MA2BP_PDM1 Diskrétní matematika 1

9. Prohledávání grafu

Lukáš Másilko

Středisko pro pomoc studentům se specifickými nároky
Masarykova univerzita

6. 12. 2017

Program prezentace

Základní pojmy

Definice 4.1, 4.2, 4.3 (MILKOVÁ):

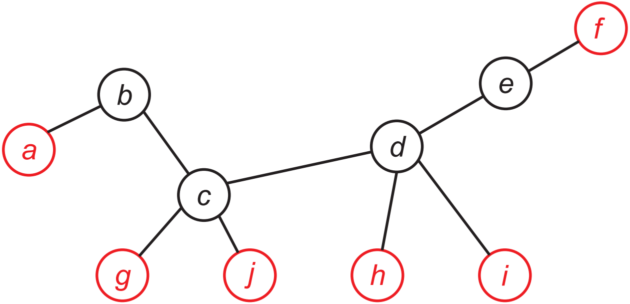
1. **Les** je graf, který neobsahuje kružnici.
2. **Strom** je souvislý graf, který neobsahuje kružnici.
3. Buď $T = (V, E)$ strom. Vrchol $u \in V$ stupně 1 nazýváme **listem**, vrchol $v \in V$ stupně většího než 1 nazývám **vnitřním vrcholem** stromu T.

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 listů

Mějme strom $T = (V, E)$. Pak platí následující tři tvrzení.

- **Tvrzení 4.1** (MILKOVÁ): Je-li T netriviální strom, pak obsahuje list.
- **Věta 4.4** (FUCHS): Je-li T netriviální strom, pak obsahuje alespoň dva listy.
- **Tvrzení 4.2** (MILKOVÁ): Je-li T netriviální strom a odebereme z T list, pak dostaneme opět strom.
- **Tvrzení 4.3** (MILKOVÁ): Pro libovolný vrchol $w \notin V$ platí, že spojíme-li jej hranou s vrcholem $v \in V$, dostaneme opět strom. *(Nově přidaný vrchol w je listem.)*

Příklad:

1. Strom s právě dvěma listy se nazývá **had**.
2. Strom o n vrcholech s právě $n - 1$ listy se nazývá **hvězda**.

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.

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.

Důkaz je poměrně zdlouhavý a opírá se o prokázání ekvivalencí tvrzení $1 \Leftrightarrow 2, 1 \Leftrightarrow 3, 1 \Leftrightarrow 4, 1 \Leftrightarrow 5$. My si uvedeme pouze důkaz $1 \Leftrightarrow 2$.

Věta 4.1, důkaz $1 \Rightarrow 2$

Předpokládáme, že $G = (V, E)$ je strom. Chceme ukázat, že pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y.

Věta 4.1, důkaz $1 \Rightarrow 2$

Předpokládáme, že $G = (V, E)$ je strom. Chceme ukázat, že pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y.

Důkaz: G je strom, tedy souvislý graf bez kružnic. Z definice souvislosti existuje mezi x a y alespoň jedna cesta.

Věta 4.1, důkaz 1 $\Rightarrow$ 2

Předpokládáme, že $G = (V, E)$ je strom. Chceme ukázat, že pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y.

Důkaz: G je strom, tedy souvislý graf bez kružnic. Z definice souvislosti existuje mezi x a y alespoň jedna cesta.

Předpokládejme **sporem**, že mezi vrcholy $x, y \in V$

1. neexistuje žádná cesta, nebo
2. existuje více cest než právě jedna.

Věta 4.1, důkaz 1 $\Rightarrow$ 2

Předpokládáme, že $G = (V, E)$ je strom. Chceme ukázat, že pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y.

Důkaz: G je strom, tedy souvislý graf bez kružnic. Z definice souvislosti existuje mezi x a y alespoň jedna cesta.

Předpokládejme **sporem**, že mezi vrcholy $x, y \in V$

1. neexistuje žádná cesta, nebo
2. existuje více cest než právě jedna.

Ad 1: když mezi x, y neexistuje cesta, není graf G souvislý. **Spor.**

Věta 4.1, důkaz 1 $\Rightarrow$ 2

Předpokládáme, že $G = (V, E)$ je strom. Chceme ukázat, že pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y.

Důkaz: G je strom, tedy souvislý graf bez kružnic. Z definice souvislosti existuje mezi x a y alespoň jedna cesta.

Předpokládejme **sporem**, že mezi vrcholy $x, y \in V$

1. neexistuje žádná cesta, nebo
2. existuje více cest než právě jedna.

Ad 1: když mezi x, y neexistuje cesta, není graf G souvislý. **Spor.**

Ad 2: uvažujme dvě různé cesty C_1, C_2 mezi x, y. Obě začínají v x, v nějakém vrcholu x se "rozcházejí". Poté pokračují odděleně a v nějakém vrcholu y se opět "sejdou". Vytvoří tedy kružnici $(x, \ldots, y, \ldots, x)$. **Spor.**

Věta 4.1, důkaz $2 \Rightarrow 1$

Předpokládáme, že pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y. Chceme ukázat, že G je strom.

Věta 4.1, důkaz 2 $\Rightarrow$ 1

Předpokládáme, že pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y. Chceme ukázat, že G je strom.

Důkaz: Z předpokladu ihned vyplývá, že graf G je souvislý. **Sporem** uvažujme, že obsahuje kružnici.

Věta 4.1, důkaz 2 $\Rightarrow$ 1

Předpokládáme, že pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y. Chceme ukázat, že G je strom.

Důkaz: Z předpokladu ihned vyplývá, že graf G je souvislý. **Sporem** uvažujme, že obsahuje kružnici. Označme
$C = (v_0, e_1, v_1, \ldots, v_{k-1}, e_k, v_0)$
kružnici v G. Všimněte si, že mezi vrcholy v_0, v_1 existují dvě cesty:

Předpokládáme, že pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y. Chceme ukázat, že G je strom.

Důkaz: Z předpokladu ihned vyplývá, že graf G je souvislý. **Sporem** uvažujme, že obsahuje kružnici. Označme
$C = (v_0, e_1, v_1, \dots, v_{k-1}, e_k, v_0)$
kružnici v G. Všimněte si, že mezi vrcholy v_0, v_1 existují dvě cesty:

1. (v_0, e_1, v_1) a
2. $(v_1, \dots, v_{k-1}, e_k, v_0)$.

Věta 4.1, důkaz 2 $\Rightarrow$ 1

Předpokládáme, že pro každé dva vrcholy $x, y \in V$ existuje právě jedna cesta z vrcholu x do vrcholu y. Chceme ukázat, že G je strom.

Důkaz: Z předpokladu ihned vyplývá, že graf G je souvislý. **Sporem** uvažujme, že obsahuje kružnici. Označme
$C = (v_0, e_1, v_1, \ldots, v_{k-1}, e_k, v_0)$
kružnici v G. Všimněte si, že mezi vrcholy v_0, v_1 existují dvě cesty:

1. (v_0, e_1, v_1) a
2. $(v_1, \ldots, v_{k-1}, e_k, v_0)$.

Spor s předpokladem, že mezi každými dvěma vrcholy $x, y \in V$ existuje právě jedna cesta.

Tvrzení 4.4 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Jestliže G je les, který obsahuje k komponent ($k \geq 1$), pak $|V| = |E| + k$.

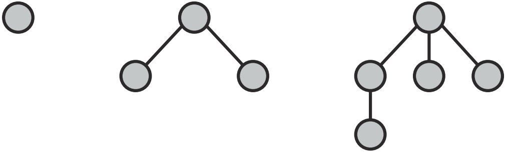
Příklad 1: Je dáno skóre lesa (0, 1, 1, 1, 1, 1, 2, 2, 3). Určete počet jeho komponent a hran. Následně zkuste les nakreslit.

Charakteristika lesa

Tvrzení 4.4 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Jestliže G je les, který obsahuje k komponent $(k \geq 1)$, pak $|V| = |E| + k$.

Příklad 1: Je dáno skóre lesa (0, 1, 1, 1, 1, 1, 2, 2, 3). Určete počet jeho komponent a hran. Následně zkuste les nakreslit.

Řešení: Sečteme všech devět hodnot ve skóre a získáme číslo 12. Platí: $|V| = 9, |E| = 6$, z čehož můžeme usoudit, že graf má 3 komponenty. Počáteční číslo 0 znamená, že jedna komponenta je izolovaný vrchol. Zbylé komponenty intuitivně nakreslíme takto:



Kořenový strom

Definice 4.4 (MILKOVÁ):
Dvojici (T, r), kde $T = (V, E)$ je strom a $r \in V$ je pevně zvolený vrchol, nazýváme **kořenový strom** a vrchol r **kořen** stromu (T, r).

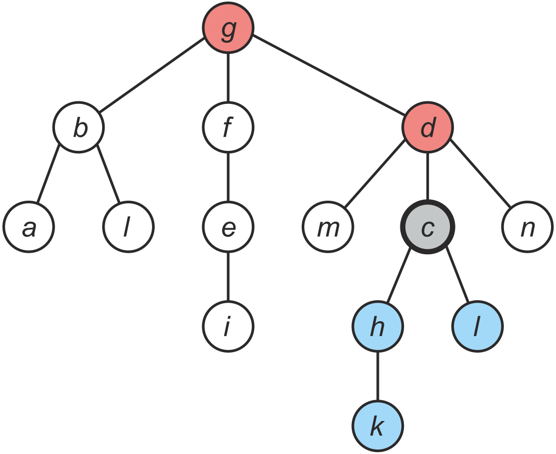
Další pojmy (viz Definice 4.5, MILKOVÁ):

1. Jestliže vrchol x leží na cestě z kořene r do vrcholu y, říkáme, že
   x je předchůdce y,
   y je následník x.
2. Jsou-li navíc x, y sousední vrcholy, pak
   x je přímý předchůdce (rodič, otec) y,
   y je přímý následník (syn) x.

Příklad 2.4

Na obrázku je kořenový strom (T, g), zvýrazněný vrchol c a jeho

1. předchůdci – uzly g a d (přímý předchůdce)
2. následníci – uzly h, l (oba přímí následníci) a k



Hloubka kořenového stromu

Definice 4.6 (MILKOVÁ):
Nechť (T, r) je kořenový strom, v jeho libovolný vrchol a k délka cesty z kořene r do vrcholu v. Pak říkáme, že

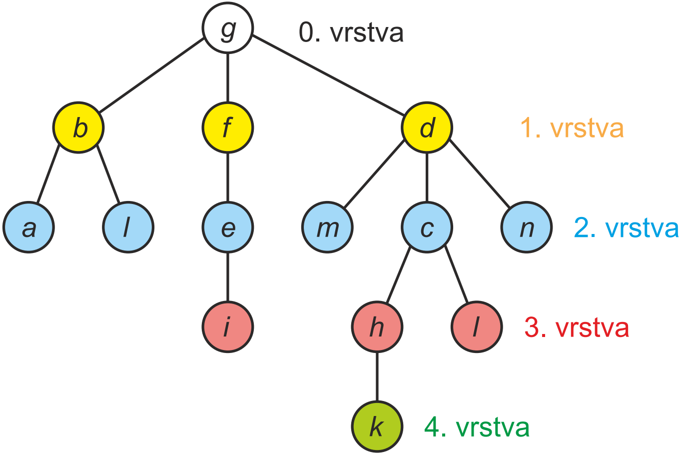
1. vrchol v leží v k-té **vrstvě** stromu (T, r) nebo též ve vrstvě k.
2. **Hloubka** stromu (T, r) je rovna největšímu číslu z čísel vrstev, ve kterých leží listy stromu (T, r).

Poznámka: Kořen r leží ve vrstvě 0, jeho přímí následníci (synové) v 1. vrstvě.

Definice 4.7 (MILKOVÁ):
Nechť (T, r) je kořenový strom a (T', v) je podgraf (T, r) *indukovaný* vrcholem v a všemi jeho následníky. Podgraf (T', v) nazýváme **podstromem** stromu (T, r) s kořenem v.

Příklad 2.4

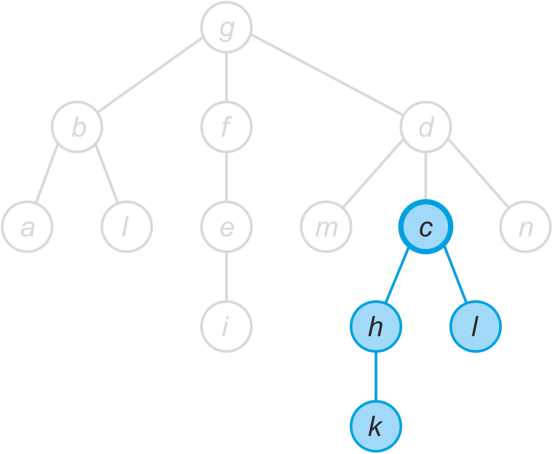
Na obrázku je kořenový strom (T, g). Barevně jsou vyznačeny vrstvy.



Hloubka stromu je 4.

Příklad 2.4

Na obrázku je podstrom (T', c) kořenového stromu (T, g).

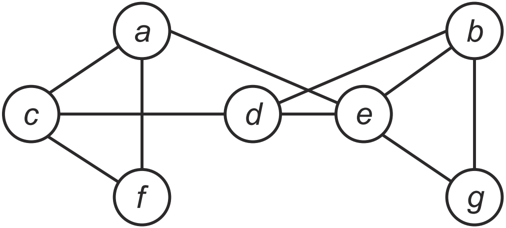


Prohledávání grafů

- **Motivace:** potřebujeme systematicky prozkoumat všechny vrcholy zadaného grafu.
- Existují dva algoritmy, u nichž je třeba stanovit počátek prohledávání (vrchol, odkud začneme):
  1. **Prohledávání do hloubky** – založeno na principu LIFO (Last In, First Out) – vrcholy prohledáváme *po podstromech dolů*
  2. **Prohledávání do šířky** – založeno na principu FIFO (First In, First Out) – vrcholy prohledáváme *po vrstvách*

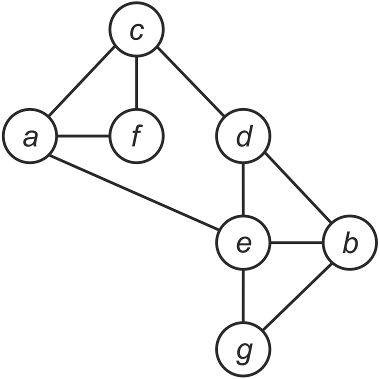
Příklad 7.1 – prohledávání do šířky

Je dán následující graf, který chceme prohledat do do šířky, přičemž začínáme ve vrcholu c.



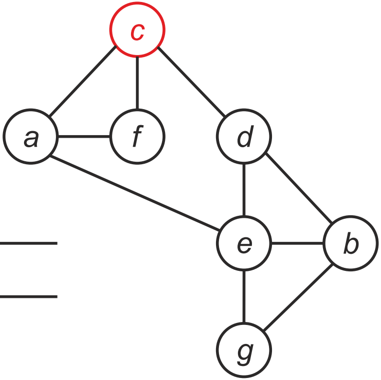
Příklad 7.1 – prohledávání do šířky (řešení)

Nejprve si graf překreslíme do uspořádání odpovídající kořenovému stromu, tj. počátek prohledávání bude nahoře (ve vrstvě 0), ostatní vrcholy pak rozmístěny po vrstvách podle toho, jaká je jejich vzdálenost od vrcholu c.



Příklad 7.1 – prohledávání do šířky (řešení)

Začínáme vrcholem c, který vložíme do fronty. Prozkoumáme přímé následníky vrcholu c.

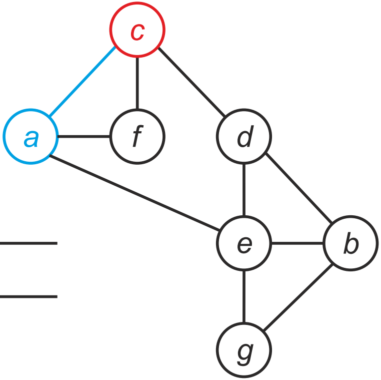


Fronta:

c

Příklad 7.1 – prohledávání do šířky (řešení)

Dle lexikografického pravidla vybereme nejdříve a a vložíme jej na konec fronty, přičemž označíme modře hranu $\{c, a\}$.

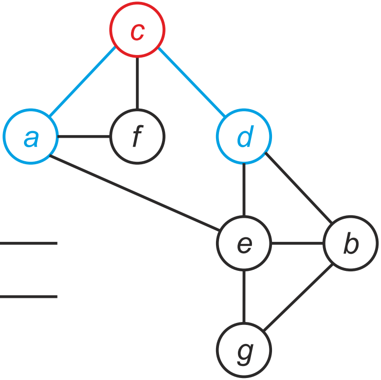


Fronta:

$a\ c$

Příklad 7.1 – prohledávání do šířky (řešení)

Dle lexikografického pravidla vybereme dále d a vložíme jej na konec fronty, přičemž označíme modře hranu $\{c, d\}$.

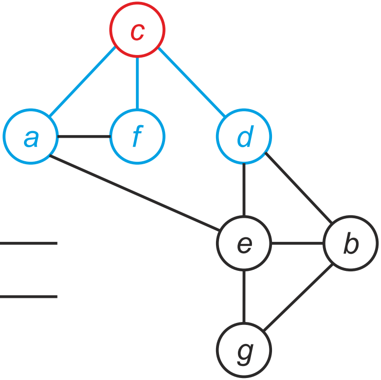


Fronta:

d a c

Příklad 7.1 – prohledávání do šířky (řešení)

Dle lexikografického pravidla vybereme nakonec f a vložíme jej na konec fronty, přičemž označíme modře hranu $\{c, f\}$.

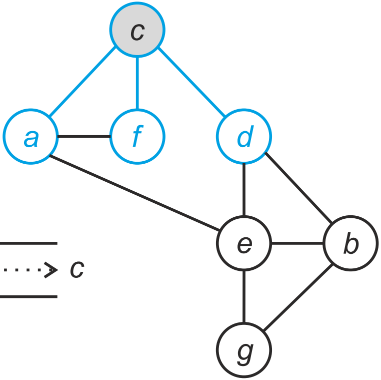


Fronta:

$f\ d\ a\ c$

Příklad 7.1 – prohledávání do šířky (řešení)

Vrchol c nemá další přímé následníky. Označíme jej jako vyřešený a odebereme ze začátku fronty.



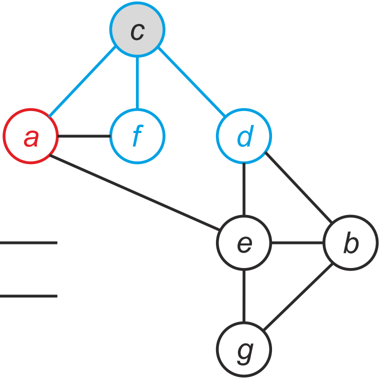
Fronta:

$f\ d\ a \dashrightarrow c$

Pořadí:

Příklad 7.1 – prohledávání do šířky (řešení)

Na začátku fronty je vrchol a, vybereme jej ke zpracování.



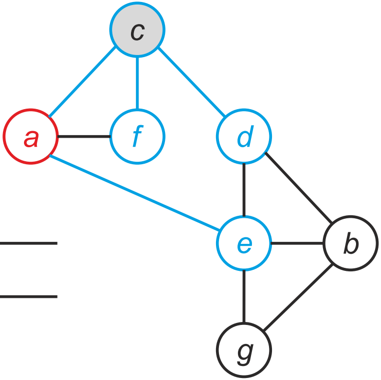
Fronta:

$f\ d\ a$

Pořadí: c

Příklad 7.1 – prohledávání do šířky (řešení)

Dle lexikografického pravidla vybereme nejdříve e a vložíme jej na konec fronty, přičemž označíme modře hranu $\{a, e\}$.



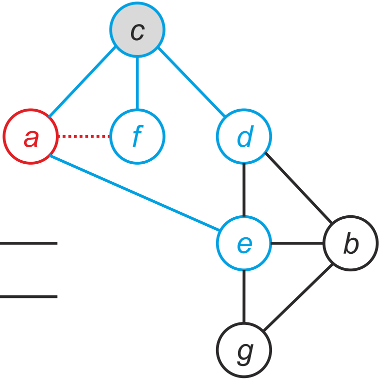
Fronta:

$e\ f\ d\ a$

Pořadí: c

Příklad 7.1 – prohledávání do šířky (řešení)

Dle lexikografického pravidla poté vybereme f – je však již obarvený (navštívený), proto pouze označíme červeně hranu $\{a, f\}$.



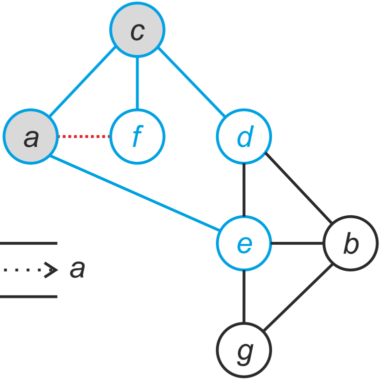
Fronta:

e f d a

Pořadí: *c*

Příklad 7.1 – prohledávání do šířky (řešení)

Vrchol *a* nemá další přímé následníky. Označíme jej jako vyřešený a odebereme ze začátku fronty.



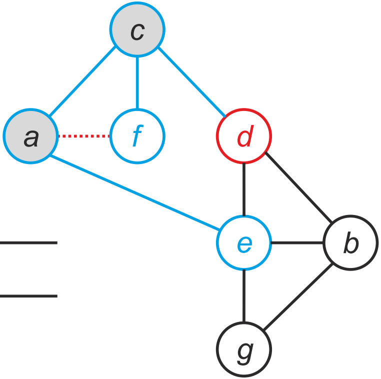
Fronta:

e f d ····> *a*

Pořadí: *c*

Příklad 7.1 – prohledávání do šířky (řešení)

Na začátku fronty je vrchol d, vybereme jej ke zpracování.



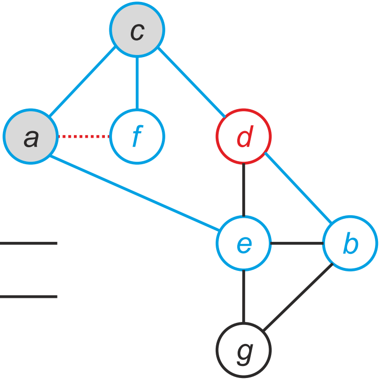
Fronta:

$e\ f\ d$

Pořadí: c, a

Příklad 7.1 – prohledávání do šířky (řešení)

Dle lexikografického pravidla vybereme nejdříve b a vložíme jej na konec fronty, přičemž označíme modře hranu $\{d, b\}$.



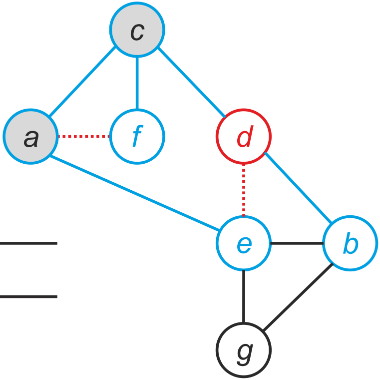
Fronta:

b e f d

Pořadí: *c*, *a*

Příklad 7.1 – prohledávání do šířky (řešení)

Dle lexikografického pravidla poté vybereme e – je však již obarvený (navštívený), proto pouze označíme červeně hranu $\{d, e\}$.



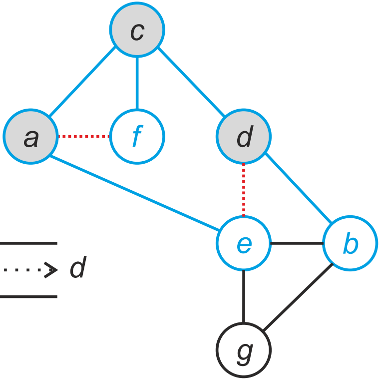
Fronta:

b e f d

Pořadí: *c*, *a*

Příklad 7.1 – prohledávání do šířky (řešení)

Vrchol d nemá další přímé následníky. Označíme jej jako vyřešený a odebereme ze začátku fronty.



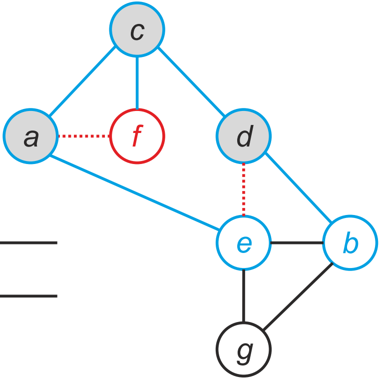
Fronta:

$b\ e\ f \dashrightarrow d$

Pořadí: c, a

Příklad 7.1 – prohledávání do šířky (řešení)

Na začátku fronty je vrchol f, vybereme jej ke zpracování.



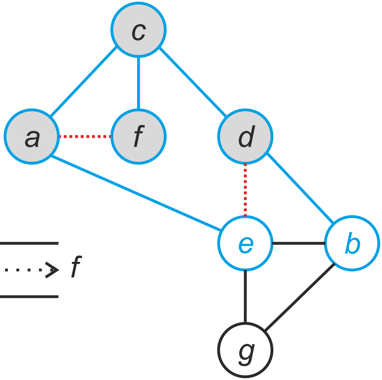
Fronta:

b e f

Pořadí: c, a, d

Příklad 7.1 – prohledávání do šířky (řešení)

Z vrcholu f již nevede žádná neobarvená hrana. Označíme jej jako vyřešený a odebereme ze začátku fronty.



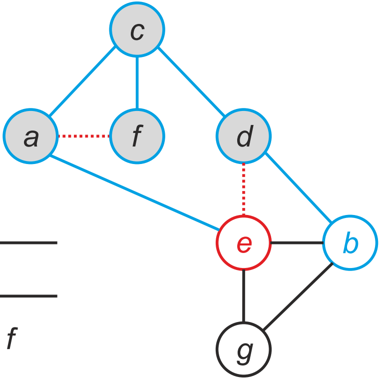
Fronta:

b e $\cdots\to$ f

Pořadí: c, a, d

Příklad 7.1 – prohledávání do šířky (řešení)

Na začátku fronty je vrchol e, vybereme jej ke zpracování.



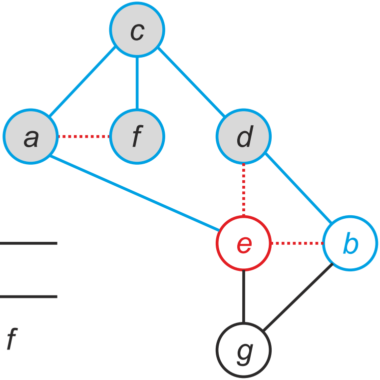
Fronta:

$b\ e$

Pořadí: c, a, d, f

Příklad 7.1 – prohledávání do šířky (řešení)

Dle lexikografického pravidla vybereme nejdříve b – je však již obarvený (navštívený), proto pouze označíme červeně hranu $\{e, b\}$.



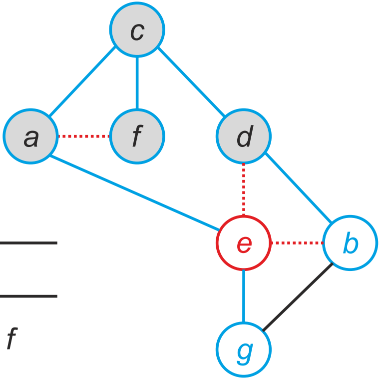
Fronta:

b e

Pořadí: *c*, *a*, *d*, *f*

Příklad 7.1 – prohledávání do šířky (řešení)

Dle lexikografického pravidla vybereme dále g a vložíme jej na konec fronty, přičemž označíme modře hranu $\{e, g\}$.



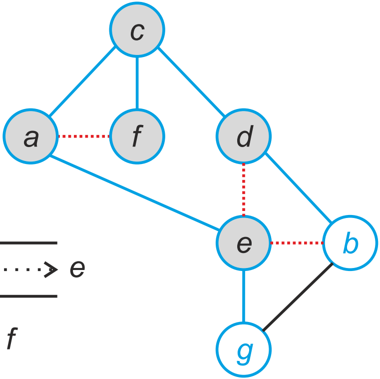
Fronta:

g b e

Pořadí: *c*, *a*, *d*, *f*

Příklad 7.1 – prohledávání do šířky (řešení)

Vrchol e nemá další přímé následníky. Označíme jej jako vyřešený a odebereme ze začátku fronty.



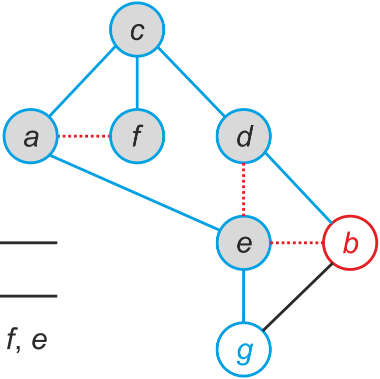
Fronta:

$g\ b \quad \cdots\!\!> e$

Pořadí: c, a, d, f

Příklad 7.1 – prohledávání do šířky (řešení)

Na začátku fronty je vrchol b, vybereme jej ke zpracování.



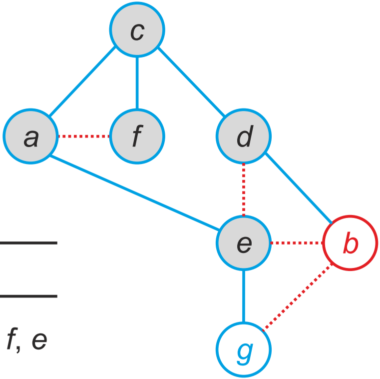
Fronta:

g b

Pořadí: c, a, d, f, e

Příklad 7.1 – prohledávání do šířky (řešení)

Dle lexikografického pravidla vybereme nejdříve g – je však již obarvený (navštívený), proto pouze označíme červeně hranu $\{b, g\}$.



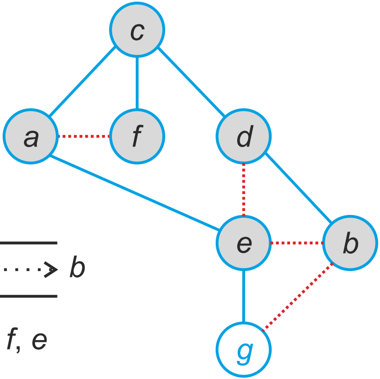
Fronta:

g b

Pořadí: *c*, *a*, *d*, *f*, *e*

Příklad 7.1 – prohledávání do šířky (řešení)

Vrchol b nemá další přímé následníky. Označíme jej jako vyřešený a odebereme ze začátku fronty.



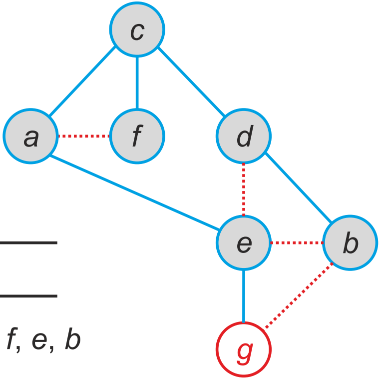
Fronta:

g $\cdots\!\!>$ b

Pořadí: c, a, d, f, e

Příklad 7.1 – prohledávání do šířky (řešení)

Na začátku fronty je vrchol g, vybereme jej ke zpracování.



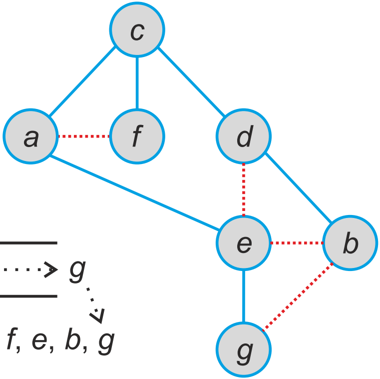
Fronta:

g

Pořadí: *c*, *a*, *d*, *f*, *e*, *b*

Příklad 7.1 – prohledávání do šířky (řešení)

Z vrcholu g již nevede žádná neobarvená hrana. Označíme jej jako vyřešený a odebereme ze začátku fronty.



Fronta:

g

Pořadí: c, a, d, f, e, b, g

Prohledávání do šířky - shrnutí

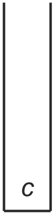
Definice 7.1 (MILKOVÁ, mírně pozměněno): Při prohledávání souvislého grafu $G = (V, E)$ konstruujeme modrý **strom prohledávání do šířky** (T_s, v) s kořenem $v \in V$ jako počátkem prohledávání. Hrany, které nepatří do (T_s, v), označujeme červenou barvou a nazýváme **nestromové hrany**. Pro libovolný vrchol $w \in V$ je zavedeno nezáporné číslo $h(w)$ udávající číslo vrstvy stromu (T_s, v), ve které leží vrchol w.

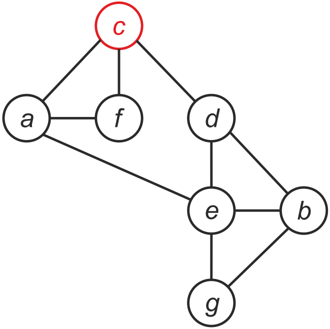
Poznámka:

- Aktuální vrchol je na začátku fronty. Označíme jej za zpracovaný a odebereme z fronty, až projdeme veškeré neobarvené hrany s ním incidentní.
- Koncové vrcholy neobarvených hran vkládáme na konec fronty pouze tehdy, když nebyly dříve navštíveny. V takovém případě značíme hranu modře. Pokud neobarvená hrana vede do navštíveného vrcholu, tak pouze obarvíme hranu červeně.
- Algoritmus končí vyprázdněním fronty.

Příklad 7.1 – prohledávání do hloubky (řešení)

Začínáme vrcholem c, který vložíme do zásobníku. Prozkoumáme přímé následníky vrcholu c.

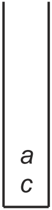




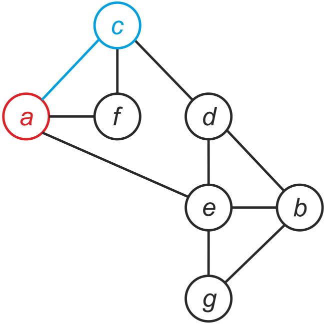
Zásobník:
Pořadí: c

Příklad 7.1 – prohledávání do hloubky (řešení)

Dle lexikografického pravidla vybereme a, vložíme jej na zásobník a označíme jako aktuální, přičemž obarvíme modře hranu $\{c, a\}$.

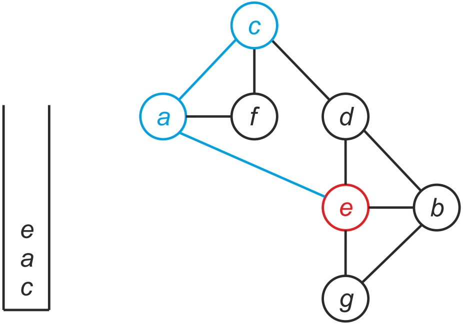


Zásobník:
Pořadí: c, a



Příklad 7.1 – prohledávání do hloubky (řešení)

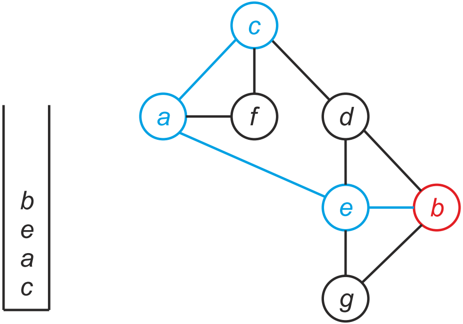
Dle lexikografického pravidla vybereme e, vložíme jej na zásobník a označíme jako aktuální, přičemž obarvíme modře hranu $\{a, e\}$.



Zásobník:
Pořadí: c, a, e

Příklad 7.1 – prohledávání do hloubky (řešení)

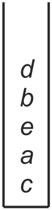
Dle lexikografického pravidla vybereme b, vložíme jej na zásobník a označíme jako aktuální, přičemž obarvíme modře hranu $\{e, b\}$.

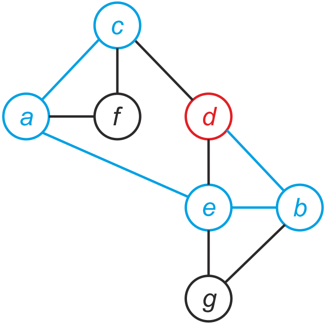


Zásobník:
Pořadí: c, a, e, b

Příklad 7.1 – prohledávání do hloubky (řešení)

Dle lexikografického pravidla vybereme d, vložíme jej na zásobník a označíme jako aktuální, přičemž obarvíme modře hranu $\{b, d\}$.

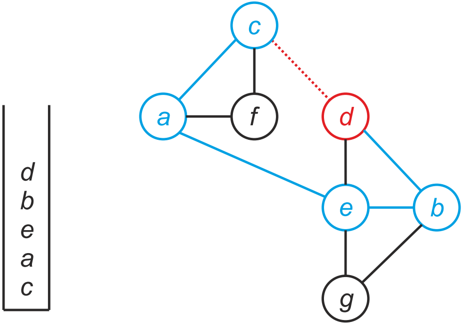




Zásobník:
Pořadí: c, a, e, b, d

Příklad 7.1 – prohledávání do hloubky (řešení)

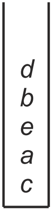
Dle lexikografického pravidla vybereme c, který však už byl navštívený. Pouze obarvíme červeně hranu $\{d, c\}$.

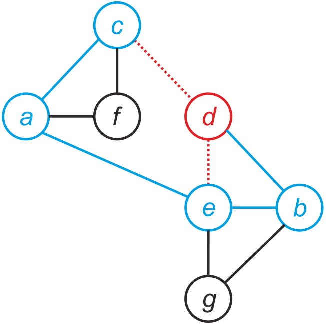


Zásobník:
Pořadí: c, a, e, b, d

Příklad 7.1 – prohledávání do hloubky (řešení)

Dle lexikografického pravidla vybereme e, který však už byl navštívený. Pouze obarvíme červeně hranu $\{d, e\}$.



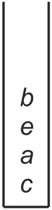


Zásobník:

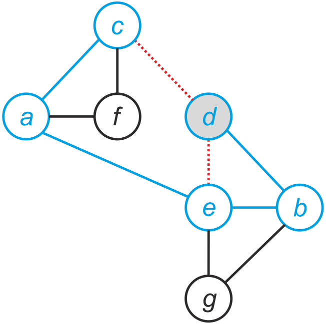
Pořadí: c, a, e, b, d

Příklad 7.1 – prohledávání do hloubky (řešení)

Z vrcholu d již nevede žádná neobarvená hrana. Označíme jej jako vyřešený a odebereme ze zásobníku.



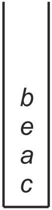
Zásobník:

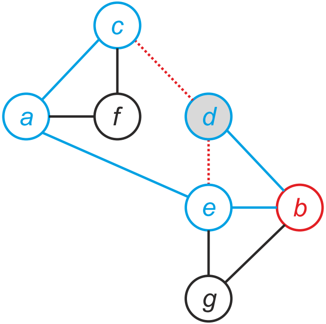


Pořadí: c, a, e, b, d

Příklad 7.1 – prohledávání do hloubky (řešení)

Na vrcholu zásobníku je *b*. Označíme jej jako aktuální a hledáme přímé následníky.

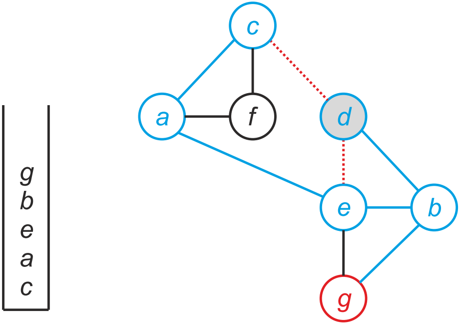




Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*

Příklad 7.1 – prohledávání do hloubky (řešení)

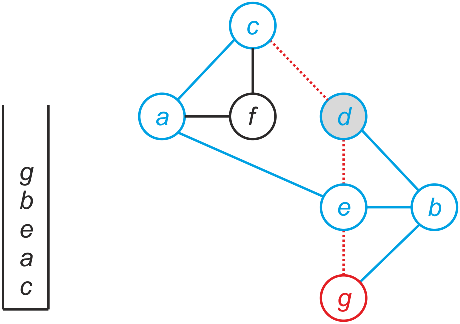
Z vrcholu b vede jediná neobarvená hrana do nenavštíveného uzlu g. Vložíme g na zásobník a označíme jako aktuální, přičemž obarvíme modře hranu $\{b, g\}$.



Zásobník:
Pořadí: c, a, e, b, d, g

Příklad 7.1 – prohledávání do hloubky (řešení)

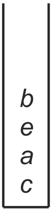
Z vrcholu g vede jediná neobarvená hrana do navštíveného uzlu e. Pouze obarvíme červeně hranu $\{g, e\}$.

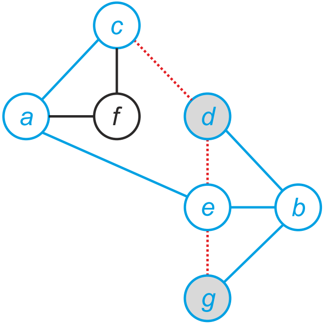


Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*

Příklad 7.1 – prohledávání do hloubky (řešení)

Z vrcholu g již nevede žádná neobarvená hrana. Označíme jej jako vyřešený a odebereme ze zásobníku.

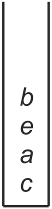


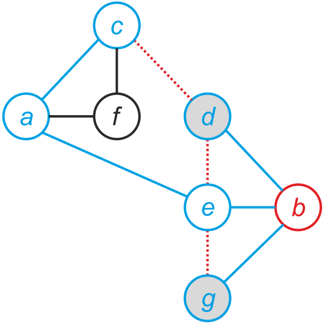


Zásobník:
Pořadí: c, a, e, b, d, g

Příklad 7.1 – prohledávání do hloubky (řešení)

Na vrcholu zásobníku je *b*. Označíme jej jako aktuální a hledáme přímé následníky.



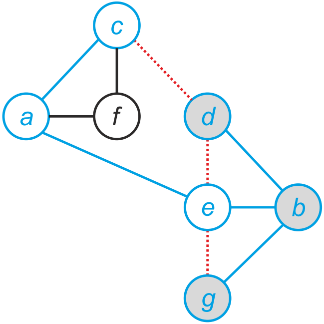


Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*

Příklad 7.1 – prohledávání do hloubky (řešení)

Z vrcholu *b* již nevede žádná neobarvená hrana. Označíme jej jako vyřešený a odebereme ze zásobníku.

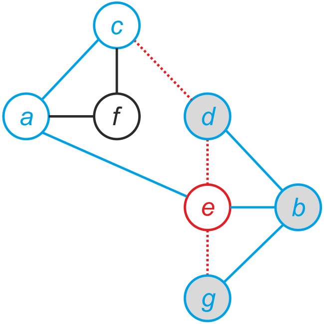
e
a
c



Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*

Příklad 7.1 – prohledávání do hloubky (řešení)

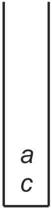
Na vrcholu zásobníku je e. Označíme jej jako aktuální a hledáme přímé následníky.

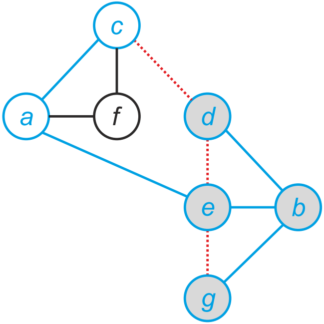


Zásobník:
Pořadí: c, a, e, b, d, g

Příklad 7.1 – prohledávání do hloubky (řešení)

Z vrcholu *e* již nevede žádná neobarvená hrana. Označíme jej jako vyřešený a odebereme ze zásobníku.

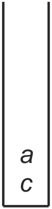


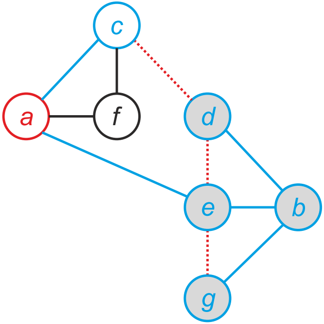


Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*

Příklad 7.1 – prohledávání do hloubky (řešení)

Na vrcholu zásobníku je *a*. Označíme jej jako aktuální a hledáme přímé následníky.

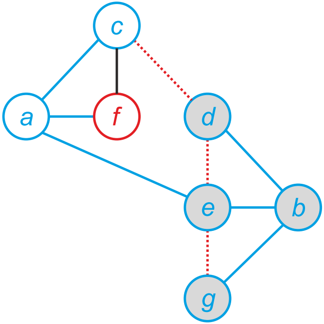




Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*

Příklad 7.1 – prohledávání do hloubky (řešení)

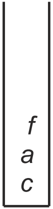
Z vrcholu a vede jediná neobarvená hrana do nenavštíveného uzlu f. Vložíme f na zásobník a označíme jako aktuální, přičemž obarvíme modře hranu $\{a, f\}$.

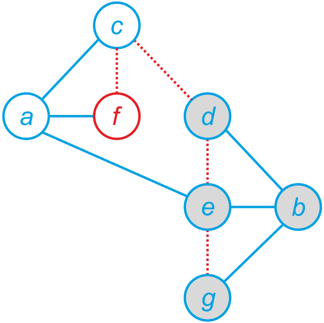


Zásobník:
Pořadí: c, a, e, b, d, g, f

Příklad 7.1 – prohledávání do hloubky (řešení)

Z vrcholu f vede jediná neobarvená hrana do navštíveného uzlu c. Pouze obarvíme červeně hranu $\{f, c\}$.

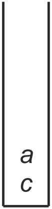


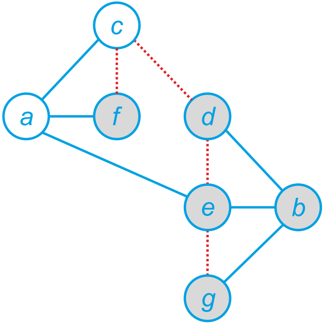


Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*, *f*

Příklad 7.1 – prohledávání do hloubky (řešení)

Z vrcholu *f* již nevede žádná neobarvená hrana. Označíme jej jako vyřešený a odebereme ze zásobníku.

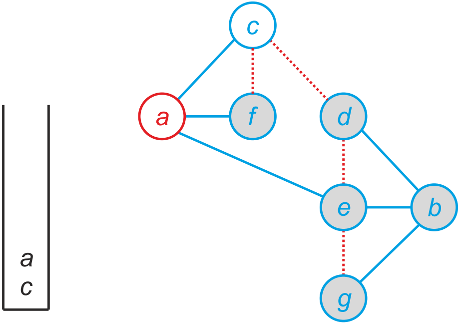




Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*, *f*

Příklad 7.1 – prohledávání do hloubky (řešení)

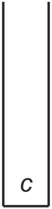
Na vrcholu zásobníku je *a*. Označíme jej jako aktuální a hledáme přímé následníky.

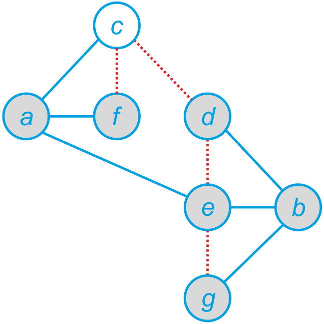


Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*, *f*

Příklad 7.1 – prohledávání do hloubky (řešení)

Z vrcholu *a* již nevede žádná neobarvená hrana. Označíme jej jako vyřešený a odebereme ze zásobníku.

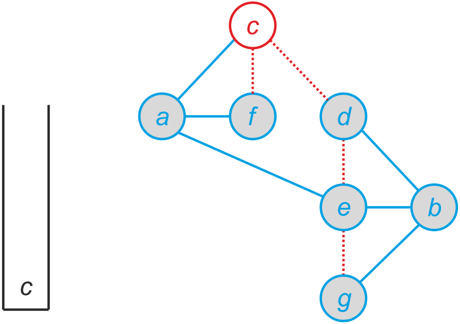




Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*, *f*

Příklad 7.1 – prohledávání do hloubky (řešení)

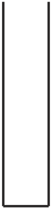
Na vrcholu zásobníku je *c*. Označíme jej jako aktuální a hledáme přímé následníky.

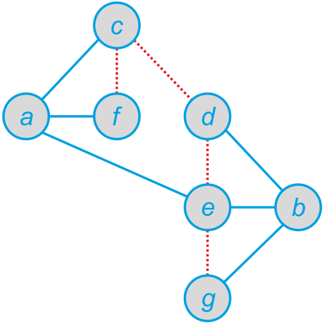


Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*, *f*

Příklad 7.1 – prohledávání do hloubky (řešení)

Z vrcholu *c* již nevede žádná neobarvená hrana. Označíme jej jako vyřešený a odebereme ze zásobníku.





Zásobník:
Pořadí: *c*, *a*, *e*, *b*, *d*, *g*, *f*

Prohledávání do hloubky - shrnutí

Definice 7.2 (MILKOVÁ, mírně pozměněno): Při prohledávání souvislého grafu $G = (V, E)$ konstruujeme modrý **strom prohledávání do hloubky** (T_h, v) s kořenem $v \in V$ jako počátkem prohledávání. Hrany, které nepatří do (T_s, v), označujeme červenou barvou a nazýváme **nestromové hrany**.

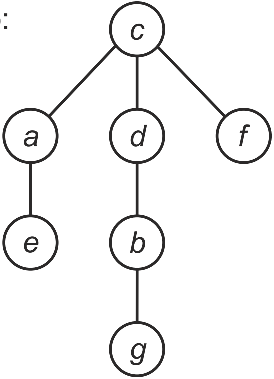
Poznámka:

- Aktuální vrchol je vždy na vrcholu zásobníku. Označíme jej za zpracovaný a odebereme ze zásobníku, až projdeme veškeré neobarvené hrany s ním incidentní.
- Koncové vrcholy neobarvených hran vkládáme na zásobník pouze tehdy, když nebyly dříve navštíveny. V takovém případě značíme hranu modře. Pokud neobarvená hrana vede do navštíveného vrcholu, tak pouze obarvíme hranu červeně.
- Algoritmus končí vyprázdněním zásobníku.

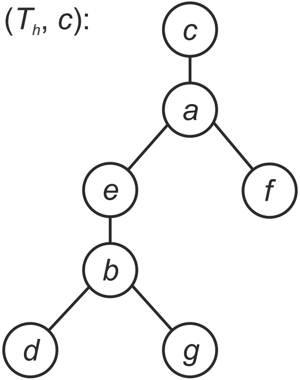
Strompy prohledávání z Příkladu 7.1

- Pořadí vrcholů při procházení do šířky: c, a, d, f, e, b, g
- Pořadí vrcholů při procházení do hloubky: c, a, e, b, d, g, f

(T_s, c):

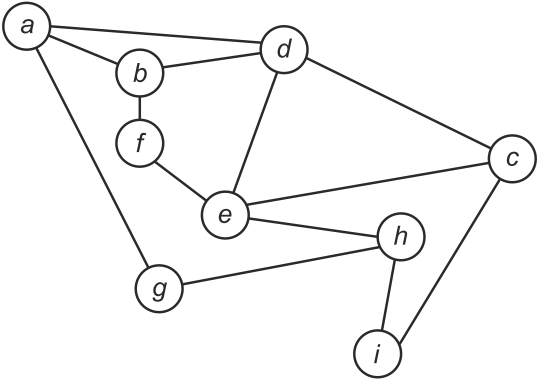


(T_h, c):



Příklad 2

Prohledejte následující graf do šířky a do hloubky, přičemž začněte v obou případech z vrcholu *a*. Na závěr nakreslete oba stromy prohledávání a zapište pořadí, ve kterém jste navštívili uzly.



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.