MA2BP_PDM1 Diskrétní matematika 1

7. Eulerovské grafy

Lukáš Másilko

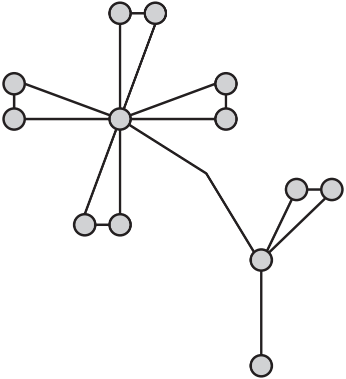
Středisko pro pomoc studentům se specifickými nároky
Masarykova univerzita

28. 11. 2017

Program prezentace

Úvodní příklad

Nakreslete jedním tahem květinu zobrazenou na následujícím grafu.



Úkol spočívající v nakreslení eulerovského tahu.

Eulerovský tah a graf

Definice 2.3 (MILKOVÁ):

- **Eulerovský tah** v grafu G je uzavřený nebo otevřený tah, který obsahuje všechny hrany grafu G.
- **Eulerovský graf** je souvislý graf G, ve kterém existuje eulerovský tah.

Historická poznámka: Název se vztahuje ke slavnému švýcarskému matematikovi Leonardovi Eulerovi (1707–1783), který v r. 1736 řešil

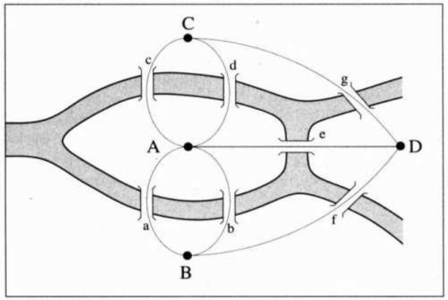
Problém sedmi mostů města Královce

(původně Königsberg, dnes Kaliningrad na Kaliningradském území Ruska mezi Polskem a Litvou, do r. 1945 hlavní město Východního Pruska).

Jde o první příspěvek k teorii grafů a rok 1736 je pokládán za počátek teorie grafů.

Problém sedmi mostů města Královce

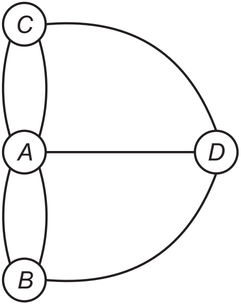
Ve městě Královec jsou v centru města na řece Pregel (dnes Pregolya) dva ostrovy, které v 18. století spojovalo s oběma břehy sedm mostů:



Obyvatelé města při procházkách často přemýšleli, zda by bylo možné projít se městem tak, aby procházku začali v jednom místě, přešli všechny mosty, přes každý právě jednou, a vrátili se tam, kde začali.

Problém sedmi mostů města Královce

Převedeno do teorie grafů: existuje v následujícím grafu uzavřený eulerovský tah?



Leonard Euler ukázal, že to není možné, a dokonce formuloval obecnou charakteristiku úloh daného typu.

Nutná a postačující podmínka pro Eulerovský graf

Věta 2.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je eulerovský *právě tehdy, když* G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně.

Nutná a postačující podmínka pro Eulerovský graf

Věta 2.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je eulerovský *právě tehdy, když* G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně.

Důkaz (nebude vyžadován u zkoušky): jedná se tvrzení tvaru logické ekvivalence, je tedy třeba ukázat oba implikační směry:

1. "$\Rightarrow$:" Graf G je eulerovský $\Rightarrow$ G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně.
2. "$\Leftarrow$:" Graf G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně $\Rightarrow$ G je eulerovský.

Nutná a postačující podmínka pro Eulerovský graf

Věta 2.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je eulerovský *právě tehdy, když* G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně.

Důkaz (nebude vyžadován u zkoušky): jedná se tvrzení tvaru logické ekvivalence, je tedy třeba ukázat oba implikační směry:

1. "$\Rightarrow$:" Graf G je eulerovský $\Rightarrow$ G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně.
2. "$\Leftarrow$:" Graf G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně $\Rightarrow$ G je eulerovský.

Směr "$\Rightarrow$" si nyní ukážeme. Pro směr "$\Leftarrow$" budeme potřebovat několik pomocných tvrzení.

Důkaz Věty 2.2 ve směru “$\Rightarrow$”

Předpokládejme, že je graf G eulerovský. Z definice 2.1 tedy je souvislý a existuje v něm (uzavřený či otevřený) eulerovský tah T obsahující všechny hrany grafu G.

1

2

Důkaz Věty 2.2 ve směru "$\Rightarrow$"

Předpokládejme, že je graf G eulerovský. Z definice 2.1 tedy je souvislý a existuje v něm (uzavřený či otevřený) eulerovský tah T obsahující všechny hrany grafu G.

1. Je-li T **uzavřený**, pak každý vrchol G, kterým tah minimálně jednou prochází, je sudého stupně. Pro každý $v \in V$ totiž platí $\deg_G(v) = 2p$, kde p počet výskytů vrcholu v v tahu T.
2.

Důkaz Věty 2.2 ve směru “$\Rightarrow$”

Předpokládejme, že je graf G eulerovský. Z definice 2.1 tedy je souvislý a existuje v něm (uzavřený či otevřený) eulerovský tah T obsahující všechny hrany grafu G.

1. Je-li T **uzavřený**, pak každý vrchol G, kterým tah minimálně jednou prochází, je sudého stupně. Pro každý $v \in V$ totiž platí $\deg_G(v) = 2p$, kde p počet výskytů vrcholu v v tahu T.
2. Je-li T **otevřený**, pak počáteční a koncový vrchol tahu T mají lichý stupeň. Ostatní *vnitřní* vrcholy tahu T mají sudý stupeň.

Připomenutí a první pomocné tvrzení

Pro připomenutí:

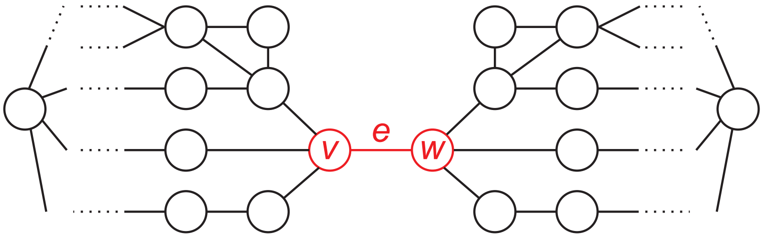
- **Důsledek 1.1** (Milková): Počet vrcholů lichého stupně v grafu G je sudý.
- **Důsledek 1.2** (Milková): Hrana grafu je buď most nebo leží na nějaké kružnici grafu.
- **Definice:** Graf $G = (V, E)$ nazveme **sudým grafem**, jsou-li všechny vrcholy $v \in V$ sudého stupně.

Tvrzení 2.1 (Milková): Pro každý graf $G = (V, E)$ platí:
Jestliže je graf G sudý, pak G neobsahuje most.

Důkaz: provedeme sporem, tj. předpokládejme, že existuje sudý graf $G = (V, E)$, který obsahuje most: hranu $e = \{v, w\}$, viz následující slajd.

Důkaz Tvrzení 2.1

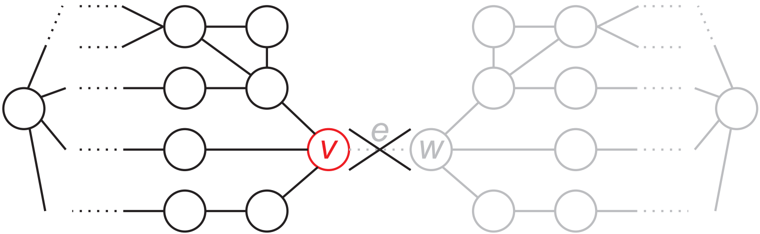
Graf G s mostem $e = \{v, w\}$:



Z grafu odebereme hranu e a podíváme se na komponentu Q_1 obsahující vrchol v.

Důkaz Tvrzení 2.1

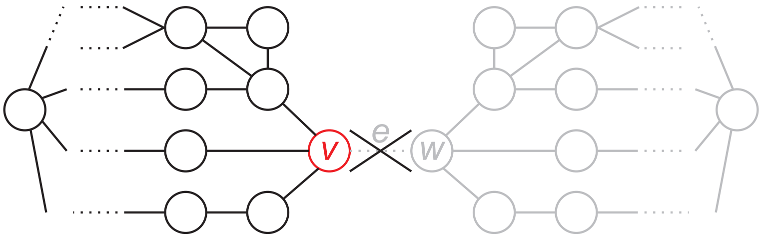
Komponenta Q_1 s vrcholem v:



Protože graf G byl sudý, jsou všechny vrcholy komponenty Q_1 sudého stupně s výjimkou vrcholu v, u něhož se po odebrání hrany e změnil stupeň ze sudého na lichý.

Důkaz Tvrzení 2.1

Komponenta Q_1 s vrcholem v:



Protože graf G byl sudý, jsou všechny vrcholy komponenty Q_1 sudého stupně s výjimkou vrcholu v, u něhož se po odebrání hrany e změnil stupeň ze sudého na lichý.

Spor s Důsledkem 1.1 (počet vrcholů lichého stupně grafu musí být sudý) $\Rightarrow$ platí Tvrzení 2.1: *Sudý graf neobsahuje most.*

Tvrzení 2.2 (Milková): Pro každý graf $G = (V, E)$ platí:
Jestliže graf G je sudý, pak každá hrana grafu G leží na nějaké kružnici grafu G.

Důkaz:

Druhé pomocné tvrzení

Tvrzení 2.2 (Milková): Pro každý graf $G = (V, E)$ platí:
Jestliže graf G je sudý, pak každá hrana grafu G leží na nějaké kružnici grafu G.

Důkaz:

1. Dle Důsledku 1.2 je hrana libovolného grafu buď most nebo leží na nějaké kružnici grafu.
2. Dle Tvrzení 2.1 sudý graf neobsahuje most.

Druhé pomocné tvrzení

Tvrzení 2.2 (Milková): Pro každý graf $G = (V, E)$ platí:
Jestliže graf G je sudý, pak každá hrana grafu G leží na nějaké kružnici grafu G.

Důkaz:

1. Dle Důsledku 1.2 je hrana libovolného grafu buď most nebo leží na nějaké kružnici grafu.
2. Dle Tvrzení 2.1 sudý graf neobsahuje most.

Z toho vyplývá, že každá hrana sudého grafu leží na nějaké kružnici.

Třetí pomocné tvrzení

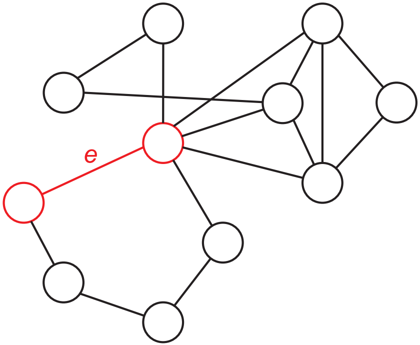
Tvrzení 2.3 (Milková): Pro každý graf $G = (V, E)$ platí:
Jestliže graf G je sudý, pak graf G lze zapsat jako sjednocení kružnic, které jsou navzájem po dvou hranově disjunktní.

Důkaz (nebude vyžadován u zkoušky):

Třetí pomocné tvrzení

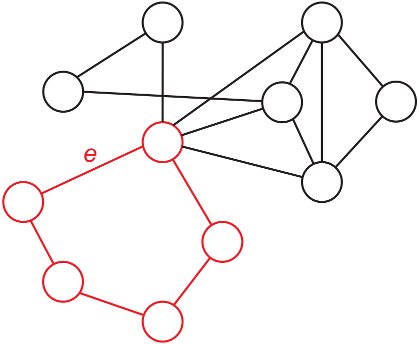
Tvrzení 2.3 (Milková): Pro každý graf $G = (V, E)$ platí:
Jestliže graf G je sudý, pak graf G lze zapsat jako sjednocení kružnic, které jsou navzájem po dvou hranově disjunktní.

Důkaz (nebude vyžadován u zkoušky): Nechť $G = (V, E)$ je sudý graf a e jeho libovolná hrana, viz následující ilustrační obrázek.



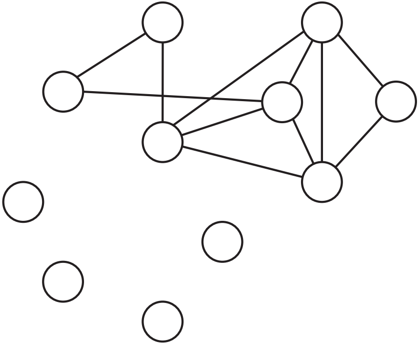
Důkaz Tvrzení 2.3

Dle Tvrzení 2.2 leží hrana e na nějaké kružnici C:



Důkaz Tvrzení 2.3

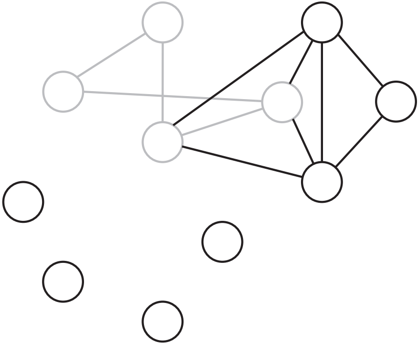
Když odebereme všechny hrany kružnice C z grafu G, dostaneme opět sudý graf:



Proč? Vyjmutím hran kružnice se stupeň libovolného vrcholu kružnice sníží přesně o dva, tj. zůstane sudý.

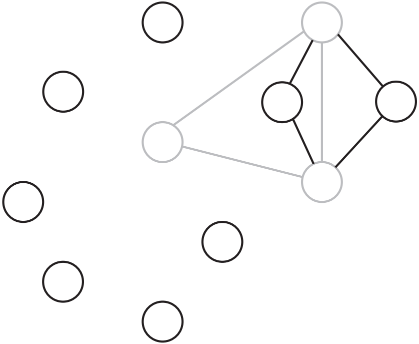
Důkaz Tvrzení 2.3

Postup opakujeme tak dlouho, dokud získáme graf $G' = (V, \emptyset)$:



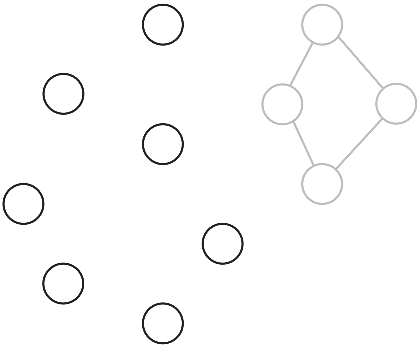
Důkaz Tvrzení 2.3

Postup opakujeme tak dlouho, dokud získáme graf $G' = (V, \emptyset)$:



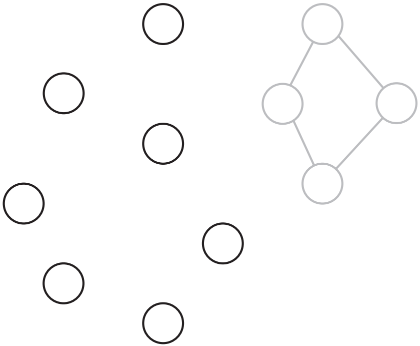
Důkaz Tvrzení 2.3

Postup opakujeme tak dlouho, dokud získáme graf $G' = (V, \emptyset)$:



Důkaz Tvrzení 2.3

Postup opakujeme tak dlouho, dokud získáme graf $G' = (V, \emptyset)$:



Když postupně sjednotíme kružnice, které jsme před chvílí odebírali (a které byly hranově disjunktní), dostaneme původní graf G.

Dokončení důkazu Věty 2.2

Vraťme se zpátky k Větě 2.2:

Věta 2.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je eulerovský *právě tehdy, když* G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně.

Už jsme dokázali směr "$\Rightarrow$".

Dokončení důkazu Věty 2.2

Vraťme se zpátky k Větě 2.2:

Věta 2.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je eulerovský *právě tehdy, když* G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně.

Už jsme dokázali směr "$\Rightarrow$".

Chybí dokázat směr "$\Leftarrow$:"
(*) Graf G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně $\Rightarrow$ G je eulerovský.

Dokončení důkazu Věty 2.2

Vraťme se zpátky k Větě 2.2:

Věta 2.2 (MILKOVÁ): Pro každý graf $G = (V, E)$ platí:
Graf G je eulerovský *právě tehdy, když* G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně.

Už jsme dokázali směr "$\Rightarrow$".

Chybí dokázat směr "$\Leftarrow$:"
(*) Graf G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně $\Rightarrow$ G je eulerovský.

Nejdříve si tvrzení (*) dokážeme pro souvislý graf, v němž jsou všechny vrcholy sudého stupně. Provedeme to matematickou indukcí vzhledem k počtu hran m, $m \geq 1$.

Důkaz Věty 2.2 ve směru “$\Leftarrow$”

Mějme sudý souvislý graf G.

Důkaz Věty 2.2 ve směru "$\Leftarrow$"

Mějme sudý souvislý graf G.

Báze: Nejmenší sudý souvislý graf G má 3 hrany – jedná se o kružnici (trojúhelník) $C_3 \Rightarrow$ kružnice C_3 je uzavřený eulerovský tah.

Důkaz Věty 2.2 ve směru "$\Leftarrow$"

Mějme sudý souvislý graf G.

Báze: Nejmenší sudý souvislý graf G má 3 hrany – jedná se o kružnici (trojúhelník) $C_3 \Rightarrow$ kružnice C_3 je uzavřený eulerovský tah.

Indukční předpoklad: Sudý souvislý graf s počtem hran $\leq m$ obsahuje uzavřený eulerovský tah.

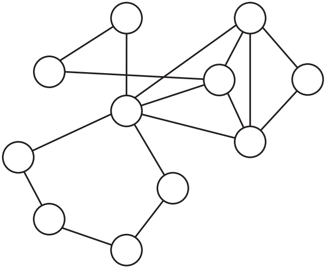
Důkaz Věty 2.2 ve směru "$\Leftarrow$"

Mějme sudý souvislý graf G.

Báze: Nejmenší sudý souvislý graf G má 3 hrany – jedná se o kružnici (trojúhelník) $C_3 \Rightarrow$ kružnice C_3 je uzavřený eulerovský tah.

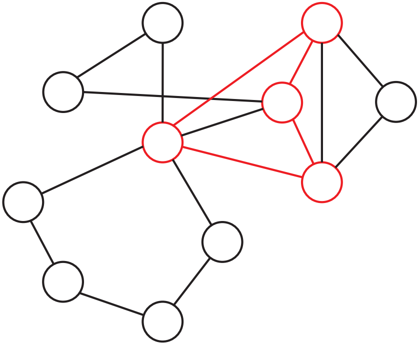
Indukční předpoklad: Sudý souvislý graf s počtem hran $\leq m$ obsahuje uzavřený eulerovský tah.

Indukční krok: Uvažujme sudý souvislý graf, který má $m + 1$ hran.



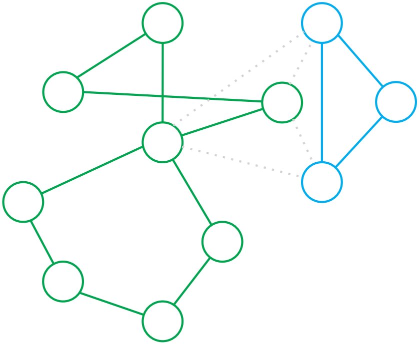
Indukční krok důkazu Věty 2.2 ve směru “$\Leftarrow$”

Dle Tvrzení 2.3 je sudý graf sjednocením několika kružnic navzájem hranově disjunktních. Vybereme některou z těchto kružnic a označíme ji C:



Po odebrání hran kružnice C bude graf $G' = (V, E - \{e \in C\})$ sudý, ne však nutně souvislý.

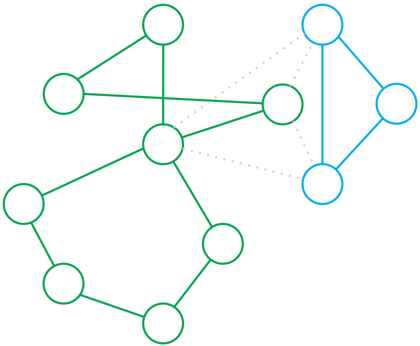
Zbylé komponenty grafu G' jsou sudé grafy, které mají určitě $\leq m$ hran:



Tyto komponenty obsahují dle IP uzavřené eulerovské tahy (jsou souvislé a mají $\leq m$ hran).

Indukční krok důkazu Věty 2.2 ve směru "$\Leftarrow$"

Komponenty (uzavřené eulerovské tahy) spojíme do grafu G hranami kružnice C a získáme uzavřený eulerovský tah o $m + 1$ hranách.



Platí tedy:

(**) *Sudý souvislý graf obsahuje uzavřený eulerovský tah.*

Indukční krok důkazu Věty 2.2 ve směru “$\Leftarrow$”

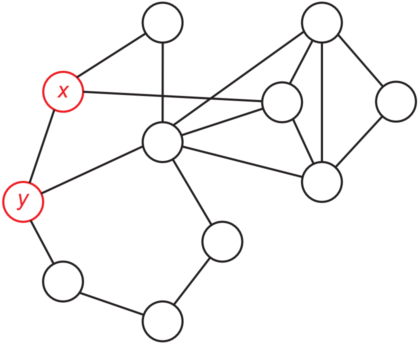
Chtěli jsme dokázat Větu 2.2 ve směru “$\Leftarrow$”, tj. tvrzení:

(*) Graf G je souvislý a má buď všechny vrcholy sudého stupně nebo právě dva vrcholy lichého stupně $\Rightarrow$ G je eulerovský.

Pro sudé souvislé grafy je již tvrzení (*) je dokázáno.

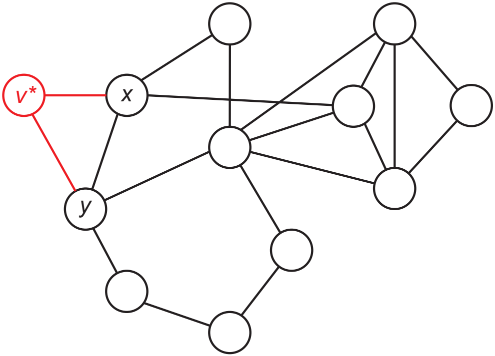
Indukční krok důkazu Věty 2.2 ve směru "$\Leftarrow$"

Uvažujme tedy souvislý graf $G = (V, E)$ s právě dvěma vrcholy $x, y \in V$ lichého stupně. Chceme ukázat, že G obsahuje otevřený eulerovský tah.



Indukční krok důkazu Věty 2.2 ve směru "$\Leftarrow$"

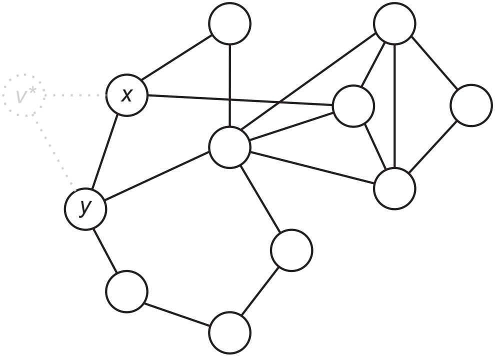
Spojíme vrcholy x, y s novým uzlem v^* a vytvoříme **sudý** graf $G' = (V \cup \{v^*\}, E \cup \{\{x, v^*\}, \{y, v^*\}\}$:



Nový sudý a souvislý graf G' obsahuje, dle dokázaného tvrzení (**), uzavřený eulerovský tah.

Indukční krok důkazu Věty 2.2 ve směru "$\Leftarrow$"

Odebráním uzlu v^* z grafu G' získáme otevřený eulerovský tah v grafu G.



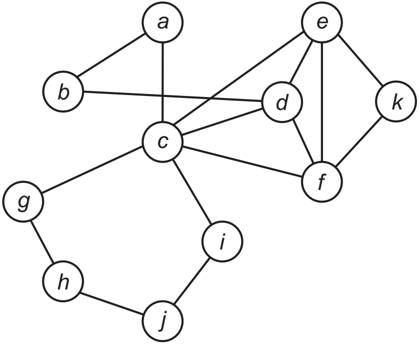
Souvislý graf G s právě dvěma vrcholy lichého stupně je tudíž eulerovský.

Eulerovské vs. Hamiltonovské grafy

- Věta 2.2 je jednoznačným kritériem pro rozhodnutí, zda je zadaný graf eulerovský. Takovou nutnou a postačující podmínku jsme pro hamiltonovské grafy neměli.
- K rozlišení obou typů grafů nám pomůže následující srovnání:
  1. **Eulerovský** graf = problém **průzkumníka** (vydává se na průzkum všech ulic města nebo jeho části, chtěl by každou ulici projít právě jednou).
  2. **Hamiltonovský** graf = problém **cestujícího** (vyjíždí na putování po určených městech a chtěl by si trasu naplánovat tak, aby každé navštívil právě jednou).

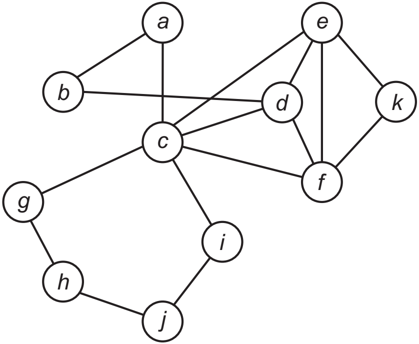
Příklad 2.4

Určete, zda je následující graf eulerovský.



Příklad 2.4

Určete, zda je následující graf eulerovský.



Řešení: zapíšeme si skóre grafu: (2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 6). Jedná se o souvislý a sudý graf, obsahuje tedy uzavřený eulerovský tah. Graf je eulerovský.

Odklízení sněhu z chodníků

Představte si, že jste na brigádě a Vaším úkolem je odklízení sněhu z chodníků v ulicích jisté části města Brna, jejíž plánek máte k dispozici. Víte, že se jedná o ulice, které mají chodníky po obou stranách.

Jak si naplánujete úklid, aby průchod ulicemi byl efektivní (tj. abychom každou stranu ulice prošli právě jednou)?

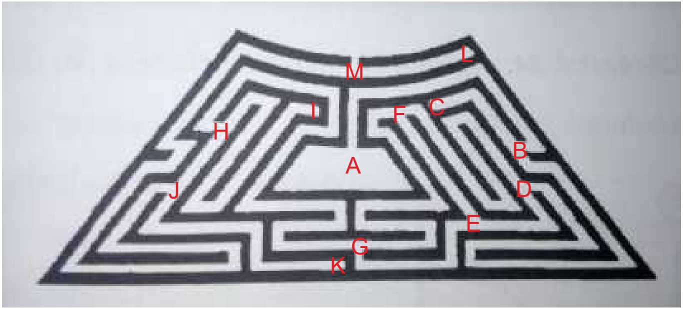
Procházení labyrintů

Ukázka bludiště v zahradě Hampton Court Palace (jihozápadní Londýn) – viz plánek:

První pokusy o nalezení algoritmu pro procházení labyrintu byly učiněny v letech 1873 až 1895 (Wiener, Trémaux, Tarry).

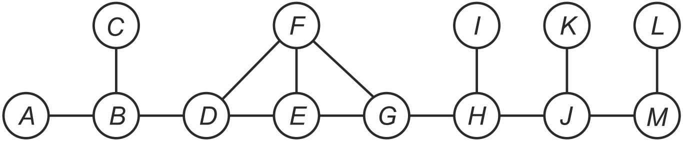
Jak labyrinty převést do grafu?

Křižovatky a konce chodeb označíme písmeny:



Jak labyrinty převést do grafu?

Písmena budou vrcholy grafu. Dva uzly x, y propojíme hranou, pokud mezi nimi existuje chodba:



Edmons-Johnsonův algoritmus

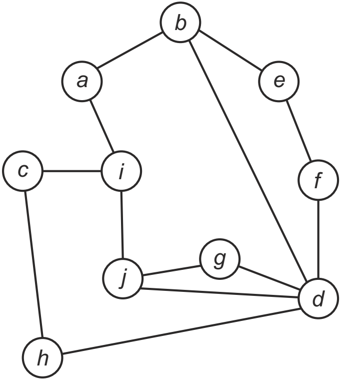
- Tarryho a Trémauxův algoritmus pro procházení labyrintu inspirovaly v r. 1973 **Edmonse a Johnsona** k formulaci vlastní metody procházení labyrintem.
- Edmons-Johnsonův algoritmus slouží k nalezení průchodu labyrintem, přičemž každou chodbu (hranu) procházíme v obou směrech (tam i zpět).
- Hlavní zásady algoritmu:
  1. Každou hranou (chodbou) můžeme projít v jednom směru nejvýše jednou.
  2. Po hraně, po které jsme do vrcholu přišli poprvé, se můžeme vracet pouze tehdy, nemáme-li jinou možnost.
  3. Z hran, které máme k odchodu z vrcholu k dispozici, upřednostňujeme tu hranu, kterou jsme dosud nikdy nešli.

Edmons-Johnsonův algoritmus – principy

1. **Označení chodby**, kterou procházíme poprvé:
   - na jejím začátku vkládáme dvě značky ••
   - na jejím konci vkládáme
     1. jednu značku •, vede-li chodba k již navštívené křižovatce,
     2. tři značky • • •, vede-li chodba k dosud nenavštívené křižovatce.
2. **Označení chodby**, kterou procházíme podruhé (tj. jdeme v opačném směru):
   - na svém začátku má jednu značku •,
   - k této značce na začátku chodby přidáme druhou značku •.
3. **Výběr chodby** k procházení, jsme-li na křižovatce:
   - primárně vybíráme chodbu, která nemá značení;
   - pokud žádná taková neexistuje, vybereme chodbu s jednou značkou • na svém začátku;
   - není-li k dispozici chodba bez značení či s jednou značkou •, vybereme chodbu se třemi značkami • • •.

Příklad 6.1

Mějme plánek města, v jehož ulicích (s chodníky na obou stranách) máme odklidit sníh. Pomocí Edmons-Johnsonova algoritmu najděte posloupnost orientovaných hran označujících průchod ulicemi při odklízení sněhu tam i zpět.



Příklad 6.1 – několik zásad

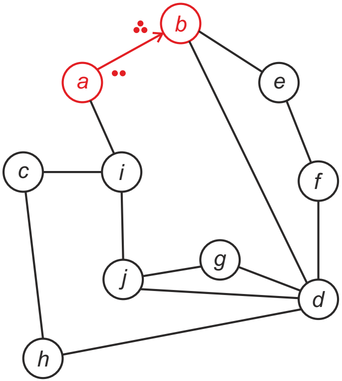
- Poprvé navštívenou chodbu budeme značit uspořádanou dvojicí (x, y), kde x je začátek chodby, y konec chodby.
- Při druhém průchodu chodbou v opačném směru zapíšeme $(\mathbf{y}, \mathbf{x})$, čímž naznačíme, že už je sníh odklizen z obou chodníků ulice.
- Máme-li více možností, jakou chodbou se vydat, respektujeme lexikografické (abecední) pravidlo.

Příklad 6.1 – řešení

Začínáme ve vrcholu a, s nímž sousedí vrcholy b, i. Lexikograficky je blíž b, vydáme se chodbou do b, přičemž na její začátek vložíme ••, na její konec • • •, jelikož křižovatku b jsme ještě nenavštívili.

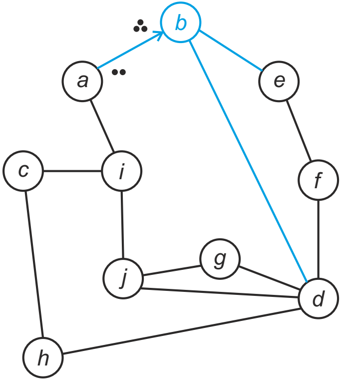
Navštívené chodby: $\emptyset$

Aktuální chodba: (a, b)



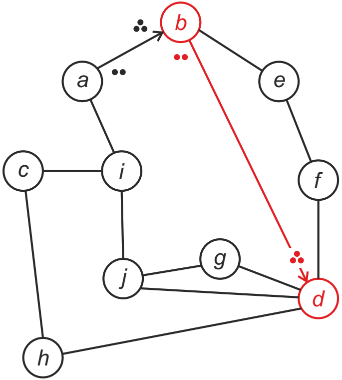
Příklad 6.1 – řešení

V křižovatce b vybíráme chodbu, kterou půjdeme. Máme ještě k dispozici neoznačené chodby, takže vybereme "lexikograficky" hranu (b, d).
Navštívené chodby: (a, b)
Aktuální chodba: (b, d)



Příklad 6.1 – řešení

Označíme chodbu (b, d), přičemž na její začátek vložíme ••, na její konec • • •, jelikož křižovatku d jsme ještě nenavštívili.
Navštívené chodby: (a, b)
Aktuální chodba: (b, d)

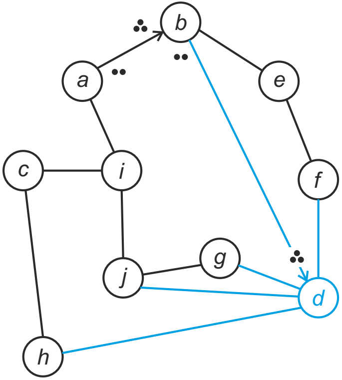


Příklad 6.1 – řešení

V křižovatce d vybíráme chodbu, kterou půjdeme. Máme ještě k dispozici neoznačené chodby, takže vybereme “lexikograficky” hranu (d, f).

Navštívené chodby: $(a, b); (b, d)$

Aktuální chodba: (d, f)

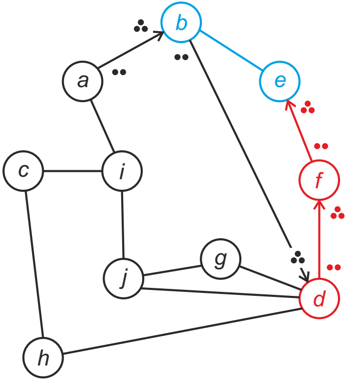


Příklad 6.1 – řešení

Ve dvou krocích projdeme hranami $(b, d); (d, e)$, jejich začátek označíme $\bullet\bullet$, konec $\bullet\bullet\bullet$. Další hrana v pořadí je (e, b).

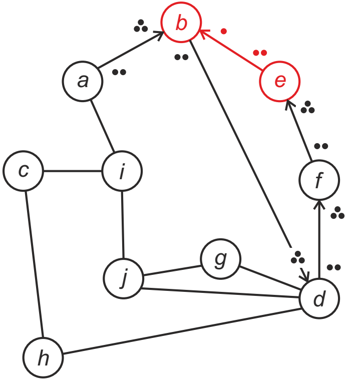
Navštívené chodby: $(a, b); (b, d); (d, e)$

Aktuální chodba: (e, b)



Příklad 6.1 – řešení

Označíme chodbu (e, b), přičemž na její začátek vložíme ••, na její konec pouze •, jelikož křižovatku d jsme již navštívili.
Navštívené chodby: $(a, b); (b, d); (d, e)$
Aktuální chodba: (e, b)

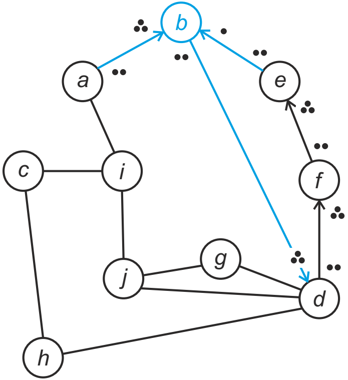


Příklad 6.1 – řešení

V křižovatce b vybíráme chodbu, kterou půjdeme. Nemáme už k dispozici neoznačené chodby, vybereme tedy *zpětnou* hranu $(\mathbf{b}, \mathbf{e})$ s •.

Navštívené chodby: $(a, b); (b, d); (d, e); (e, b)$

Aktuální chodba: $(\mathbf{b}, \mathbf{e})$

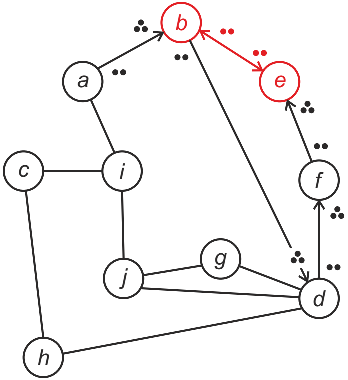


Příklad 6.1 – řešení

Označíme chodbu $(\mathbf{b}, \mathbf{e})$, přičemž na její začátek doplníme • navíc, její konec necháme beze změny.

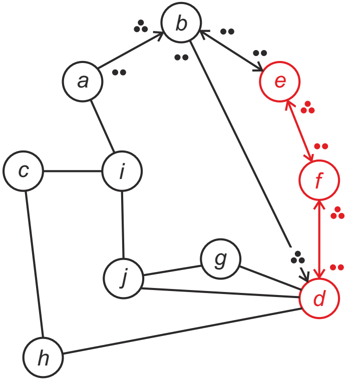
Navštívené chodby: $(a, b); (b, d); (d, e); (e, b)$

Aktuální chodba: $(\mathbf{b}, \mathbf{e})$



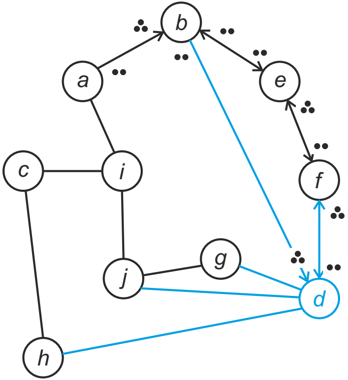
Příklad 6.1 – řešení

Ve dvou krocích projdeme zpětnými hranami $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$ a jejich značení ponecháme beze změny.
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$



Příklad 6.1 – řešení

V křižovatce d vybíráme chodbu, kterou půjdeme. Máme ještě k dispozici neoznačené chodby, takže vybereme “lexikograficky” hranu (d, g).
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$
Aktuální chodba: (d, g)

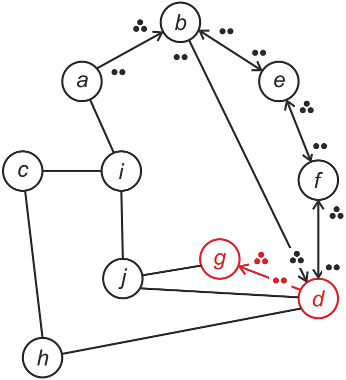


Příklad 6.1 – řešení

Označíme chodbu (d, g), přičemž na její začátek vložíme ••, na její konec • • •, jelikož křižovatku g jsme ještě nenavštívili.

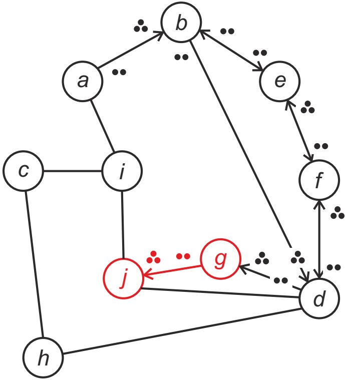
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$

Aktuální chodba: (d, g)



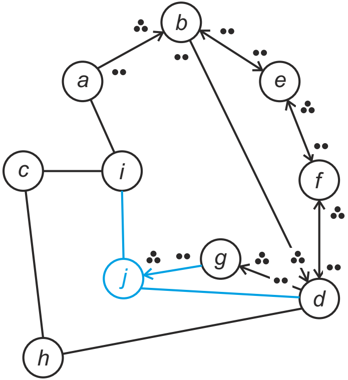
Příklad 6.1 – řešení

Označíme chodbu (g, j), přičemž na její začátek vložíme ••, na její konec • • •, jelikož křižovatku j jsme ještě nenavštívili.
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j)

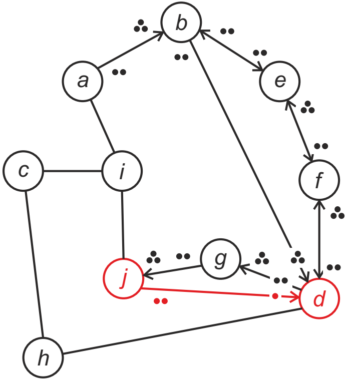


Příklad 6.1 – řešení

V křižovatce j vybíráme chodbu, kterou půjdeme. Máme ještě k dispozici neoznačené chodby, takže vybereme "lexikograficky" hranu (j, d).
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j)



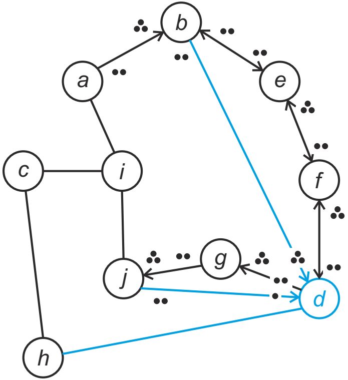
Označíme chodbu (j, d), přičemž na její začátek vložíme ••, na její konec pouze •, jelikož křižovatku d jsme již navštívili.
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j)



Příklad 6.1 – řešení

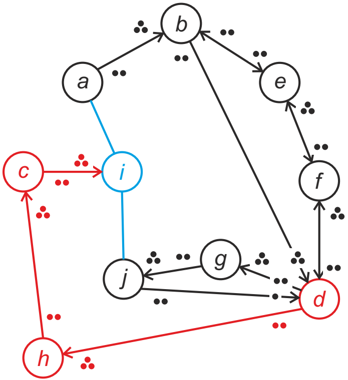
V křižovatce d vybíráme chodbu, kterou půjdeme. Máme ještě k dispozici neoznačenou chodbu (d, h).

Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j); (j, d)



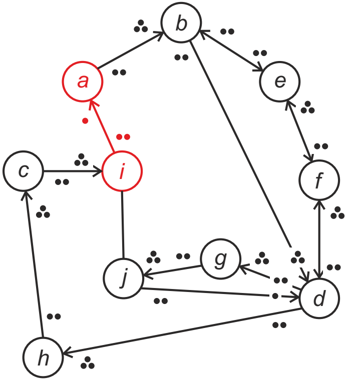
Příklad 6.1 – řešení

Ve několika krocích zpracujeme chodby (d, h); (h, c); (c, i), na jejichž konci jsou neznámé vrcholy – dostaneme se až ke křižovatce i, u níž "lexikograficky" vybereme hranu (i, a).
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j); (j, d); (d, h); (h, c); (c, i)



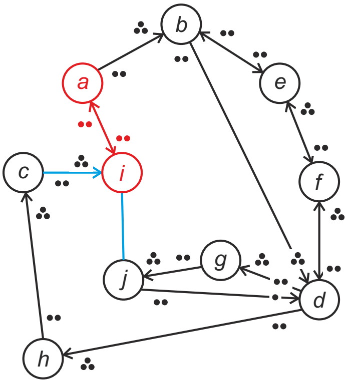
Příklad 6.1 – řešení

Hranu (i, a) projdeme v jednom směru...
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j); (j, d); (d, h); (h, c); (c, i); (i, a)



Příklad 6.1 – řešení

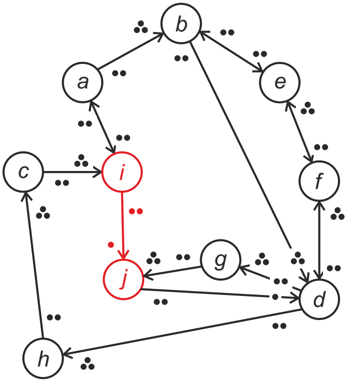
...i ve zpětném směru. V křižovatce i vybíráme chodbu, kterou půjdeme. Máme ještě k dispozici neoznačenou chodbu (i, j).
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j); (j, d); (d, h); (h, c); (c, i); (i, a); **(a, i)**



Příklad 6.1 – řešení

Označíme chodbu (i, j). V křižovatce j vybíráme chodbu, kterou půjdeme. Nemáme už k dispozici neoznačené chodby, vybereme tedy *zpětnou* hranu $(\mathbf{j}, \mathbf{i})$ s $\bullet$.

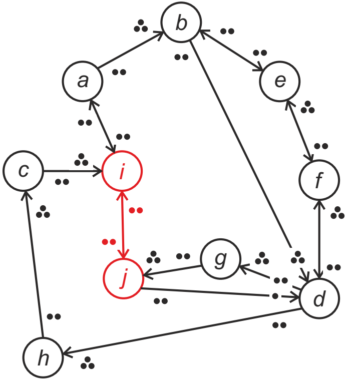
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j); (j, d); (d, h); (h, c); (c, i); (i, a); **(a, i)**; (i, j)



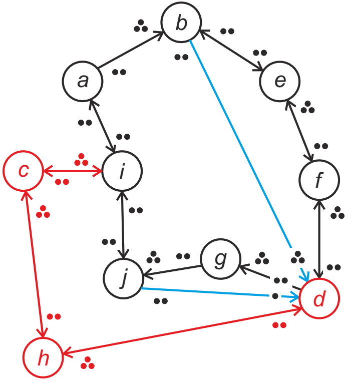
Příklad 6.1 – řešení

Označíme chodbu $(\mathbf{j}, \mathbf{i})$, na začátku přidáme $\bullet$, protože pomocí ní přijdeme ke známé křižovatce i.

Navštívené chodby: $(a, b); (b, d); (d, e); (e, b); (\mathbf{b}, \mathbf{e}); (\mathbf{e}, \mathbf{f}); (\mathbf{f}, \mathbf{d}); (d, g);$ $(g, j); (j, d); (d, h); (h, c); (c, i); (i, a);$ **(a, i)**; $(i, j); (\mathbf{j}, \mathbf{i})$



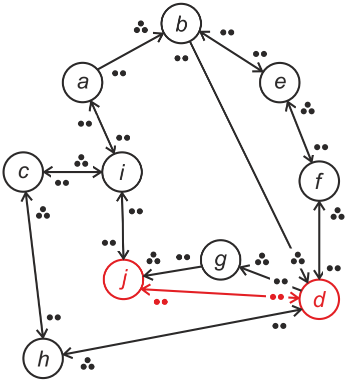
Ve několika krocích zpracujeme chodby $(\mathbf{i}, \mathbf{c})$; $(\mathbf{c}, \mathbf{h})$; $(\mathbf{h}, \mathbf{d})$ a dostaneme se až ke křižovatce d, u níž vybereme hranu $(\mathbf{d}, \mathbf{j})$ s $\bullet$.
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j); (j, d); (d, h); (h, c); (c, i); (i, a); **(a, i)**; (i, j); $(\mathbf{j}, \mathbf{i})$; $(\mathbf{i}, \mathbf{c})$; $(\mathbf{c}, \mathbf{h})$; $(\mathbf{h}, \mathbf{d})$



Příklad 6.1 – řešení

Označíme chodbu $(\mathbf{d}, \mathbf{j})$, na začátku přidáme $\bullet$, protože pomocí ní přijdeme ke známé křižovatce j.

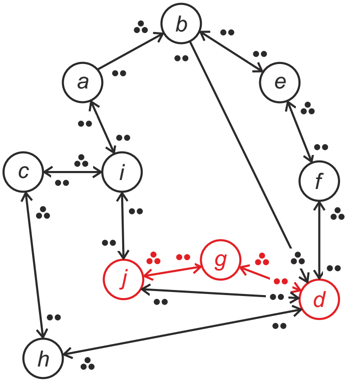
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j); (j, d); (d, h); (h, c); (c, i); (i, a); $(\mathbf{a}, \mathbf{i})$; (i, j); $(\mathbf{j}, \mathbf{i})$; $(\mathbf{i}, \mathbf{c})$; $(\mathbf{c}, \mathbf{h})$; $(\mathbf{h}, \mathbf{d})$ $(\mathbf{d}, \mathbf{j})$



Příklad 6.1 – řešení

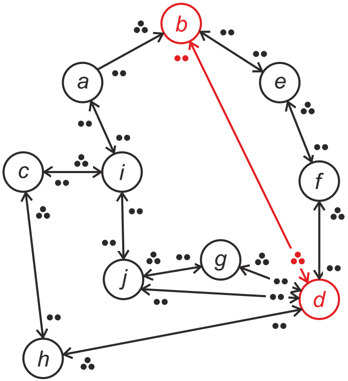
Ve dvou krocích projdeme zpětnými hranami $(\mathbf{j}, \mathbf{g})$; $(\mathbf{g}, \mathbf{d})$ a jejich značení ponecháme beze změny.

Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j); (j, d); (d, h); (h, c); (c, i); (i, a); **(a, i)**; (i, j); $(\mathbf{j}, \mathbf{i})$; $(\mathbf{i}, \mathbf{c})$; $(\mathbf{c}, \mathbf{h})$; $(\mathbf{h}, \mathbf{d})$ $(\mathbf{d}, \mathbf{j})$; $(\mathbf{j}, \mathbf{g})$; $(\mathbf{g}, \mathbf{d})$



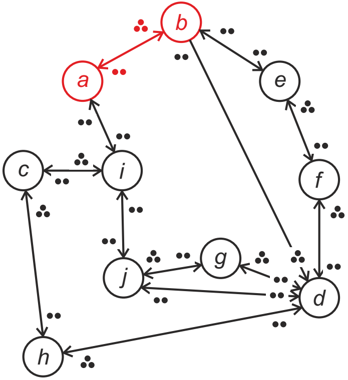
Příklad 6.1 – řešení

Na křižovatce d už máme jedinou možnost, "zpětnou" chodbu $(\mathbf{d}, \mathbf{b})$ s • • • na začátku. Pouze jí projdeme ke křižovatce b.
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j); (j, d); (d, h); (h, c); (c, i); (i, a); $(\mathbf{a}, \mathbf{i})$; (i, j); $(\mathbf{j}, \mathbf{i})$; $(\mathbf{i}, \mathbf{c})$; $(\mathbf{c}, \mathbf{h})$; $(\mathbf{h}, \mathbf{d})$ $(\mathbf{d}, \mathbf{j})$; $(\mathbf{j}, \mathbf{g})$; $(\mathbf{g}, \mathbf{d})$; $(\mathbf{d}, \mathbf{b})$



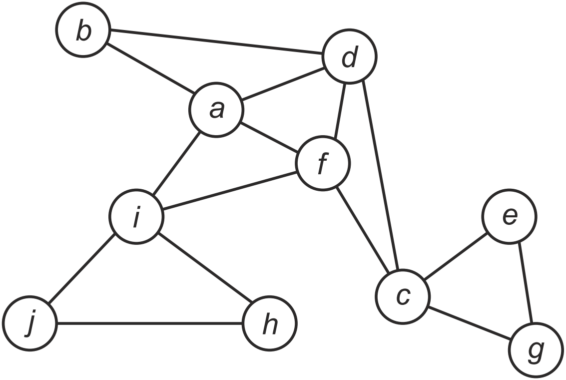
Příklad 6.1 – řešení

Dokončíme naší "pouť" průchodem chodby $(\mathbf{b}, \mathbf{a})$.
Navštívené chodby: (a, b); (b, d); (d, e); (e, b); $(\mathbf{b}, \mathbf{e})$; $(\mathbf{e}, \mathbf{f})$; $(\mathbf{f}, \mathbf{d})$; (d, g); (g, j); (j, d); (d, h); (h, c); (c, i); (i, a); $(\mathbf{a}, \mathbf{i})$; (i, j); $(\mathbf{j}, \mathbf{i})$; $(\mathbf{i}, \mathbf{c})$; $(\mathbf{c}, \mathbf{h})$; $(\mathbf{h}, \mathbf{d})$ $(\mathbf{d}, \mathbf{j})$; $(\mathbf{j}, \mathbf{g})$; $(\mathbf{g}, \mathbf{d})$; $(\mathbf{d}, \mathbf{b})$; $(\mathbf{b}, \mathbf{a})$



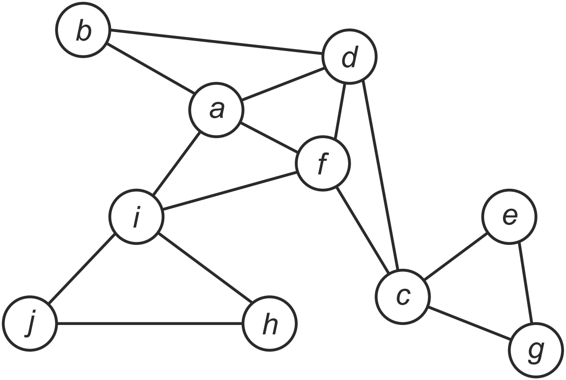
Hledání zvonků se jménem *Novák* či *Nováková*

Úloha *Zvonky*: Představme si, že máme k dispozici plánek určité městské části Brna (viz níže) a s jeho pomocí chceme projít všemi ulicemi a zjistit, přečtením jmen u zvonků každého domu, kolik se v městské části nachází domů, ve kterých bydlí Novákovi.



Hledání pamětních desek na domech

Úloha *Pamětní desky*: Mějme opět plánek stejné městské části Brna (viz níže) a s jeho pomocí chceme projít všemi ulicemi a zjistit, kolik se tam nachází domů s pamětní deskou informující o tom, že tam žil známý člověk.



Jak obě úlohy řešit?

1. Úlohu *Zvonky* řešíme pomocí algoritmů pro průchod labyrintem, protože ulice potřebujeme projít v obou směrech, abychom přečetli zvonky u každého domu.
2. Při řešení úlohy *Pamětní desky* stačí každou ulicí projít právě jednou – hledáme tedy v daném plánku eulerovský tah.

Hledání eulerovského tahu

- Když vhodně aplikujeme Edmons-Johnsonův algoritmus na eulerovský graf, můžeme pomocí něj najít eulerovský tah.
- *Zpětný průchod* hranami při Edmons-Johnsonově algoritmu určuje eulerovský tah.
- Je třeba využít datovou strukturu *zásobník* fungující na principu LIFO (Last In First Out).

Princip algoritmu pro nalezení eulerovského tahu

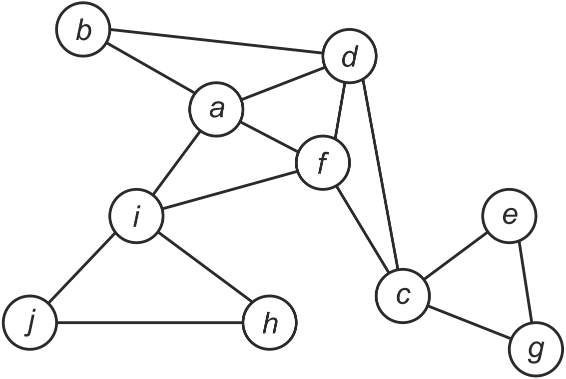
1. Začneme v nějakém vrcholu x a modrou pastelkou kreslíme nějaký tah, dokud to jde.
   - Buď skončíme znovu ve vrcholu x a našli jsme uzavřený tah.
   - Nebo skončíme v jiném vrcholu y a našli jsme otevřený tah.
   - V obou případech zbylý "neobarvený" graf je sudý $\Rightarrow$ existuje v něm uzavřený eulerovský tah.
2. Vracíme se zpět proti směru tahu a hrany označujeme červenou pastelkou, přičemž hledáme vrchol, ze kterého lze začít další tah, tj. uzel, který je incidentní s nějakou neobarvenou hranou.
3. Jakmile neobarvenou hranu najdeme, vybereme jí a pokračujeme krokem 1.
4. Pokračujeme tak dlouho, dokud máme neobarvené hrany.

Použití zásobníků

- K uchování přesné informace o průběhu tahu nakresleného modrou pastelkou používáme zásobník Z, kam vkládáme každý vrchol, kterým modrý tah prochází.
- Ke zpětné rekonstrukci zpětného průchodu nakresleného červeného pastelkou používáme zásobník ET, kam vkládáme každý vrchol, kterým zpětný červený průchod prochází.
- Na začátku algoritmu je zásobník ET prázdný, zásobník Z obsahuje
  1. libovolně vybraný vrchol v, je-li eulerovský graf sudý.
  2. jeden z vrcholů lichého stupně, obsahuje-li eulerovský graf právě dva uzly lichého stupně.
- Provádíme-li zpětný průchod, kopírujeme uzel z *vrcholu* zásobníku Z na *vrchol* zásobníku ET.
- Algoritmus končí v okamžiku, kdy je zásobník Z prázdný. V té chvíli je eulerovský tah "zaznamenán" na zásobníku ET.

Příklad 6.2

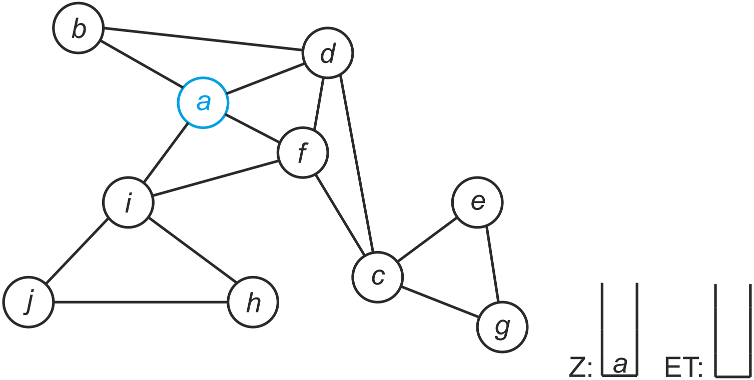
Nalezněte eulerovský tah v následujícím grafu.



Při volbě následující hrany tahu uplatňujte lexikografické pravidlo.

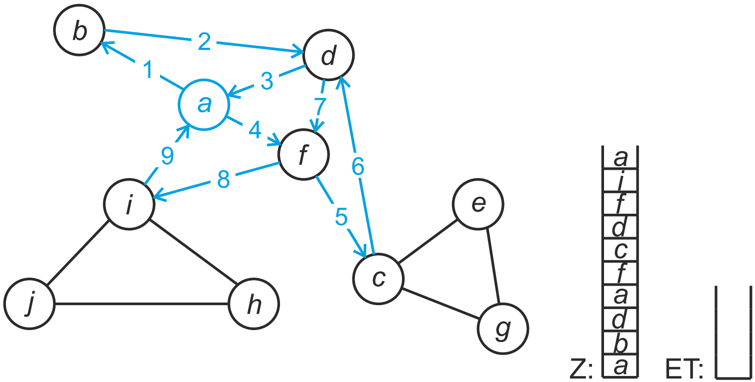
Příklad 6.2 – řešení

Nejdříve si připravíme zásobníky Z a ET. Začínáme ve vrcholu *a* a hledáme co nejdelší tah.



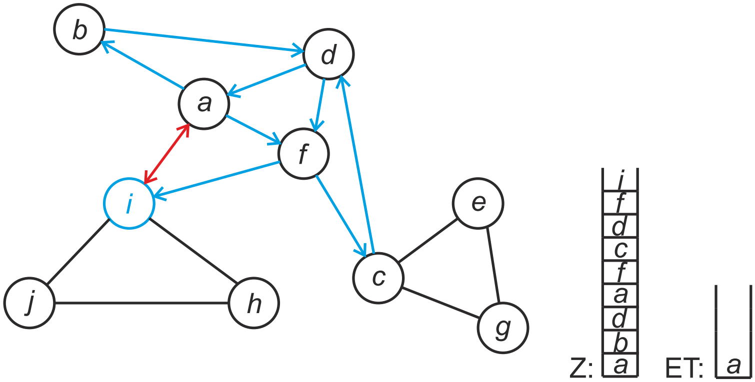
Příklad 6.2 – řešení

Modrou barvou vyznačíme tah, přičemž veškeré vrcholy, které projdeme, vkládáme do zásobníku Z. Skončíme ve vrcholu *a*.



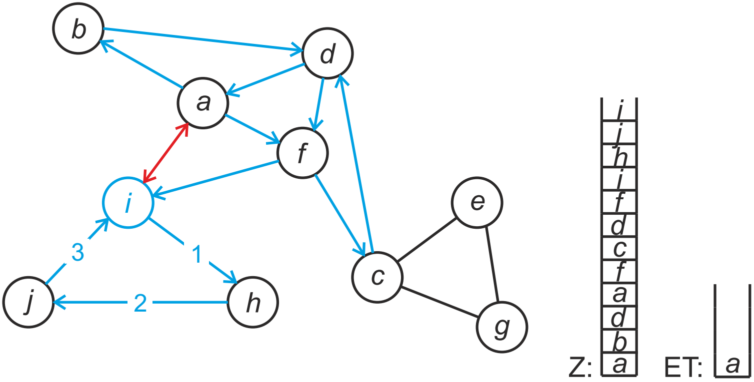
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol *a* a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



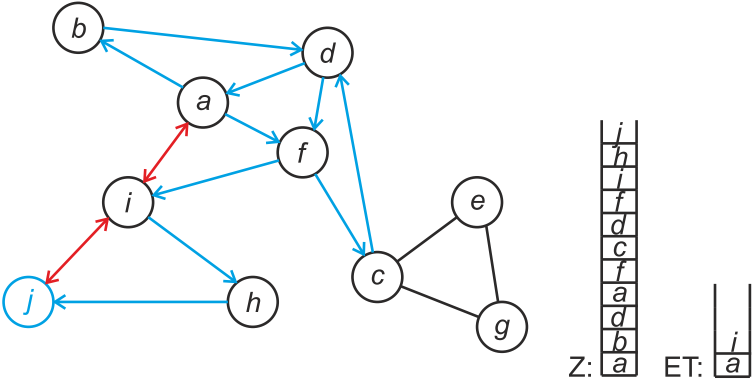
Příklad 6.2 – řešení

Modrou barvou vyznačíme tah z vrcholu i, přičemž veškeré vrcholy, které projdeme, vkládáme do zásobníku Z. Skončíme opět ve vrcholu i.



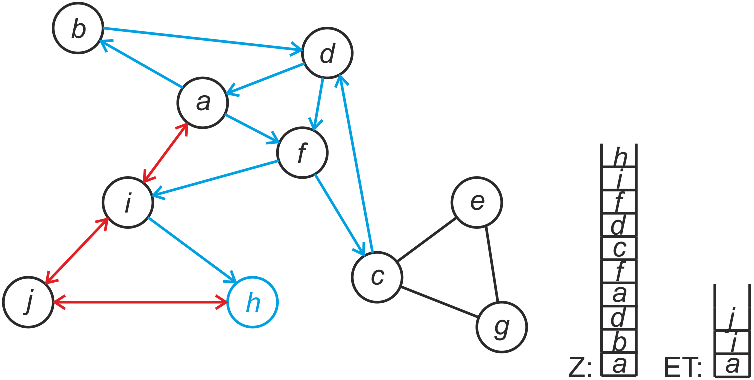
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol *i* a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



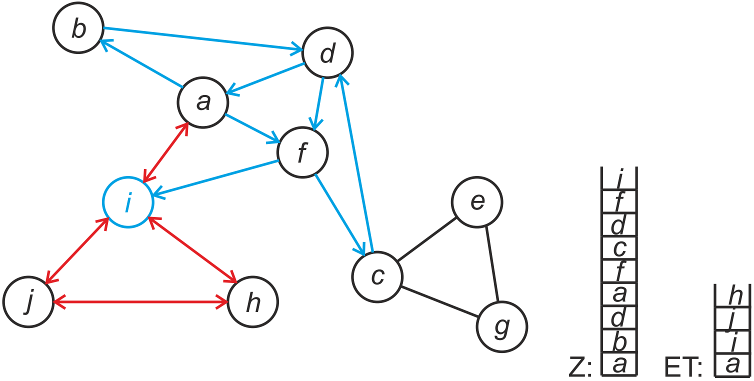
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol *h* a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



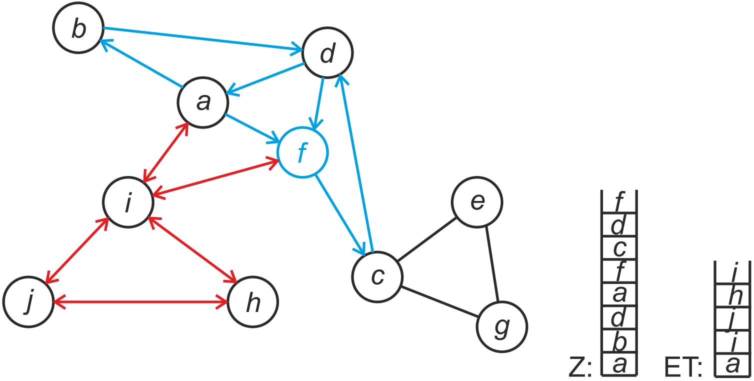
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol j a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



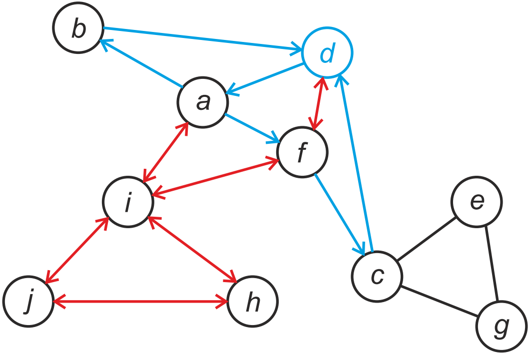
Příklad 6.2 – řešení

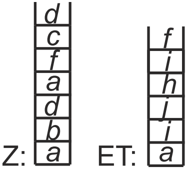
Ze zásobníku Z vezmeme vrchol *i* a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



Příklad 6.2 – řešení

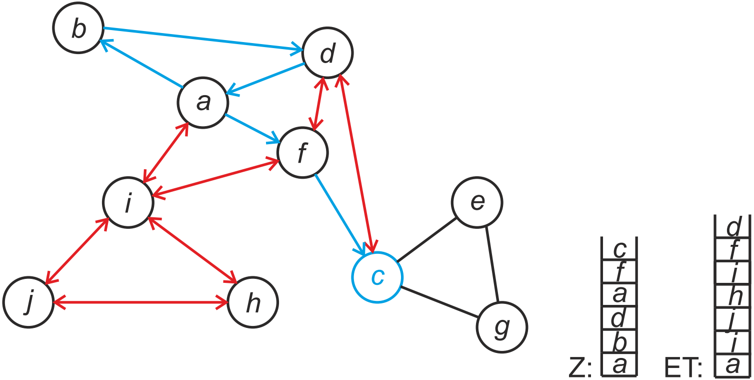
Ze zásobníku Z vezmeme vrchol f a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.





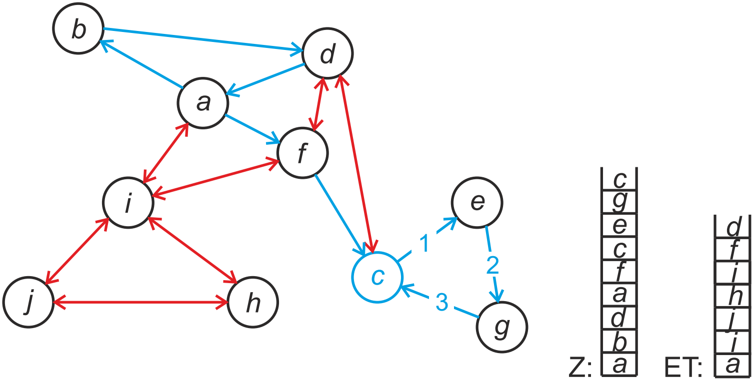
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol *d* a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



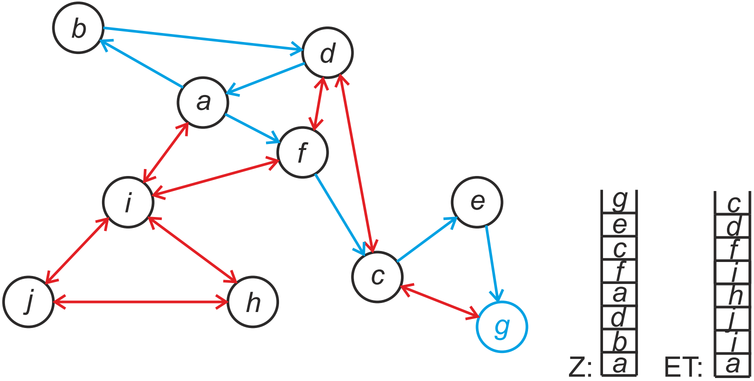
Příklad 6.2 – řešení

Modrou barvou vyznačíme tah z vrcholu c, přičemž veškeré vrcholy, které projdeme, vkládáme do zásobníku Z. Skončíme opět ve vrcholu c.



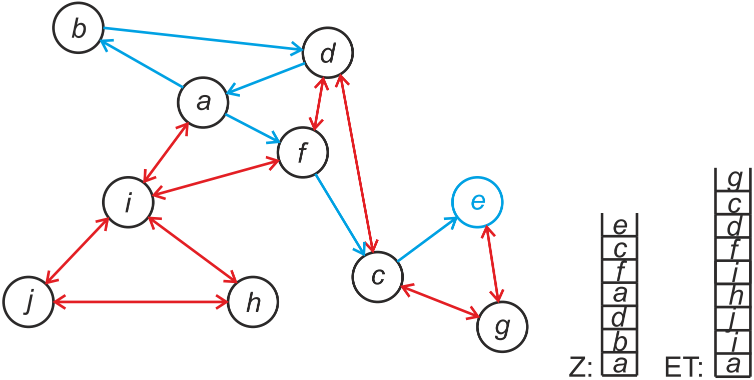
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol c a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



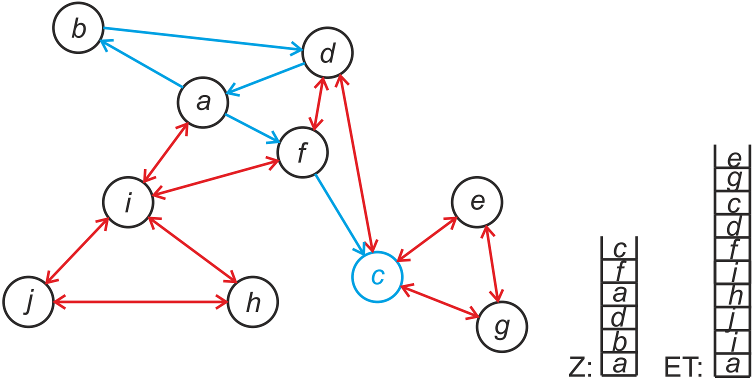
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol g a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



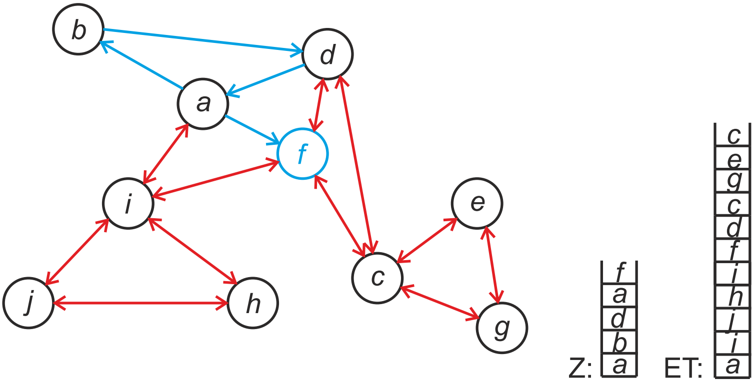
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol e a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



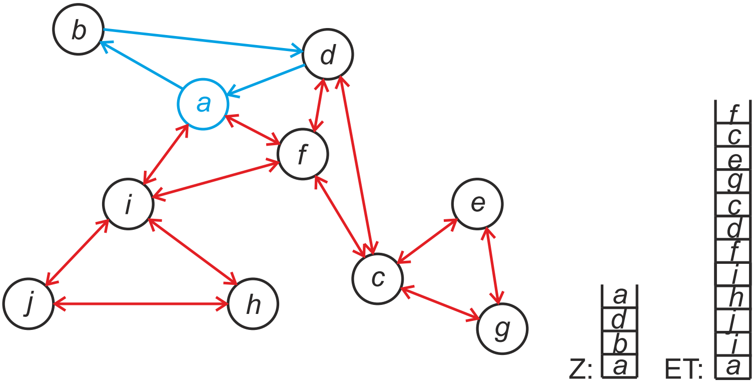
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol c a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



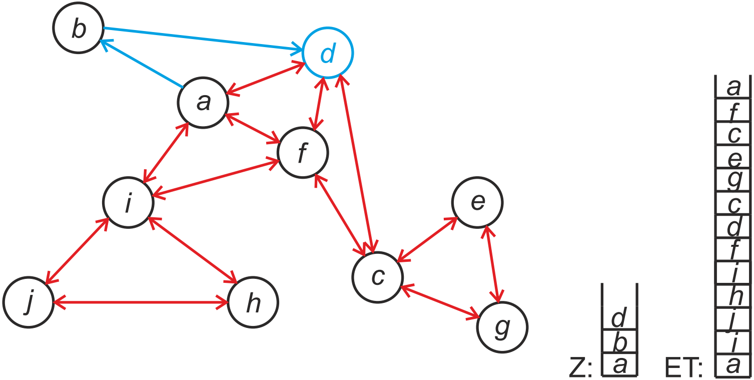
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol f a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



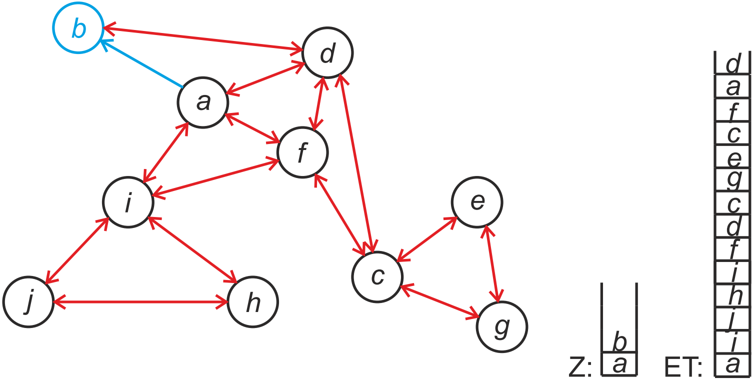
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol *a* a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



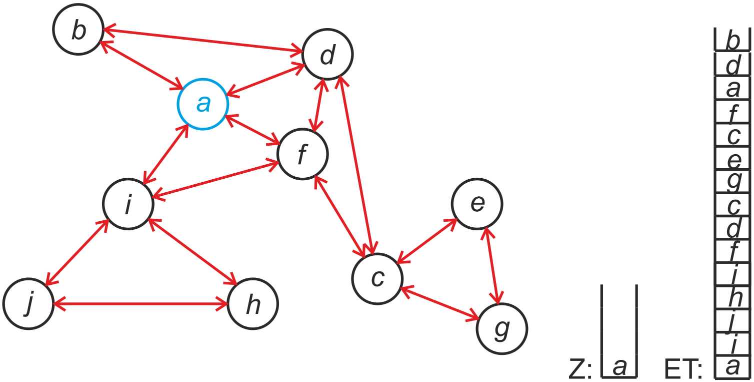
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol d a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



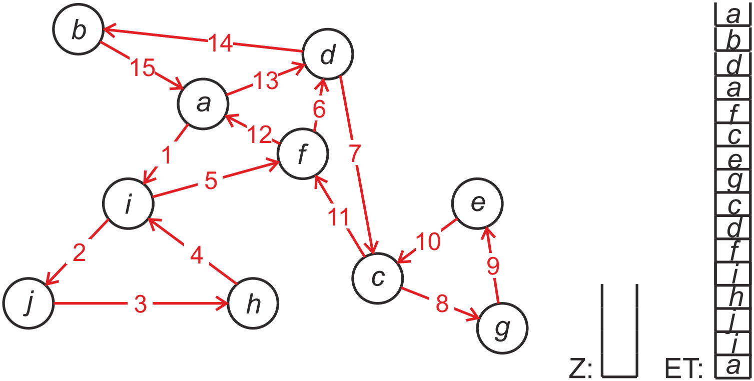
Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol b a vložíme jej do zásobníku ET. Nový vrchol zásobníku Z ukazuje, kam máme jít zpětnou hranou, kterou označíme červeně.



Příklad 6.2 – řešení

Ze zásobníku Z vezmeme vrchol a a vložíme jej do zásobníku ET. Zásobník Z je prázdný $\Rightarrow$ algoritmus je u konce. Pomocí zásobníku ET zpětně zrekonstruujeme eulerovský tah.



Použité zdroje

MILKOVÁ, Eva. *Teorie grafů a grafové algoritmy.* 1. vyd. Hradec Králové: Nakladatelství Gaudeamus, 2013. 123 s. ISBN 978-80-7435-267-6.