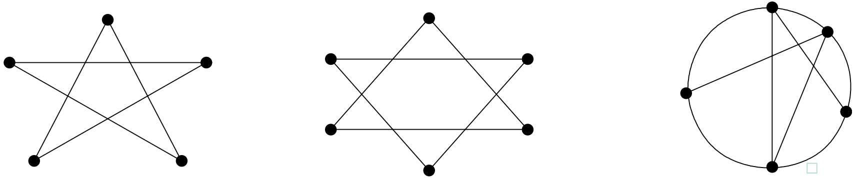
2 Souvislost grafů

Pokud máme graf, který modeluje nějaká spojení či síť, přirozeně nás zajímá, jakou máme možnost se dostat odněkud někam v tomto grafu. To má množství praktických motivací – například počítačové, dopravní, telefonní či potrubní sítě. Je pochopitelné, že v takových sítích chceme mít možnost se dostat z každého místa do každého jiného.

Grafům s takovou vlastností říkáme *souvislé*.



□

Stručný přehled lekce

- Definice souvislosti grafu, vrcholová / hranová, vyšší souvislost.
- Algoritmus procházení grafem (souvislou komponentou).
- Eulerovské grafy.

2.1 Spojení vrcholů, komponenty

Definice: *Sledem* délky n v grafu G rozumíme posloupnost vrcholů a hran

$$v_0, e_1, v_1, e_2, v_2, \dots, e_n, v_n ,$$

ve které vždy hrana e_i má koncové vrcholy v_{i-1}, v_i.

Sled je vlastně procházka po hranách grafu z u do v. Příkladem sledu může být průchod IP paketu internetem (včetně cyklení). □

Lema 2.1. *Mějme relaci $\sim$ na množině vrcholů $V(G)$ libovolného grafu G takovou, že pro dva vrcholy $u \sim v$ právě když existuje v G sled začínající v u a končící ve v. Pak $\sim$ je relací ekvivalence.*

Důkaz. Relace $\sim$ je reflexivní, neboť každý vrchol je spojený sám se sebou sledem délky 0. Symetrická je také, protože sled z u do v snadno obrátíme na sled z v do u. Stejně tak je $\sim$ tranzitivní, protože dva sledy můžeme na sebe navázat v jeden. □ □

Definice: Třídy ekvivalence výše popsané (Lema 2.1) relace $\sim$ na $V(G)$ se nazývají *komponenty souvislosti* grafu G.

Jinak se taky *komponentami souvislosti myslí podgrafy* indukované na těchto třídách ekvivalence.

Připomeňme si, že *cesta v grafu* je vlastně sledem bez opakování vrcholů.

Věta 2.2. *Pokud mezi dvěma vrcholy grafu G existuje sled, pak mezi nimi existuje cesta.* □

Důkaz. Nechť $u = v_0, e_1, v_1, \dots, e_n, v_n = v$ je sled délky n mezi vrcholy u a v v G. Začneme budovat nový sled W z vrcholu $w_0 = u$, který už bude cestou:

– Předpokládejme, že nový sled W už má počátek $w_0, e_1, w_1, \dots, w_i$ (na začátku $i = 0$, tj. jen w_0 bez hran), kde $w_i = v_j$ pro některé $j \in \{0, 1, \dots, n\}$. □

– Najdeme největší index $k \geq j$ takový, že $v_k = v_j = w_i$, a sled W pokračujeme krokem $\dots, w_i = v_j = v_k, e_{k+1}, w_{i+1} = v_{k+1}, \dots$. □

– Zbývá dokázat, že nový vrchol $w_{i+1} = v_{k+1}$ se ve sledu W neopakuje. Pokud by tomu ale tak bylo $w_l = w_{i+1}$, $l \leq i$, pak bychom na vrchol w_{i+1} „přeskočili" už dříve z vrcholu w_l, spor.

– Nakonec skončíme, když $w_i = v$. □

□

Ačkoliv uvedený důkaz vypadá složitě, je to jen jeho formálním zápisem. Ve skutečnosti se v důkaze neděje nic jiného, než že se původní sled zkracuje o opakované vrcholy, až nakonec zákonitě vznikne cesta. Jeho výhodou je konstruktivnost – vidíme, jak cestu získat.

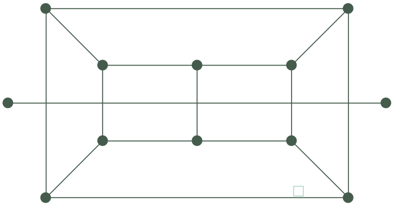
Důkaz kratší, ale nekonstruktivní, pro Větu 2.2:

Ze všech sledů mezi vrcholy u a v v G vybereme sled W s nejmenší délkou. Je snadno vidět, že pokud W zopakuje některý vrchol grafu G, můžeme W ještě zkrátit, a to je spor s předpokladem. Proto je W cestou v G. □ □

Závěrem se dostáváme k nejdůležitější definici souvislého grafu:

Definice 2.3. Graf G je souvislý
pokud je G tvořený nejvýše jednou komponentou souvislosti, tj. pokud každé dva vrcholy G jsou spojené cestou (dle Věty 2.2).

Podívejte se, kolik komponent souvislosti má tento graf:



Vidíte obě dvě komponenty?

2.2 Prohledávání grafu

Pro vytvoření co nejobecnějšího schématu *algoritmu pro procházení grafu* vystačíme s následujícími datovými stavy a pomocnou strukturou:

- **Vrchol**: má stavy ...
  - iniciační – dostane na začátku,
  - nalezený – poté, co jsme jej přes některou hranu nalezli,
  - zpracovaný – poté, co jsme už probrali všechny hrany z něj vycházející.
- **Hrana**: má stavy ...
  - iniciační – dostane na začátku,
  - zpracovaná – poté, co už byla probrána od jednoho ze svých vrcholů. □
- **Úschovna**: je pomocná datová struktura (množina),
  - udržuje nalezené a ještě nezpracované vrcholy.

Poznámka: Způsob, kterým se vybírají vrcholy z úschovny ke zpracování, určuje variantu algoritmu procházení grafu. V prohledávaných vrcholech a hranách se pak provádějí konkrétní programové akce pro prohledání a zpracování našeho grafu.

Algoritmus 2.4. Procházení souvislé komponenty grafu

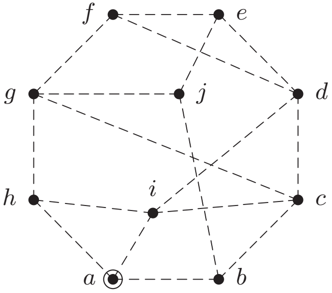
Algoritmus projde a zpracuje každou hranu a vrchol *souvislého* *grafu* G.

```
vstup < graf G;
stav(všechny vrcholy a hrany G ) = iniciační;
uschovna U = {libovolný vrchol v0 grafu G};
stav(v0) = nalezený;
while (U je neprázdná) {
    vybrat v ∈ U;   U = U\{v};
    ZPRACUJ(v);
    foreach (e hrana vycházející z v) {
        if (stav(e)==iniciační) ZPRACUJ(e);
        w = opačný vrchol hrany e = vw;
        if (stav(w)==iniciační) {
            stav(w) = nalezený;
            U = U∪{w};
        }
        stav(e) = zpracovaná;
    }
    stav(v) = zpracovaný;
}
G zpracovaný;
```

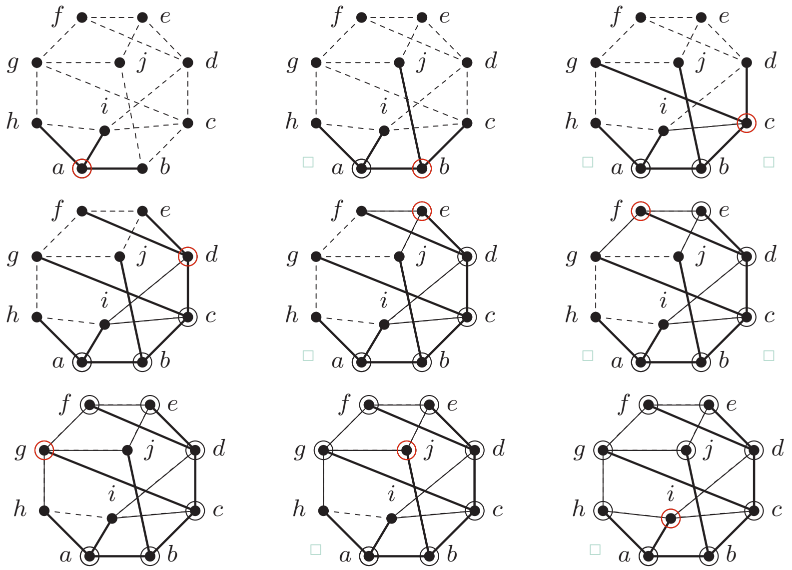
Způsoby implementace procházení grafu

- *Procházení „do hloubky“* – úschovna `U` je implementovaná jako zásobník, tj. dále prohledáváme od posledních nalezených vrcholů. □
- *Procházení „do šířky“* – úschovna `U` je implementovaná jako fronta, tj. dále prohledáváme od prvních nalezených vrcholů. □
- *Dijkstrův algoritmus* pro nejkratší cestu – z úschovny vybíráme vždy vrchol nejbližší k počátečnímu v_0. (Toto je dost podobné prohledávání do šířky, ale obecnější i pro případy, kdy hrany nejsou „stejně dlouhé“.)
  Tento algoritmus bude popsán v příští lekci. □

Příklad 2.11. *Ukázka průchodu následujícím grafem do hloubky z vrcholu a.*

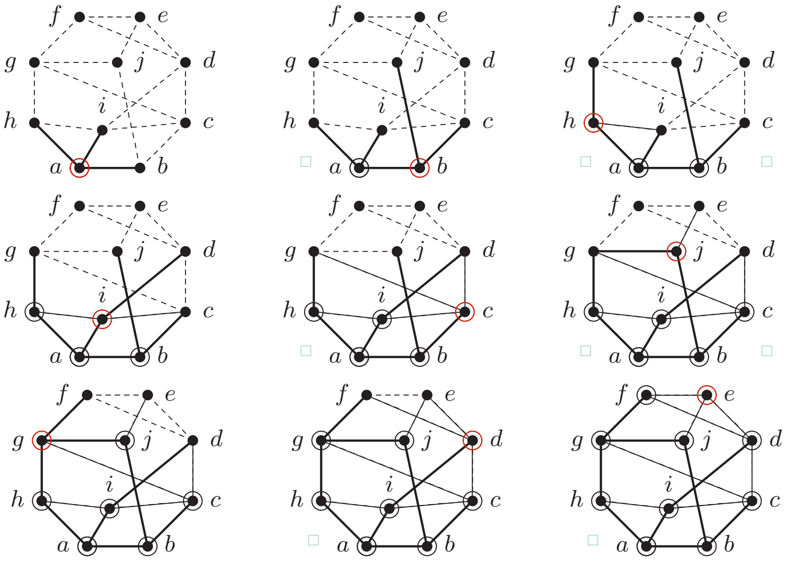


Neprohledané hrany jsou čárkované, prohledané hrany plnou čarou a hrany, které vedly k nalezení vrcholů, jsou tlustou čarou (tyto hrany často mívají speciální význam v aplikacích schématu algoritmu). Nalezené vrcholy se poznají podle příchozí tlusté hrany a zpracované vrcholy jsou značené dvojím kroužkem.



□

Příklad 2.12. *Ukázka průchodu předchozím grafem do šířky z vrcholu a.*



Tímto zpracování zadaného grafu skončilo. Vidíte rozdíly tohoto průchodu proti předchozímu příkladu? □

2.3 Vyšší stupně souvislosti

V *síťových aplikacích* nás často zajímá nejen, jestli se za normálních podmínek můžeme pohybovat mezi vrcholy/uzly, ale také, jaké spojení můžeme nalézt v případě lokálních výpadků (odolnost a redundance).

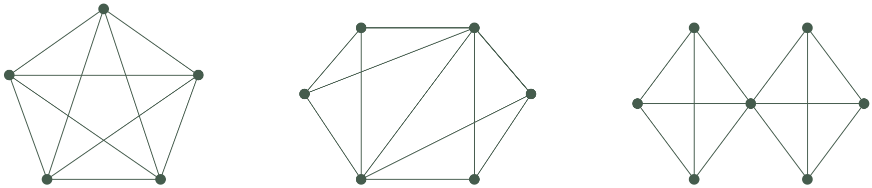
Toto lze teoreticky podchytit zkoumáním „vyšších" stupňů souvislosti grafu. □

Definice: Graf G je *hranově k-souvislý*, $k > 1$, pokud i po odebrání libovolných nejvýše $k-1$ hran z G zůstane výsledný graf souvislý. □

Definice: Graf G je *vrcholově k-souvislý*, $k > 1$, pokud i po odebrání libovolných nejvýše $k-1$ vrcholů z G zůstane výsledný graf souvislý.
Speciálně úplný graf K_n je vrcholově $(n-1)$-souvislý.

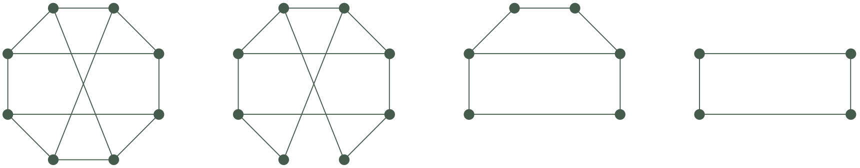
Pokud mluvíme jen o k-souvislém grafu, máme na mysli vrcholově k-souvislý graf. □

Stručně řečeno, vysoká hranová souvislost znamená vysoký stupeň odolnosti sítě proti výpadkům spojení-hran, neboli síť zůstane stále dosažitelná, i když libovolných $k-1$ spojení bude přerušeno. Vysoká vrcholová souvislost je mnohem silnějším pojmem, znamená totiž, že síť zůstane dosažitelná i po výpadku libovolných $k-1$ uzlů-vrcholů (samozřejmě mimo těch vypadlých uzlů).



Na ilustračním obrázku má první graf vrcholovou souvislost 4 a snadno vidíme, že po odebrání tří vrcholů či hran zůstává souvislý. Z druhého grafu bychom museli odebrat nejméně 3 hrany, aby se stal nesouvislým, a proto je jeho hranová souvislost 3. Na druhou stranu však stačí odebrat 2 vrcholy, aby mezi jeho levým a pravým krajním vrcholem žádné spojení nezůstalo. (Vidíte, které dva?) A jak je tomu u třetího grafu? □

Věta 2.5. *Libovolný obyčejný graf je 2-souvislý, právě když jej lze vytvořit z kružnice „přidáváním uší“; tj. iterací operace, kdy libovolné dva stávající vrcholy grafu jsou spojeny novou cestou libovolné délky (ale ne paralelní hranou).*



Mengerova věta

Důkaz následujícího důležitého výsledku by nebyl jednoduchý při použití stávajících znalostí, proto jej ponecháme na pozdější lekce... („Toky v sítích“.)

Věta 2.6. *Graf G je hranově k-souvislý právě když mezi libovolnými dvěma vrcholy lze vést aspoň k hranově-disjunktních cest (vrcholy mohou být sdílené).*
Graf G je vrcholově k-souvislý právě když mezi libovolnými dvěma vrcholy lze vést aspoň k disjunktních cest (různých až na ty dva spojované vrcholy). □

Věta nám vlastně říká, že stupeň souvislosti grafu se přirozeně rovná stupni redundance spojení vrcholů. Na výše uvedeném obrázku mezi každými dvěma vrcholy prvního grafu můžeme vést až 4 disjunktní cesty.
U druhého grafu třeba mezi levým a pravým koncem lze vést jen 2 (vrcholově) disjunktní cesty, ale mezi každými dvěma vrcholy lze vést 3 hranově-disjunktní cesty.

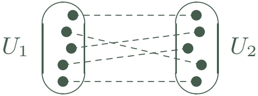
V duchu předchozí Mengerovy věty pokračujeme s následujícími poznatky.

Věta 2.7. *Nechť G je vrcholově 2-souvislý graf. Pak každé dvě hrany v G leží na společné kružnici.* □

Důkaz: Nechť $e, f \in E(G)$. Sestrojíme graf G' podrozdělením obou hran e, f novými vrcholy v_e, v_f. Je zřejmé, že i G' je vrcholově 2-souvislý graf, takže podle Věty 2.6 existují v G' dvě disjunktní cesty spojující v_e s v_f, tvořící spolu kružnici C'. Nakonec C' indukuje v G kružnici C procházející e i f. □ □

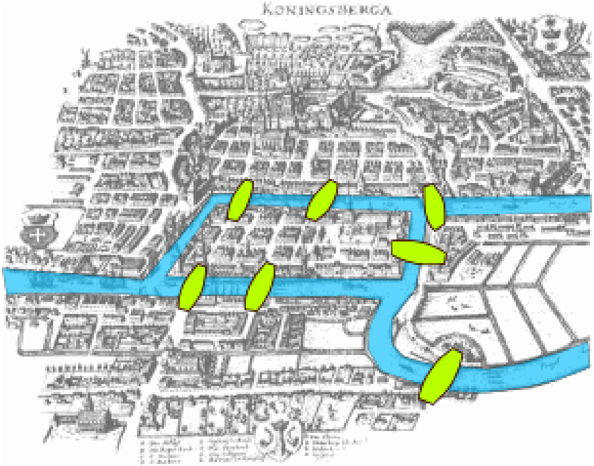
Rozšířením předchozí úvahy lze dokonce dokázat:

Věta 2.8. *Nechť G je vrcholově k-souvislý graf, $k \geq 1$. Pak pro každé dvě disjunktní množiny $U_1, U_2 \subset V(G)$, $|U_1| = |U_2| = k$ v G existuje k po dvou disjunktních cest z vrcholů U_1 do vrcholů U_2.*



2.4 Jedním tahem: Eulerovské grafy

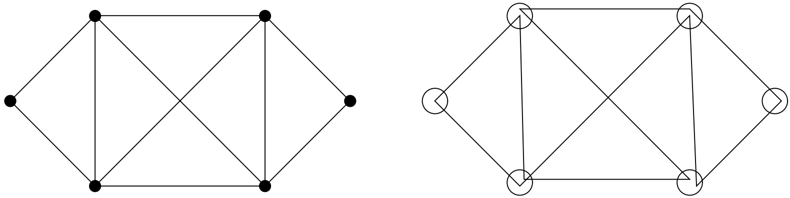
Snad *nejstarší výsledek teorie grafů* vůbec pochází od Leonarda Eulera – jedná se o slavných 7 mostů v Královci / Königsbergu / dnešním Kaliningradě.



O jaký problém se tehdy jednalo? Městští radní chtěli vědět, zda mohou suchou nohou přejít po každém ze sedmi vyznačených mostů právě jednou.

Rozbor tohoto problému vede k následující definici a odpovědi.

Definice: *Tah* je sled v grafu bez opakování hran.
Uzavřený tah je tahem, který končí ve vrcholu, ve kterém začal. *Otevřený tah* je tahem, který končí v jiném vrcholu, než ve kterém začal.



Nejstarší výsledek teorie grafů od Leonarda Eulera poté zní: □

Věta 2.9. *Graf G lze nakreslit jedním uzavřeným tahem právě když G je souvislý a všechny vrcholy v G jsou sudého stupně.* □

Důsledek 2.10. *Graf G lze nakreslit jedním otevřeným tahem právě když G je souvislý a všechny vrcholy v G až na dva jsou sudého stupně.*

Důkaz: Dokazujeme oba směry ekvivalence. Pokud lze G nakreslit jedním uzavřeným tahem, tak je zřejmě souvislý a navíc má každý stupeň sudý, neboť uzavřený tah každým průchodem vrcholem „ubere“ dvě hrany. □

Naopak zvolíme mezi všemi uzavřenými tahy T v G ten (jeden z) nejdelší. Tvrdíme, že T obsahuje všechny hrany grafu G.

- Pro spor vezměme graf $G' = G - E(T)$, o kterém předpokládejme, že je neprázdný. Jelikož G' má taktéž všechny stupně sudé, je (z indukčního předpokladu) libovolná jeho komponenta $C \subseteq G'$ nakreslená jedním uzavřeným tahem T_C. □
- Vzhledem k souvislosti grafu G každá komponenta $C \subseteq G'$ protíná náš tah T v některém vrcholе w, a tudíž lze oba tahy T_C a T „propojit přes w“. To je spor s naším předpokladem nejdelšího možného T. □

□

Důkaz důsledku: Nechť u, v jsou dva vrcholy grafu G mající lichý stupeň, neboli dva (předpokládané) konce otevřeného tahu pro G. Do G nyní přidáme nový vrchol w spojený hranami s u a v. Tím jsme náš případ převedli na předchozí případ grafu se všemi sudými stupni. □