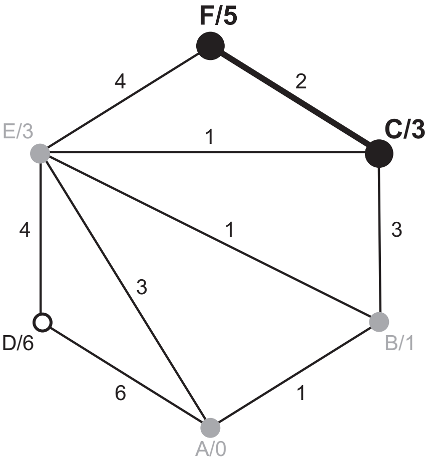
Příklad použití Dijkstrova algoritmu

Mění se návěští uzlu F (dosud nezpracovaný soused vrcholu C)



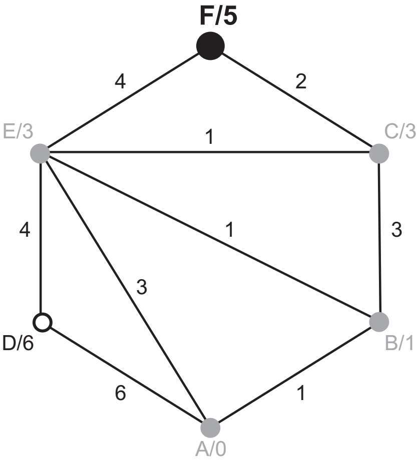
Příklad použití Dijkstrova algoritmu

Z nezpracovaných vrcholů má nejmenší vzdálenost uzel F



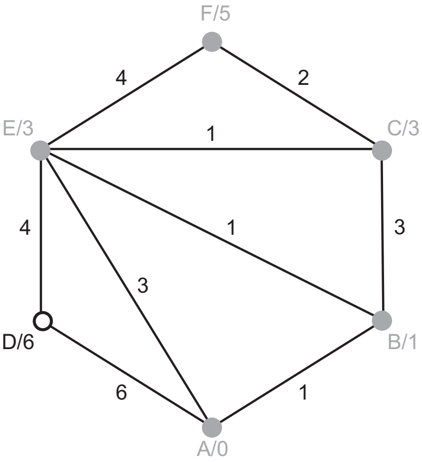
Příklad použití Dijkstrova algoritmu

Uzel F se aktuálně zpracovává, vrchol C je označen jako zpracovaný



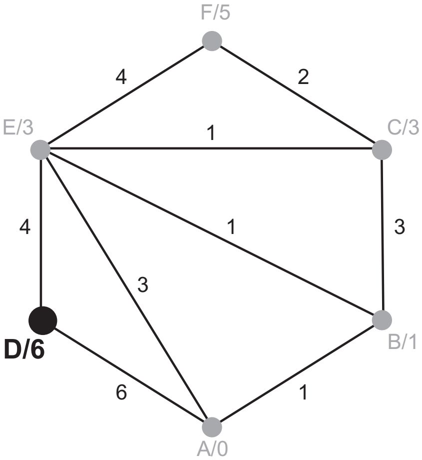
Příklad použití Dijkstrova algoritmu

Uzel F nemá nezpracované sousedy, je označen jako zpracovaný



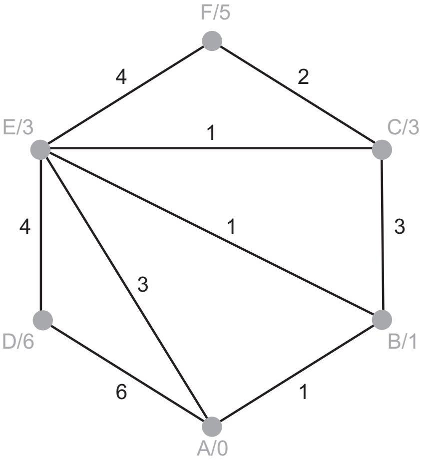
Příklad použití Dijkstrova algoritmu

Z nezpracovaných vrcholů zbývá pouze vrchol D, je označen jako aktuálně zpracovávaný.



Příklad použití Dijkstrova algoritmu

Uzel D nemá nezpracované sousedy. Je označen jako zpracovaný. Konec algoritmu.

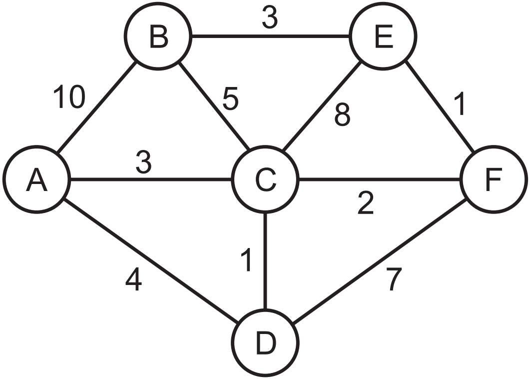


Dijkstrův algoritmus – poznámky

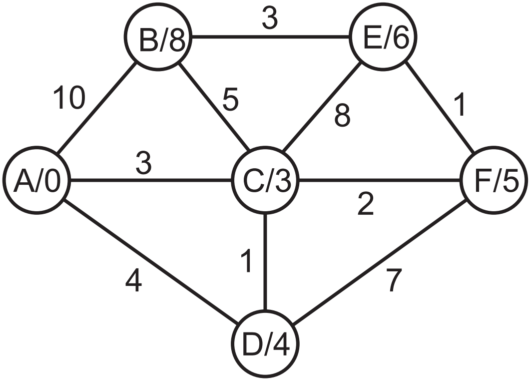
1. Dijkstrův algoritmus lze použít i pro neohodnocené grafy – postačí každé hraně přidat ohodnocení 1.
2. Dijkstrův algoritmus lze použít i pro orientované grafy, tj. grafy, v nichž má hrana směr.
3. Dijkstrův algoritmus nelze spolehlivě použít pro ohodnocené grafy, ve kterých je váha některých hran záporná.

Dijkstrův algoritmus – příklad

Nalezněte (pomocí Dijkstrova algoritmu) nejkratší cestu od uzlu A ke všem ostatním vrcholům grafu zadaného následujícím obrázkem.



Dijkstrův algoritmus – řešení příkladu



Použité zdroje

1. FUCHS, Eduard. *Diskrétní matematika pro učitele.* 1. vyd. Brno: Masarykova univerzita, 2001. 178 s. ISBN 80-210-2703-7.
2. MILKOVÁ, Eva. *Teorie grafů a grafové algoritmy.* 1. vyd. Hradec Králové: Nakladatelství Gaudeamus, 2013. 123 s. ISBN 978-80-7435-267-6.
3. ČERNÝ, Jakub. *Základní grafové algoritmy.* Dostupné z: `http://algoritmy.eu/zga/`
4. PELÁNEK, Radek a kol. *Teorie grafů – Poznámky pro učitele.* Dostupné z: `https://www.fi.muni.cz/ xpelanek/ucitele/?action=logicke`
5. ŠIŠMA, Pavel. *Teorie grafů 1736–1963.* 1. vyd. Praha: Přírodovědecká fakulta Masarykovy univerzity v Brně, 1997. ISBN 80-7196-065-9.
6. MÁSILKO, Lukáš, PECL, Jiří. *Adaptace matematických algoritmů.* Dostupné z: `https://www.teiresias.muni.cz/amalg/www/cs/`