

Teorie grafů

Petr Hanuš (Píta)

BR Solutions - Orlicky 2010

23.2. – 27.2.2010



Pojem grafu

Graf je abstraktní pojem matematiky a informatiky užitečný pro modelování reálných objektů či situací. (**Pozor, nepletme si graf s grafem funkce!**)

Intuitivně se graf skládá z:

- **vrcholů** (uzlů) - znázorňují se jako „body“,
- **hran** spojujících vrcholy.

Co lze reprezentovat grafem:

- počítačové sítě,
- mapu měst a silničního spojení,
- atomy v molekule a jejich vazby,
- vodovodní sítě,
- ...



Definice grafu

Graf je uspořádaná dvojice $G = (V, E)$, kde

- V je množina **vrcholů** a
- E je množina **hran** – množina vybraných dvouprvkových podmnožin množiny vrcholů.

Hranu mezi vrcholy u a v píšeme jako $\{u, v\}$, nebo zkráceně **uv** .

Vrcholy spojené hranou jsou **sousední**. Na množinu vrcholů známého grafu G odkazujeme jako na **$V(G)$** , na množinu hran **$E(G)$** .



- Výčtem vrcholů a výčtem hran:

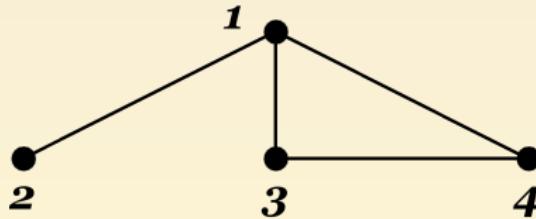
$$V = \{1, 2, 3, 4\}, \quad E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 4\}\}$$



- Výčtem vrcholů a výčtem hran:

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

- Obrázkem



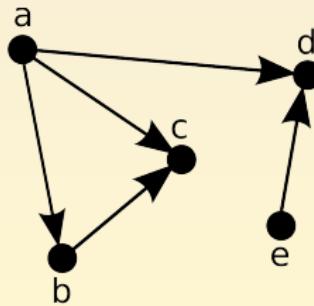
Orientovaný graf

V některých případech (například u toků v sítích) je potřeba u každé hrany vyjádřit její směr. Hrany i grafy se pak nazývají orientované.

Formální definice

Graf, jehož hrany jsou uspořádané dvojice vrcholů, se nazývá orientovaný.

O hraně (u, v) říkáme, že vychází z vrcholu u a vstupuje do v . Graficky se orientace hrany značí šipkou ve směru, kterým hrana vede.

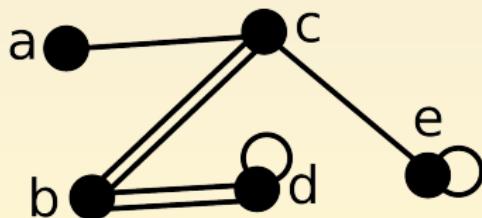


Multigraf

Definice grafu povoluje nejvýše 1 hranu mezi každou dvojicí vrcholů a požaduje, aby hrana spojovala různé vrcholy. Tato omezení odstraňuje multigraf:

Formální definice

Multigraf je graf, jenž nahrazuje množinu hran multimnožinou (smí obsahovat násobné prvky) a umožňuje existenci **smyček** - hran spojujících vrchol sám ze sebou.

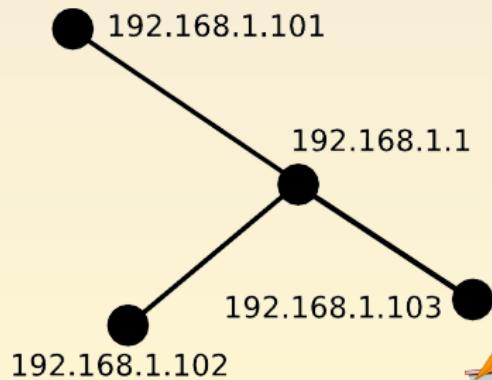
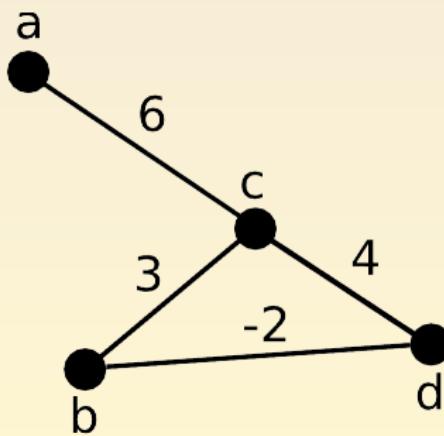


Ohodnocení grafu

Vrcholům a hranám je možné přiřadit např. číslo či barvu.

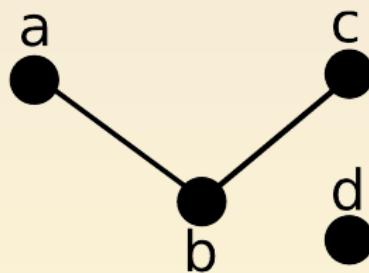
Definice

Přiřazení prvků konečné množiny vrcholům či hranám grafu nazýváme jejich ohodnocením.



Definice

Stupněm vrcholu u v neorientovaném grafu G nazýváme počet hran incidentních k vrcholu a značíme jej $d_G(u)$.



$$d_G(a) = d_G(c) = 1$$

$$d_G(b) = 2$$

$$d_G(d) = 0$$

- Nejvyšší stupeň v grafu G značíme $\Delta(G)$ a nejnižší $\delta(G)$.
- Součet stupňů v grafu je vždy sudý, roven dvojnásobku počtu hran.



Sestrojitelnost grafu na základě stupňů

Věta

Nechť $d_1 \leq d_2 \leq \dots \leq d_n$ je posloupnost přirozených čísel. Pak existuje graf s n vrcholy stupňů

$$d_1, d_2, \dots, d_n$$

právě tehdy, když existuje graf s $n - 1$ vrcholy stupňů

$$d_1, d_2, \dots, d_{n-d_n-1}, d_{n-d_n} - 1, \dots, d_{n-2} - 1, d_{n-1} - 1.$$

Komentář: Ze seřazené posloupnosti odebereme poslední (největší) stupeň d_n a od tolika d_n bezprostředně předchozích stupňů odečteme jedničky. Zbylé stupně na začátku posloupnosti se nezmění.

Při opětovné aplikaci posloupnost znova seřadit!



Mějme graf se stupni vrcholů $(1, 1, 1, 2, 3, 4)$

Dle předchozí věty upravíme posloupnost na $(1, 0, 0, 1, 2)$, uspořádáme ji $(0, 0, 1, 1, 2)$ a pak znovu vše zopakujeme. Dostáváme posloupnost $(0, 0, 0, 0)$ a takový graf už jednoznačně existuje.



Mějme graf se stupni vrcholů $(1, 1, 1, 2, 3, 4)$

Dle předchozí věty upravíme posloupnost na $(1, 0, 0, 1, 2)$, uspořádáme ji $(0, 0, 1, 1, 2)$ a pak znovu vše zopakujeme. Dostáváme posloupnost $(0, 0, 0, 0)$ a takový graf už jednoznačně existuje.

Mějme graf se stupni vrcholů $(1, 1, 1, 1, 2, 3, 4, 6, 7)$

Tuto posloupnost upravíme na $(1, 0, 0, 0, 1, 2, 3, 5)$ a uspořádáme ji $(0, 0, 0, 1, 1, 2, 3, 5)$, pak znovu upravíme na $(0, 0, -1, 0, 0, 1, 2)$. Získáváme situaci, kdy stupně v grafu jsou záporné, ale to pochopitelně nelze. Proto takový graf neexistuje.

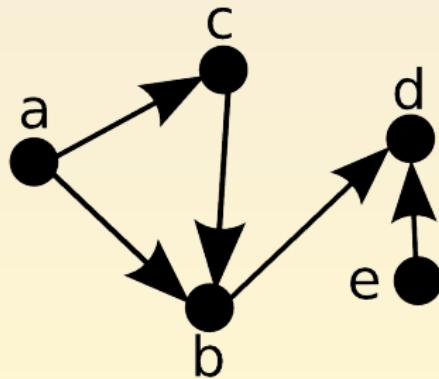


Stupeň vrcholu v orientovaném grafu

V případě orientovaného grafu rozlišujeme **vstupní** a **výstupní** stupeň.

Definice

Vstupním (výstupním) stupněm vrcholu u orientovaného grafu nazýváme počet hran vstupujících do, resp. vycházejících z, vrcholu u a značíme jej $d_G^+(u)$, resp. $d_G^-(u)$.



vrchol	d_G^-	d_G^+
a	2	0
b	1	2
c	1	1
d	0	2
e	1	0



Definice

Sledem v (neorientovaném) grafu nazýváme posloupnost vrcholů a hran

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

kde každá hrana e_i spojuje vrcholy v_{i-1} , v_i , resp. vede z v_{i-1} do v_i .

- Sled v grafu je tedy „trasou“, na které se mohou vrcholy i hrany opakovat.
- Samostatný vrchol je také sledem.
- Délku sledu měříme počtem hran v této posloupnosti.



Definice

Cesta v grafu je sled bez opakování vrcholů.

Jak je to s opakováním hran?

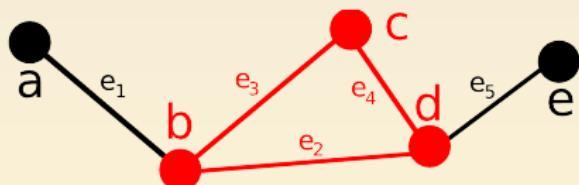


Cesty v grafu

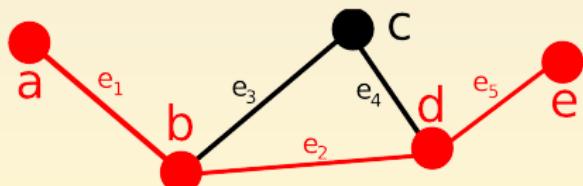
Definice

Cesta v grafu je sled bez opakování vrcholů.

Jak je to s opakováním hran?



$b, e_3, c, e_4, d, e_2, b, e_5, e$ je sledem,
ale ne cestou – b se opakuje



$a, e_1, b, e_2, d, e_5, e$ je sledem i
cestou

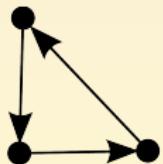


Souvislost grafu

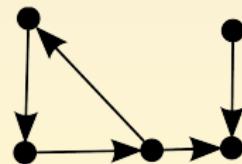
Definice

Neorientovaný graf se nazývá souvislý, pokud mezi každými dvěma jeho vrcholy vede cesta.

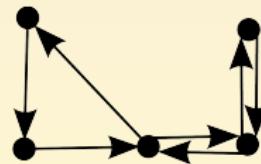
- Nahradíme-li všechny hrany orientovaného grafu G neorientovanými a získáme-li tak souvislý graf, G je **slabě souvislý**.
- Orientovaný graf je **silně souvislý**, pokud mezi každými dvěma jeho vrcholy vedou cesty v obou směrech.



Nesouvislý



Slabě souvislý



Silně souvislý

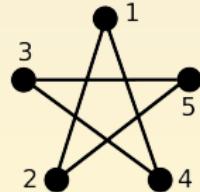
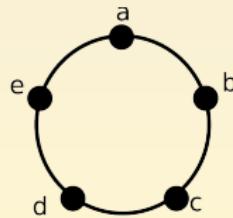


Isomorfismus grafů I

Grafy, které se liší např. nakreslením, označením vrcholů a hran, ohodnocením, se nemusí lišit svou strukturou – mohou být **isomorfní**.

Formální definice

Isomorfismus mezi grafy G, H je bijektivní zobrazení vrcholů, které zachovává hrany – tj. pokud vede v grafu G hrana mezi vrcholy u, v , pak v grafu H vede hrana mezi vrcholy $f(u), f(v)$. Pokud mezi grafy G, H existuje isomorfismus, nazývají se isomorfní.



- Co musejí isomorfní grafy splňovat?



- Co musejí isomorfní grafy splňovat?
 - Mít stejný počet vrcholů.



- Co musejí isomorfní grafy splňovat?
 - Mít stejný počet vrcholů.
 - Mít stejný počet hran.



- Co musejí isomorfní grafy splňovat?

- Mít stejný počet vrcholů.
- Mít stejný počet hran.
- Mít stejně stupně vrcholů.



- Co musejí isomorfní grafy splňovat?
 - Mít stejný počet vrcholů.
 - Mít stejný počet hran.
 - Mít stejně stupně vrcholů.
- Důkaz, že grafy isomorfní nejsou, se provádí pomocí těchto (a dalších!) invariantů. Jsou-li tyto invarianty shodné, je nutno vyloučit všechny možné isomorfismy.

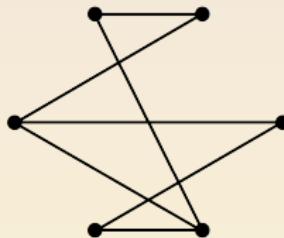
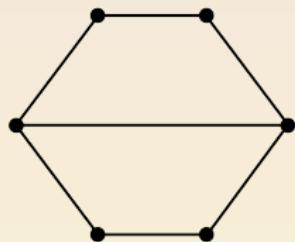


- Co musejí isomorfní grafy splňovat?
 - Mít stejný počet vrcholů.
 - Mít stejný počet hran.
 - Mít stejně stupně vrcholů.
- Důkaz, že grafy isomorfní nejsou, se provádí pomocí těchto (a dalších!) invariantů. Jsou-li tyto invarianty shodné, je nutno vyloučit všechny možné isomorfismy.
- Důkaz isomorfie dvou grafů vyžaduje přímo nalezení konkrétního isomorfismu mezi těmito grafy.



Isomorfismus grafů – příklad 1

Jsou následující dva grafy isomorfní?

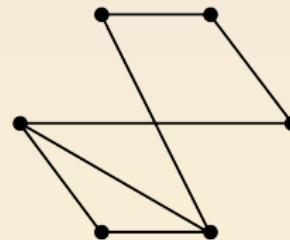
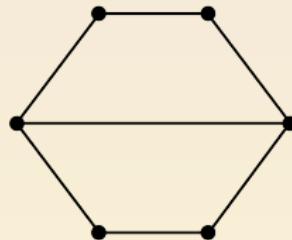


Nejprve se podíváme, zda mají stejný počet vrcholů a hran – mají. Pak se podíváme na stupně vrcholů a zjistíme, že oba mají stejnou posloupnost stupňů $2, 2, 2, 2, 3, 3$. Mohou tedy být (**ale nemusejí!**) isomorfní. Dále je tedy třeba zkoušet všechny možnosti zobrazení isomorfismu z levého grafu do pravého. Oba vrcholy stupně tři jsou si symetrické, můžeme tedy říct, že nejlevější vrchol prvního grafu se zobrazí na nejlevější vrchol v druhém grafu. Druhý vrchol stupně tři se musí zobrazit na analogický vrchol druhého grafu (pravý spodní). Pak je již patrné, že další sousedé se zobrazí na analogické sousedy.



Isomorfismus grafů – příklad 2

Jsou následující dva grafy isomorfní?



Jako v předchozím příkladě si ověříme, že oba grafy mají stejný počet vrcholů a hran i stejnou posloupnost stupňů $2, 2, 2, 2, 3, 3$. Všimněme si však, že v druhém grafu oba vrcholy stupně tří mají svého společného souseda, tvoří s ním trojúhelník. V prvním grafu tomu tak není, první graf dokonce nemá žádný trojúhelník. Proto zadané dva grafy nejsou isomorfní.

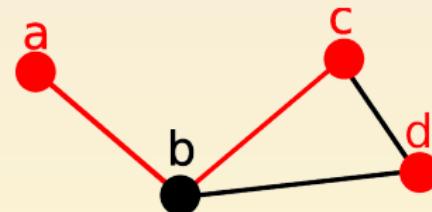
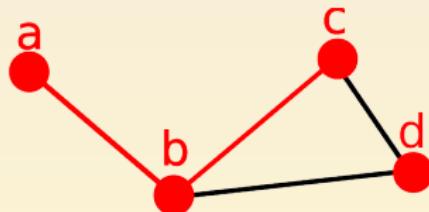


Podgraf

Graf H je **podgrafem** grafu G , pokud platí následující podmínky:

- Vrcholy grafu H tvoří podmnožinu vrcholů grafu G .
- Hrany grafu H tvoří podmnožinu hran grafu G .
- Hrany grafu H mají oba vrcholy v H .

Graf G je poté **nadgrafem** grafu H .



Vyznačené vrcholy a hrany v levém grafu tvoří podgraf,
v pravém grafu nikoliv.

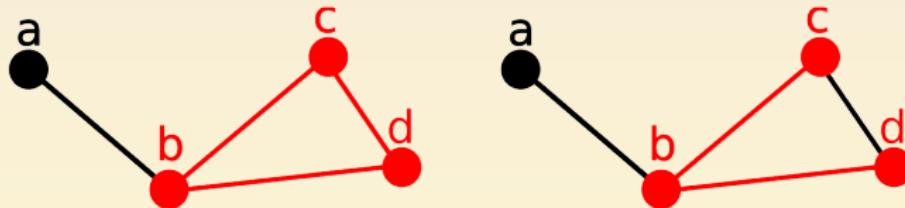


Indukovaný podgraf

Definice

Podgraf H se nazývá **indukovaný**, pokud obsahuje všechny hrany, které mezi jeho vrcholy vedou v nadřazeném grafu G .

Samostatné vrcholy libovolného grafu tvoří jeho podgraf.



Vyznačené vrcholy a hrany v levém grafu tvoří indukovaný podgraf, v pravém grafu pouze obecný podgraf.



Mějme jednoduchý graf G na n vrcholech a značme vrcholy jednoduše čísly $V(G) = \{0, 1, \dots, n - 1\}$. Jak efektivně uložit graf např. v paměti počítače?

• Matice sousednosti

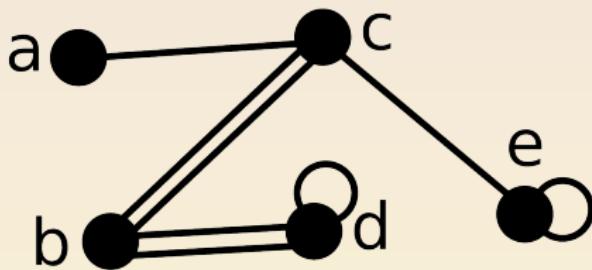
- Matice M o rozměrech $n \times n$ pro n vrcholů grafu
- $M_{ij} = 1$ pokud hrana (i, j) patří do grafu
- $M_{ij} = 0$ jinak
- Pro neorientovaný graf je M symetrická, pro orientovaný nemusí

• Seznam sousedů

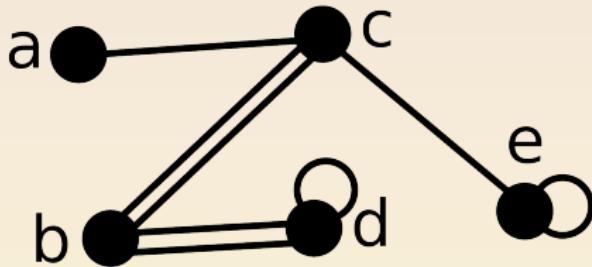
- Pro každý vrchol existuje samostatný seznam sousedů, s nimiž je tento spojen hranou
- Implementace pomocí dvou jednorozměrných polí
- V prvním jsou uloženy všechny seznamy za sebou, seřazené podle čísla vrcholu
- Druhé uchovává indexy, na kterých začínají v prvním seznamu sousedé každého vrcholu



Implementace grafu – příklad



Implementace grafu – příklad

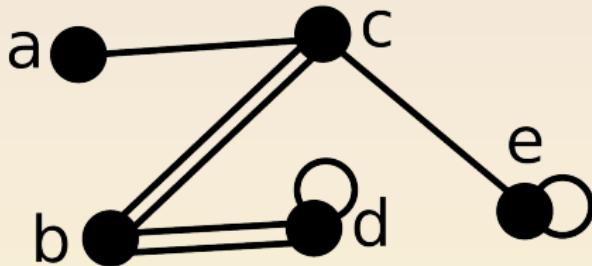


Matice sousednosti

	a	b	c	d	e
a	0	0	1	0	0
b	0	0	2	2	0
c	1	2	0	0	1
d	0	2	0	1	0
e	0	0	1	0	1



Implementace grafu – příklad



Matice sousednosti

	a	b	c	d	e
a	0	0	1	0	0
b	0	0	2	2	0
c	1	2	0	0	1
d	0	2	0	1	0
e	0	0	1	0	1

Seznam sousedů

Pole indexů do seznamu sousedů:

a	b	c	d	e
1	2	6	10	13

Seznam sousedů:

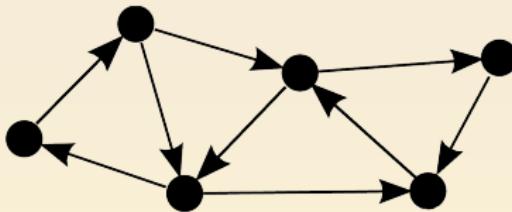
1	2	3	4	5	6	7	8	9	10	11	12	13	14
c	c	c	d	d	a	b	b	e	b	b	d	c	e



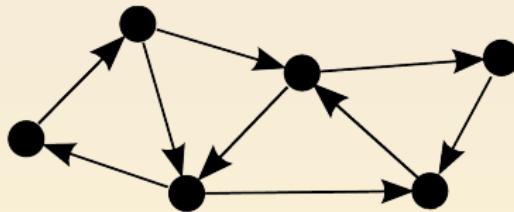
Bodované úlohy



- Je následující graf souvislý? Pokud ano, je souvislý slabě či silně?
Své tvrzení zdůvodněte.



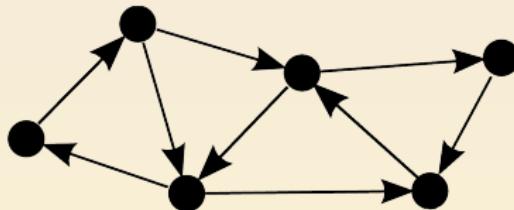
- Je následující graf souvislý? Pokud ano, je souvislý slabě či silně?
Své tvrzení zdůvodněte.



- Existuje graf s posloupností stupňů $1, 1, 3, 3, 3, 4, 4, 4, 6, 6, 6, 7$?
Své tvrzení dokažte.



- Je následující graf souvislý? Pokud ano, je souvislý slabě či silně?
Své tvrzení zdůvodněte.

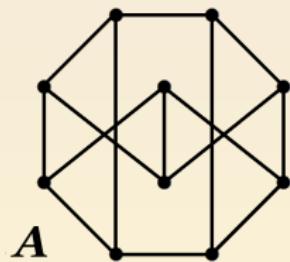


- Existuje graf s posloupností stupňů 1, 1, 3, 3, 3, 4, 4, 4, 6, 6, 6, 7?
Své tvrzení dokažte.
- Nakreslete všechny neisomorfní (jednoduché) grafy na **třech** vrcholech.

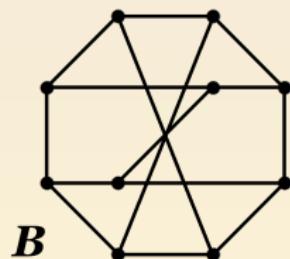


Bodované úlohy II

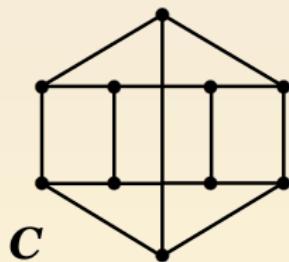
- Najděte všechny isomorfní dvojice grafů v následujících obrázcích tří 10-vrcholových grafů.



A



B

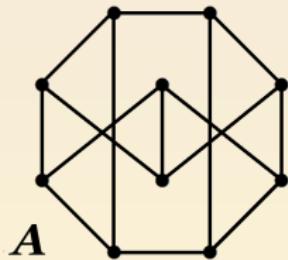


C

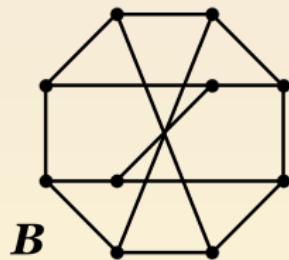


Bodované úlohy II

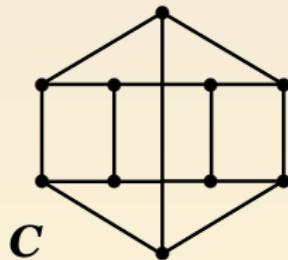
- Najděte všechny isomorfní dvojice grafů v následujících obrázcích tří 10-vrcholových grafů.



A



B



C

- Mohou dva grafy, orientovaný a neorientovaný, mít stejné matice sousednosti a zároveň stejný počty hran? Pokud ano, jak takové grafy vypadají?



Definice

Kružnice (uzavřená cesta) v grafu je netriviální neorientovaná cesta, jež začíná i končí ve stejném vrcholu. **Cyklus** je orientovaná (z orientovaných hran složená) kružnice respektující orientaci těchto hran.

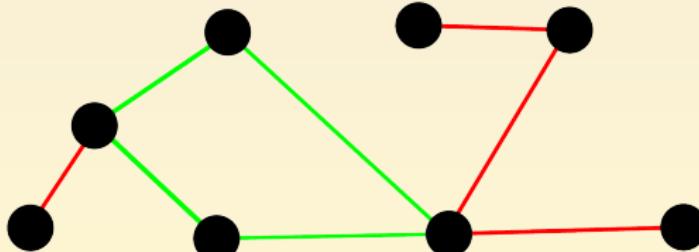
- Cyklus je tedy vždy kružnicí, ale každá kružnice nemusí být vždy cyklem.
- Zákaz opakování vrcholu znemožnuje využít násobných hran multigrafu s výjimkou kružnice na 2 vrcholech.
- Samostatný vrchol, který je cestou, za cyklus nepovažujeme.
- Graf, který obsahuje cyklus, se nazývá **cyklický**. Pokud graf cyklus neobsahuje, nazýváme jej **acyklický**.



Definice

Pokud je hrana v grafu součástí nějakého cyklu, nazývá se **cyklická**.

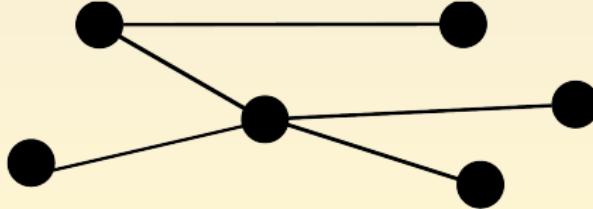
- Odstraníme-li ze souvislého grafu G hranu e , pak vzniklý graf G' bude souvislý právě tehdy, když e je cyklická.
- Na obrázku dole jsou cyklické hrany vyznačeny zeleně. Ostatní, které by způsobily rozpad grafu jsou vyznačeny červeně.



Definice

Les je graf, který neobsahuje kružnice. **Strom** je graf, který neobsahuje kružnice a je souvislý.

- Strom je tedy souvislý les.
- Na obrázku je jednoduchý příklad stromu.



- Každý strom s alespoň 1 hranou obsahuje nejméně 2 vrcholy stupně 1.



- Každý strom s alespoň 1 hranou obsahuje nejméně 2 vrcholy stupně 1.
- Strom s n vrcholy obsahuje práve $n - 1$ hran.



- Každý strom s alespoň 1 hranou obsahuje nejméně 2 vrcholy stupně 1.
- Strom s n vrcholy obsahuje práve $n - 1$ hran.
- Předchozí větu lze aplikovat také na lesy. Platí, že les o n vrcholech a k komponentách má $n - k$ hran.



- Každý strom s alespoň 1 hranou obsahuje nejméně 2 vrcholy stupně 1.
- Strom s n vrcholy obsahuje práve $n - 1$ hran.
- Předchozí větu lze aplikovat také na lesy. Platí, že les o n vrcholech a k komponentách má $n - k$ hran.
- Mezi každými 2 vrcholy ve stromu vede právě jedna cesta.
(Jelikož strom nemůže obsahovat kružnici.)



- Každý strom s alespoň 1 hranou obsahuje nejméně 2 vrcholy stupně 1.
- Strom s n vrcholy obsahuje práve $n - 1$ hran.
- Předchozí větu lze aplikovat také na lesy. Platí, že les o n vrcholech a k komponentách má $n - k$ hran.
- Mezi každými 2 vrcholy ve stromu vede právě jedna cesta.
(Jelikož strom nemůže obsahovat kružnici.)
- Přidáním libovolné hrany do stromu vznikne právě jedna kružnice.



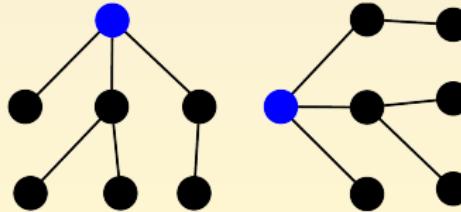
Kořenový strom

Strom, jehož hrany jsou orientované, se nazývá také orientovaný.

Definice

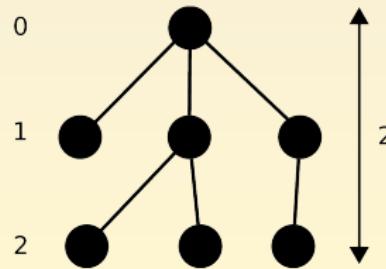
Orientovaný strom, jenž má určen význačný vrchol (**kořen**) r , a v němž existuje orientovaná cesta z r do všech ostatních vrcholů, se nazývá **kořenový strom**.

- Při kreslení kořenových grafů se vymezují šipky definující orientaci hran, předpokládá se, že směřují od kořene.
- Vzhledem k absenci cyklů je interpretace jednoznačná.



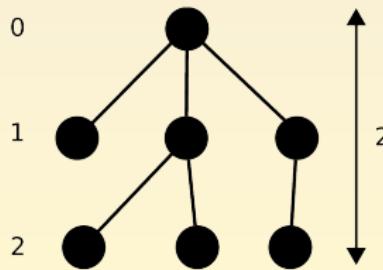
Vlastnosti kořenových stromů

- Kořen (jako jediný z vrcholů) má vstupní stupeň 0 a všechny ostatní vrcholy 1.



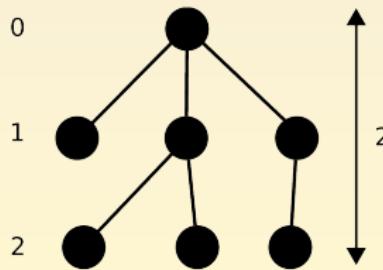
Vlastnosti kořenových stromů

- Kořen (jako jediný z vrcholů) má vstupní stupeň 0 a všechny ostatní vrcholy 1.
- Vzdálenost (počet hran na cestě) vrcholu od kořene stromu se nazývá **hloubka** či **úroveň vrcholu**.



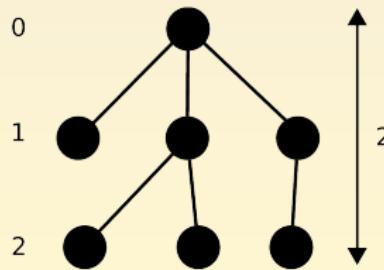
Vlastnosti kořenových stromů

- Kořen (jako jediný z vrcholů) má vstupní stupeň 0 a všechny ostatní vrcholy 1.
- Vzdálenost (počet hran na cestě) vrcholu od kořene stromu se nazývá **hloubka** či **úroveň vrcholu**.
- Hloubka kořene je rovna 0.



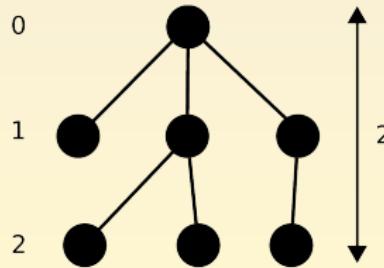
Vlastnosti kořenových stromů

- Kořen (jako jediný z vrcholů) má vstupní stupeň 0 a všechny ostatní vrcholy 1.
- Vzdálenost (počet hran na cestě) vrcholu od kořene stromu se nazývá **hloubka** či **úroveň vrcholu**.
- Hloubka kořene je rovna 0.
- Je zvykem kreslit vrcholy jedné úrovně ve stejné výšce.



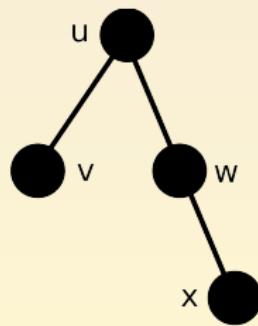
Vlastnosti kořenových stromů

- Kořen (jako jediný z vrcholů) má vstupní stupeň 0 a všechny ostatní vrcholy 1.
- Vzdálenost (počet hran na cestě) vrcholu od kořene stromu se nazývá **hloubka** či **úroveň vrcholu**.
- Hloubka kořene je rovna 0.
- Je zvykem kreslit vrcholy jedné úrovně ve stejné výšce.
- **Výškou kořenového stromu** označujeme nejvyšší z hloubek všech jeho vrcholů.



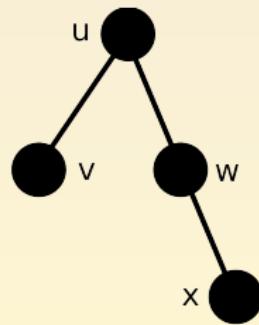
Vlastnosti kořenových stromů II

- Vede-li v kořenovém stromu hrana z vrcholu u do v , nazývá se u rodičem (otcem) v a v potomkem (synem).



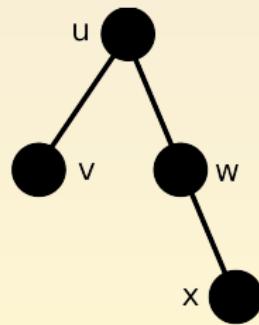
Vlastnosti kořenových stromů II

- Vede-li v kořenovém stromu hrana z vrcholu u do v , nazývá se u **rodičem (otcem)** v a v **potomkem (synem)**.
- Vrcholy mající společného rodiče nazýváme **sourozenci**.



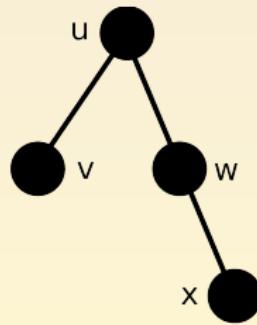
Vlastnosti kořenových stromů II

- Vede-li v kořenovém stromu hrana z vrcholu u do v , nazývá se u **rodičem (otcem)** v a v **potomkem (synem)**.
- Vrcholy mající společného rodiče nazýváme **sourozenci**.
- Vrchol, jenž nemá žádné potomky, nazýváme **list** stromu.



Vlastnosti kořenových stromů II

- Vede-li v kořenovém stromu hrana z vrcholu u do v , nazývá se u **rodičem (otcem) v** a v **potomkem (synem)**.
- Vrcholy mající společného rodiče nazýváme **sourozenci**.
- Vrchol, jenž nemá žádné potomky, nazýváme **list** stromu.
- Vrcholy, které nejsou listy, označujeme jako **vnitřní**.

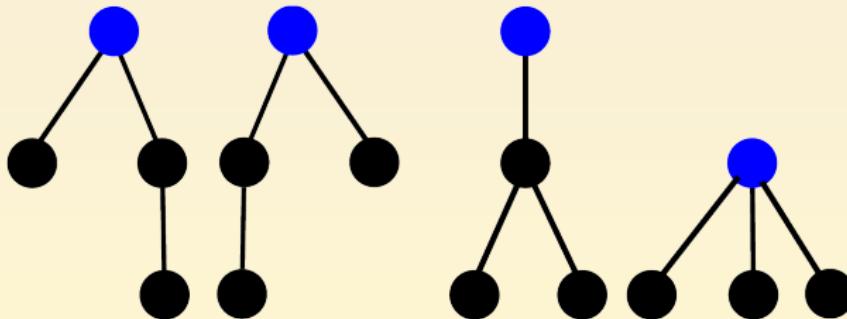


Isomorfismus kořenových stromů

Definice

Kořenové stromy považujeme za isomorfní, pokud mezi nimi existuje isomorfismus, který zobrazí kořen stromu na kořen.

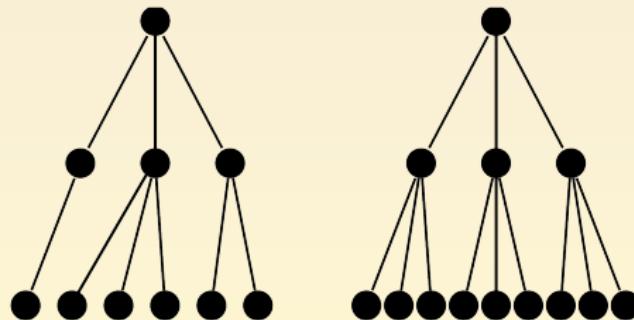
Dvojice grafů vlevo jsou isomorfní kořenové stromy. Dva pravé grafy jsou isomorfní stromy, ale ne isomorfní kořenové stromy (neexistuje isomorfismus zobrazující kořen na kořen).



Definice

Kořenový strom, jehož každý vrchol má nejvýše n potomků, se nazývá **n -árni**. Má-li navíc n -árni strom právě n potomků u každého vnitřního vrcholu a všechny listy stejné hloubky, nazývá se **úplný n -árni** strom.

Na obrázku je levý strom 3-árni (ternární), pravý je úplný ternární.



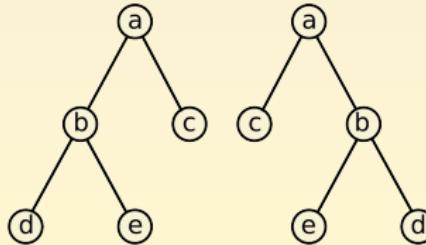
Uspořádané stromy

V některých případech může být výhodné potomky každého vrcholu jednoznačně pojmenovat a seřadit:

Definice

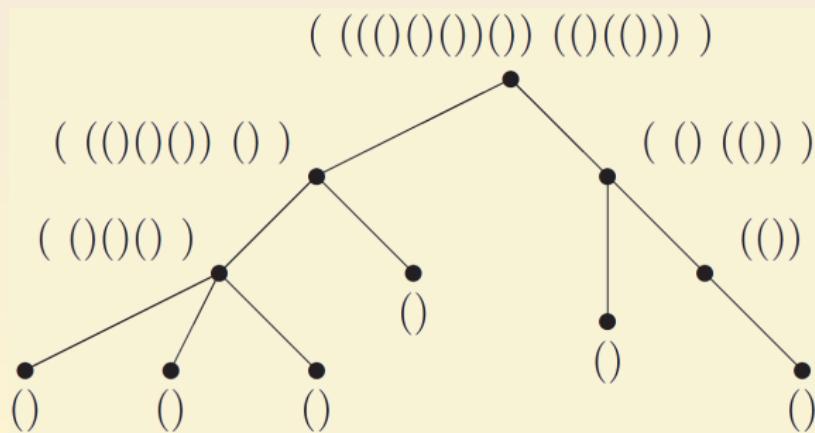
Uspořádaný strom je kořenový strom s daným pořadím potomků každého vrcholu.

- Uspořádaný kořenový strom se někdy také nazývá **pěstovaný strom**.
- Při kreslení uspořádaného grafu se dané pořadí vrcholů dodržuje ve směru zleva doprava.



Kódování uspořádaných kořenových stromů

Kód uspořádaného kořenového stromu se spočítá rekuzivně z kódů všech podstromů kořene, seřazených v daném pořadí a uzavřených do páru závorek.



Místo znaků (a) lze použít i jiné symboly, třeba 0 a 1.



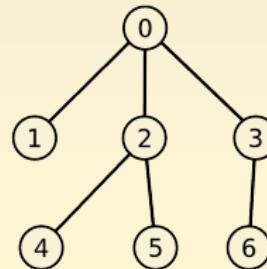
Binární stromy

Nejpoužívanější n -ární stromy

Definice

Uspořádaný 2-ární strom se nazývá **binární**. Potomci každého vrcholu jsou označováni jako **levý** a **pravý**.

- Každý kořenový strom lze převést na binární.
- Úplný binární strom výšky h má právě $2^{h+1} - 1$ vrcholů.
- Binární stromy lze velmi efektivně reprezentovat **polem rodičů**, v němž je pro každý vrchol uložen pouze název jeho rodiče.



Pole rodičů: 0 0 0 2 2 3

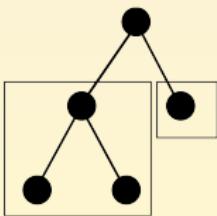


Podstromy binárních stromů

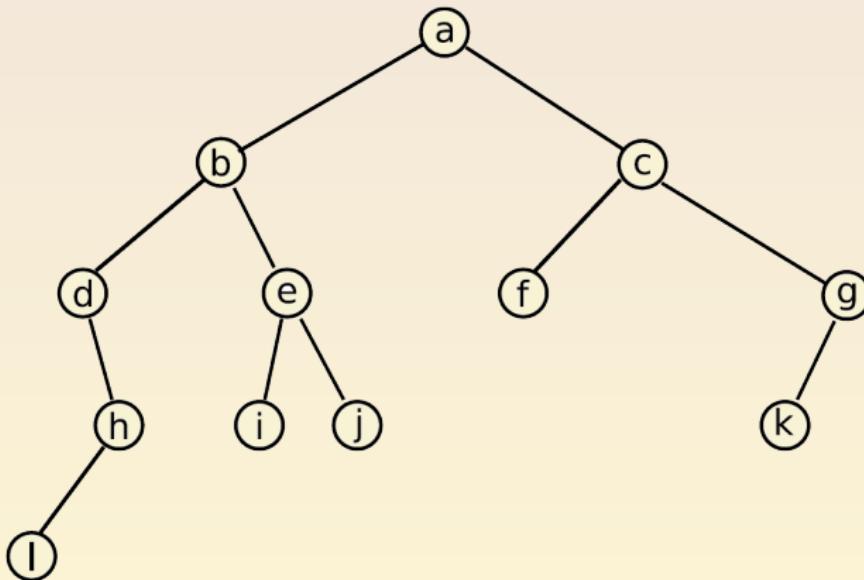
Definice

Indukovaný podgraf binárního stromu G , tvořený jedním potomkem vrcholu v a všemi jeho následníky, se nazývá **podstromem** vrcholu v a stromu G .

- Podstrom binárního stromu je také binárním stromem.
- Podstromy se pro konkrétní vrchol označují jako levý a pravý, přičemž kořenem takového podstromu je levý (resp.) pravý potomek daného vrcholu.
- Pravý a levý podstrom binárního stromu o výšce h mají výšku nejvýše $h - 1$, přičemž nejméně jeden z nich má výšku právě $h - 1$.



Průchod binárním stromem po úrovních

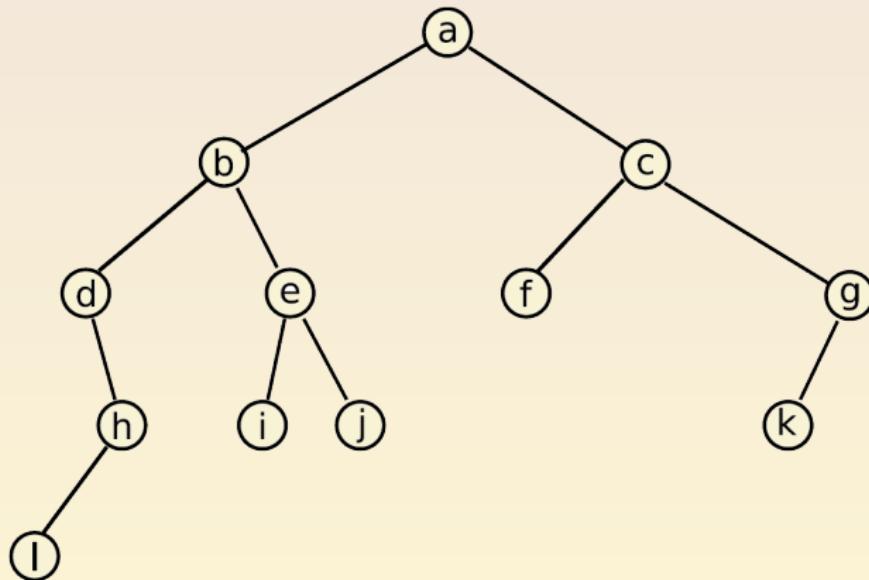


Vrcholy stromu jsou navštíveny v pořadí:

a, b, c, d, e, f, g, h, i, j, k, l



Průchod binárním stromem po podstromech



Vrcholy stromu jsou navštíveny v pořadí:

a, b, d, h, l, e, i, j, c, f, g, k



Bodované úlohy



Bodované úlohy

- Najdete graf se dvěmi kružnicemi, z něhož lze odebráním jedné hrany vytvořit strom? Zdůvodněte a případně nakreslete.



Bodované úlohy

- Najdete graf se dvěmi kružnicemi, z něhož lze odebráním jedné hrany vytvořit strom? Zdůvodněte a případně nakreslete.
- Kolik vznikne kružnic, přidáme-li ke stromu dvě hrany?



Bodované úlohy

- Najdete graf se dvěmi kružnicemi, z něhož lze odebráním jedné hrany vytvořit strom? Zdůvodněte a případně nakreslete.
- Kolik vznikne kružnic, přidáme-li ke stromu dvě hrany?
- Nakreslete pěstovaný strom odpovídající závorkovému kódu

$(((()((())(())(()))((()) .$

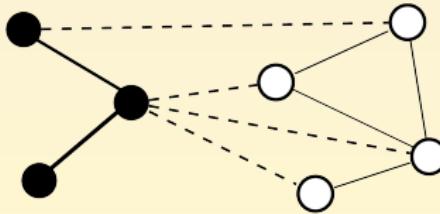


Strom v grafu

Definice

Stromem v grafu G rozumíme podgraf grafu G , který je stromem.

- Hrany a vrcholy, které do takového stromu patří, označujeme jako **stromové** (v opačném případě **nestromové**).
- Hranu, jejíž jeden krajní vrchol je součástí stromu T v neorient. grafu, budeme značit jako **okrajovou** hranu stromu T .
- Je-li graf orientovaný, značíme hranu jako okrajovou pokud je součástí stromu T její počáteční vrchol.
- Je-li G graf a T strom v G , potom graf vzniklý z T přidáním jeho libovolné okrajové hrany je také stromem.

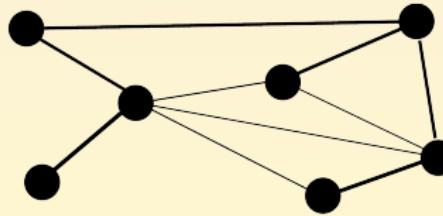


Kostra grafu

Definice

Kostra grafu G je takový strom T v grafu G , pro který platí $V(T) = V(G)$.

- Graf může mít více než jednu kostru.
- Každý acyklický podgraf grafu G je obsažen v alespoň jedné kostře grafu G .
- Graf má kostru pouze tehdy, je-li souvislý.
- Pokud souvislý není, uvažujeme kostry jeho komponent souvislosti. Kostra nesouvislého grafu je tedy lesem, nikoliv stromem.



- Průchod grafem do šířky



● Průchod grafem do šířky

- Vstupem je neorientovaný souvislý graf.



● Průchod grafem do šířky

- Vstupem je neorientovaný souvislý graf.
- Slouží k prohledání a navštívení všech vrcholů grafu.



● Průchod grafem do šířky

- Vstupem je neorientovaný souvislý graf.
- Slouží k prohledání a navštívení všech vrcholů grafu.
- Vrcholy jsou navštěvovány v pořadí podle vzdálenosti od počátečního vrcholu.



● Průchod grafem do šířky

- Vstupem je neorientovaný souvislý graf.
- Slouží k prohledání a navštívení všech vrcholů grafu.
- Vrcholy jsou navštěvovány v pořadí podle vzdálenosti od počátečního vrcholu.
- Nalezne nejkratší cestu z počátečního vrcholu do všech ostatních.



● Průchod grafem do šířky

- Vstupem je neorientovaný souvislý graf.
- Slouží k prohledání a navštívení všech vrcholů grafu.
- Vrcholy jsou navštěvovány v pořadí podle vzdálenosti od počátečního vrcholu.
- Nalezne nejkratší cestu z počátečního vrcholu do všech ostatních.
- Při průchodu grafem je budován strom cest do všech jeho vrcholů.



● Průchod grafem do šířky

- Vstupem je neorientovaný souvislý graf.
- Slouží k prohledání a navštívení všech vrcholů grafu.
- Vrcholy jsou navštěvovány v pořadí podle vzdálenosti od počátečního vrcholu.
- Nalezne nejkratší cestu z počátečního vrcholu do všech ostatních.
- Při průchodu grafem je budován strom cest do všech jeho vrcholů.

● Průchod grafem do hloubky



• Průchod grafem do šířky

- Vstupem je neorientovaný souvislý graf.
- Slouží k prohledání a navštívení všech vrcholů grafu.
- Vrcholy jsou navštěvovány v pořadí podle vzdálenosti od počátečního vrcholu.
- Nalezne nejkratší cestu z počátečního vrcholu do všech ostatních.
- Při průchodu grafem je budován strom cest do všech jeho vrcholů.

• Průchod grafem do hloubky

- Vstupem je rovněž neorientovaný souvislý graf



● Průchod grafem do šířky

- Vstupem je neorientovaný souvislý graf.
- Slouží k prohledání a navštívení všech vrcholů grafu.
- Vrcholy jsou navštěvovány v pořadí podle vzdálenosti od počátečního vrcholu.
- Nalezne nejkratší cestu z počátečního vrcholu do všech ostatních.
- Při průchodu grafem je budován strom cest do všech jeho vrcholů.

● Průchod grafem do hloubky

- Vstupem je rovněž neorientovaný souvislý graf
- Dokud je to možné, vybere vždy hranu vedoucí dále z vrcholu, do kterého právě vstoupil. Poté se vrací stromem ke kořenu.



● Průchod grafem do šířky

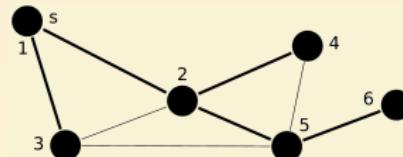
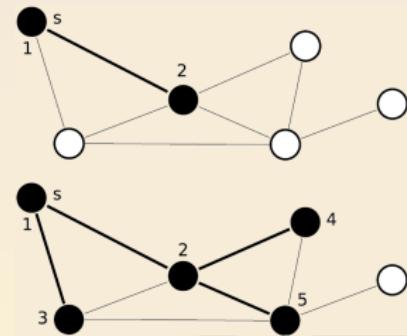
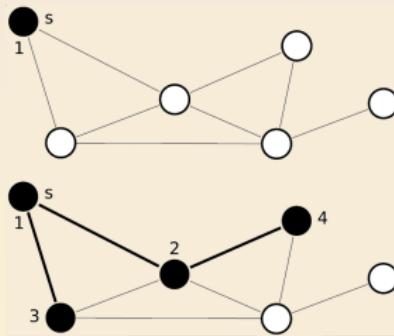
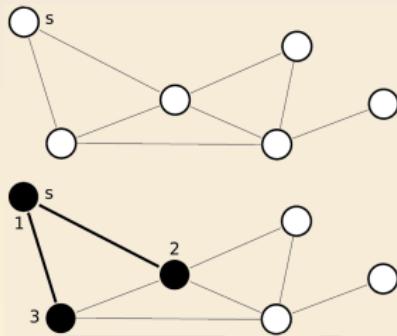
- Vstupem je neorientovaný souvislý graf.
- Slouží k prohledání a navštívení všech vrcholů grafu.
- Vrcholy jsou navštěvovány v pořadí podle vzdálenosti od počátečního vrcholu.
- Nalezne nejkratší cestu z počátečního vrcholu do všech ostatních.
- Při průchodu grafem je budován strom cest do všech jeho vrcholů.

● Průchod grafem do hloubky

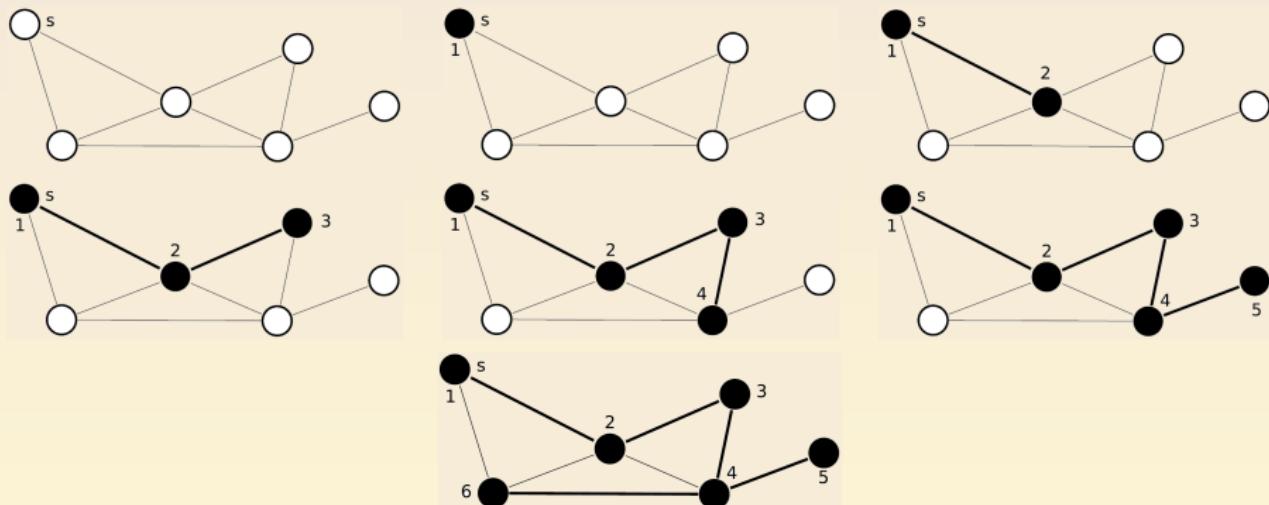
- Vstupem je rovněž neorientovaný souvislý graf
- Dokud je to možné, vybere vždy hranu vedoucí dále z vrcholu, do kterého právě vstoupil. Poté se vrací stromem ke kořenu.
- I pro tento algoritmus platí, že projde všemi vrcholy a nalezne do nich nejkratší cesty.



Průchod grafem do šířky – příklad



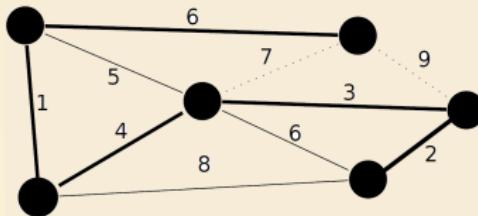
Průchod grafem do hloubky – příklad



Minimální kostra grafu

Definice

Nechť G je souvislý graf s ohodnocenými hranami. Kostra grafu G , jejíž součet ohodnocení všech hran je nejnižší, se nazývá **minimální kostra grafu G** .



Algoritmy pro hledání minimální kostry

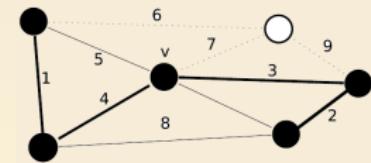
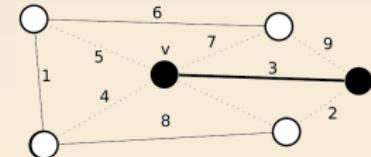
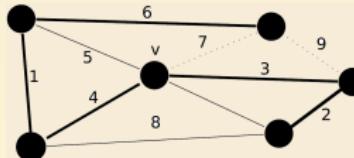
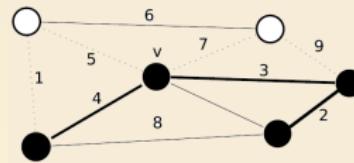
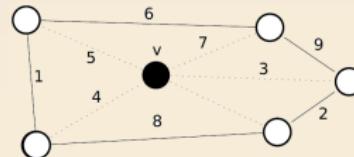
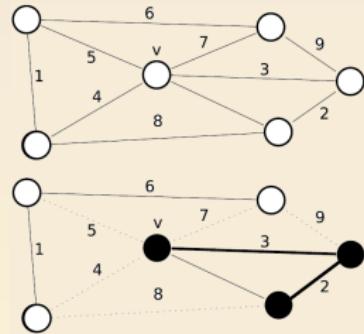
- Primův algoritmus
- Kruskalův algoritmus
- Borůvkův algoritmus



- Neprohledává systematicky všechny kostry grafu.
- Začíná v libovolném vrcholu a buduje strom.
- Nejvyšší prioritu mají hrany s nejnižším ohodnocením.
- Stále existuje jen jedna komponenta minimální kostry, která postupně roste.

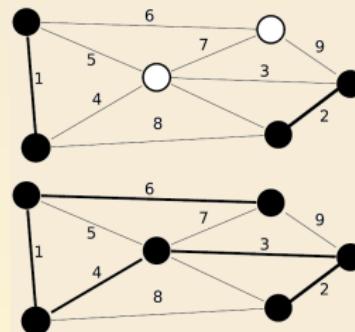
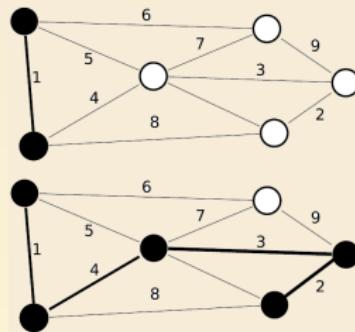
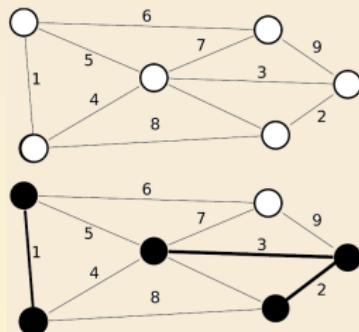


Primův algoritmus - příklad



Kruskalův algoritmus

- Nepostupuje cestou budování stromu, naopak vzniká les.
- Přidává hrany seřazené vzestupně podle jejich ohodnocení.

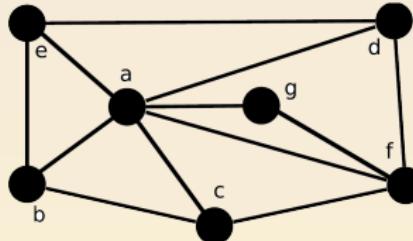


Bodované úlohy



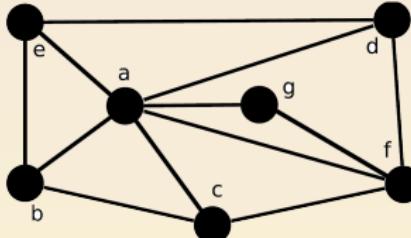
Bodované úlohy

- Jaký je průchod následujícího grafu do šírky, začínáme-li ve vrcholu b a priorita vrcholů na stejném úrovni je dáná abecedně?

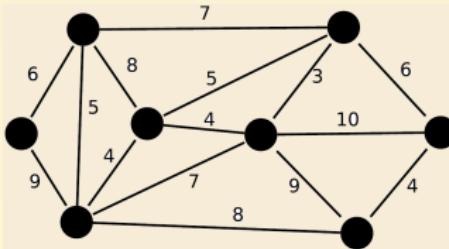


Bodované úlohy

- Jaký je průchod následujícího grafu do šírky, začínáme-li ve vrcholu b a priorita vrcholů na stejném úrovni je dáná abecedně?

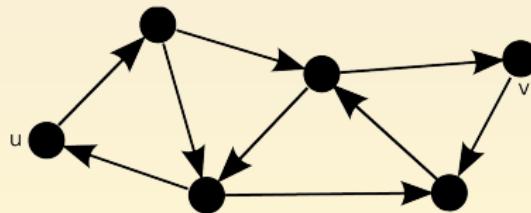


- Na graf na obrázku dole aplikujte některý z probraných algoritmů hledání minimální kostry.



Pro připomenutí:

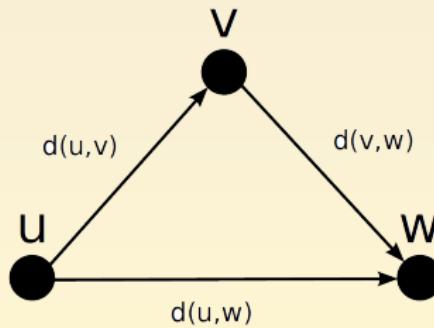
- Délka cesty v neohodnoceném grafu je rovna počtu hran na této cestě.
- Vzdálenost $\delta(u, v)$ vrcholů u, v je délka nejkratší cesty z u do v .
- Vzdálenost mezi dvěma vrcholy nemusí být v případě orientovaného grafu symetrická (viz obrázek).



Vzdálenost v ohodnoceném grafu

- Délka cesty v ohodnoceném grafu je rovna součtu ohodnocení hran na této cestě.
- Vzdálenost vrcholů je opět rovna délce nejkratší cesty.
- Vzdálenost v grafech splňuje trojúhelníkovou nerovnost:

$$\forall u, v, w \in V(G) : d_G(u, v) + d_G(v, w) \geq d_G(u, w).$$



● Dijkstrův algoritmus

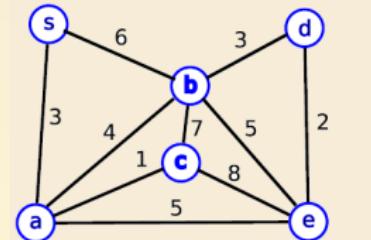
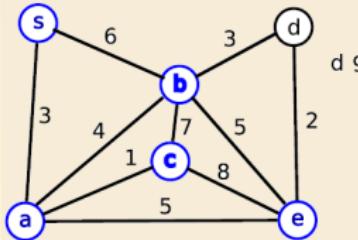
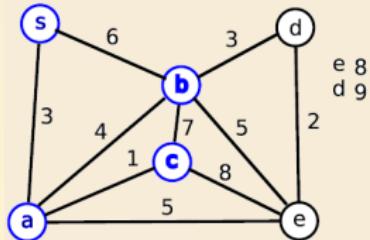
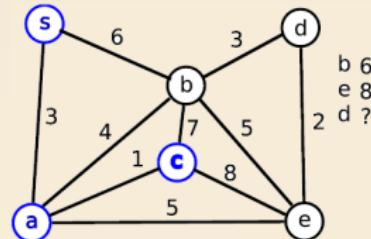
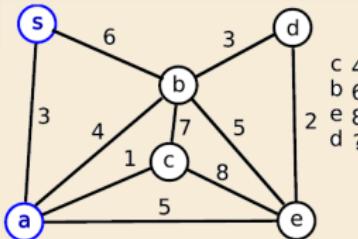
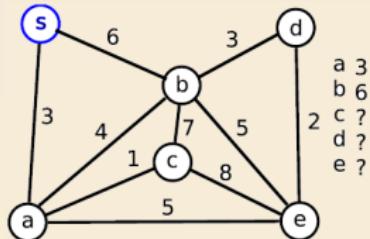
- Najde nejkratší cesty z jednoho vrcholu do všech ostatních.
- Musí tedy projít všechny vrcholy.
- Lze jej aplikovat na orientovaný i neorientovaný graf.
- Vyžaduje nezáporné ohodnocení hran.
- Vyhodnocuje situaci pro konkrétní uzel a pak jej uzavře (nevrací se k němu).

● Bellman-Ford algoritmus

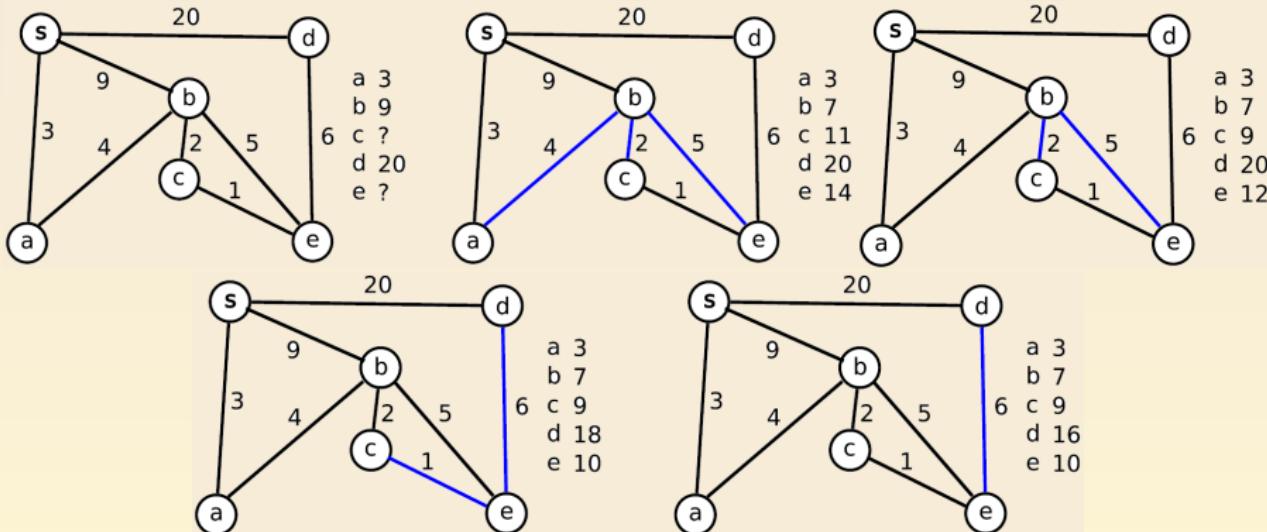
- Základní strukturou podobný Dijkstrovi algoritmu.
- Graf smí obsahovat i záporně ohodnocené hrany.
- Oproti Dijkstrovi algoritmu neuzavírá uzly po projdutí, ale necházá je otevřené až do konce (může se k nim vrátit).



Dijkstrův algoritmus – příklad



Bellman-Ford algoritmus – příklad

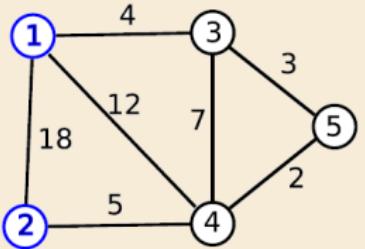
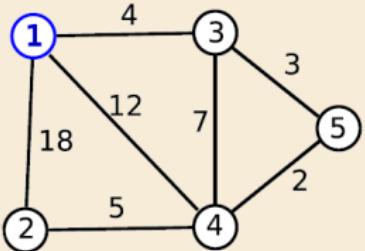
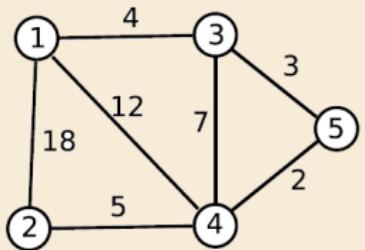


Floyd-Warshallův algoritmus

- Vypočítává nejkratší vzdálenost mezi všemi dvojicemi vrcholů.
- Graf může obsahovat záporně ohodnocené hrany, avšak cykly s celkovým záporným ohodnocením vedou k chybnému řešení.
- Mezi každými dvěma dvojicemi vrcholů postupně vylepšuje nejkratší známou vzdálenost.
- V každém kroku algoritmu je definována množina vrcholů, kterými je možno nejkratší cesty vést.
- Postupně je do této množiny přidáváno po jednom vrcholu.
- Po každém přidání vrcholu jsou aktualizovány cesty mezi všemi dvojicemi vrcholů.



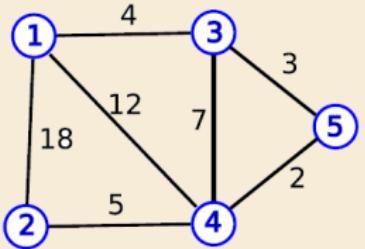
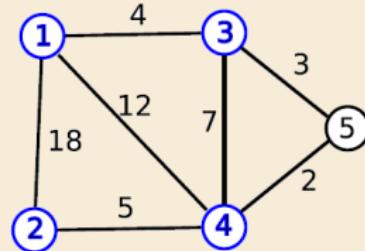
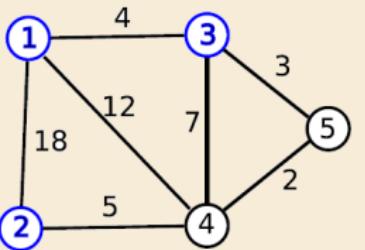
Floyd-Warshallův algoritmus – příklad



	1	2	3	4	5
1	0	18	4	12	?
2	18	0	?	5	?
3	4	?	0	7	3
4	12	5	7	0	2
5	?	?	3	2	0

	1	2	3	4	5
1	0	18	4	12	?
2	18	0	22	5	?
3	4	22	0	7	3
4	12	5	7	0	2
5	?	?	3	2	0

	1	2	3	4	5
1	0	18	4	12	?
2	18	0	22	5	?
3	4	22	0	7	3
4	12	5	7	0	2
5	?	?	3	2	0



	1	2	3	4	5
1	0	18	4	11	7
2	18	0	22	5	25
3	4	22	0	7	3
4	11	5	7	0	2
5	7	25	3	2	0

	1	2	3	4	5
1	0	16	4	11	7
2	16	0	12	5	7
3	4	12	0	7	3
4	11	5	7	0	2
5	7	7	3	2	0

	1	2	3	4	5
1	0	14	4	9	7
2	14	0	10	5	7
3	4	10	0	5	3
4	9	5	5	0	2
5	7	7	3	2	0

Je možné obarvit vrcholy grafu s použitím n barev tak, aby žádné dva sousední vrcholy nebyly obarveny stejnou barvou?

Důležité pojmy:

- **Chromatické číslo grafu**
 - Minimální počet barev n nutný k obarvení grafu tak, aby žádné dva sousední vrcholy nebyly obarveny stejnou barvou.
- **Stupeň uzlu**
- **Úroveň saturace**
 - Počet různých barev spojených s uzlem.



Algoritmus barvení

- ① Uspořádat uzly v klesajícím pořadí podle jejich stupně.



Algoritmus barvení

- ① Uspořádat uzly v klesajícím pořadí podle jejich stupně.
- ② Použít barvu 1 pro první uzel.



- ① Uspořádat uzly v klesajícím pořadí podle jejich stupně.
- ② Použít barvu 1 pro první uzel.
- ③ Vybrat neobarvený uzel s maximální úrovní saturace.
V případě rovnosti vybrat uzel s maximálním stupněm
v neobarveném podgrafu.



- ① Uspořádat uzly v klesajícím pořadí podle jejich stupně.
- ② Použít barvu 1 pro první uzel.
- ③ Vybrat neobarvený uzel s maximální úrovní saturace.
V případě rovnosti vybrat uzel s maximálním stupněm
v neobarveném podgrafu.
- ④ Obarvit vybraný uzel nejmenší možnou barvou.

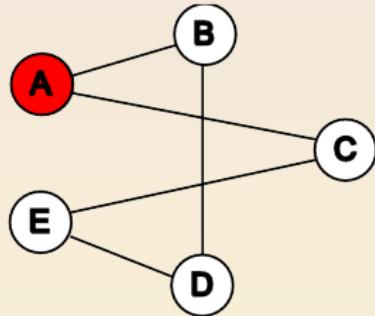


Algoritmus barvení

- ① Uspořádat uzly v klesajícím pořadí podle jejich stupně.
- ② Použít barvu 1 pro první uzel.
- ③ Vybrat neobarvený uzel s maximální úrovní saturace.
V případě rovnosti vybrat uzel s maximálním stupněm
v neobarveném podgrafu.
- ④ Obarvit vybraný uzel nejmenší možnou barvou.
- ⑤ Pokud jsou všechny uzlyobarveny, pak KONEC.
Jinak přejít k bodu 3.



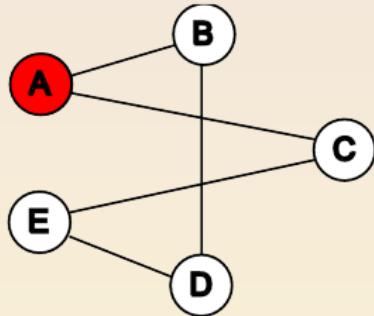
Barvení grafu – příklad



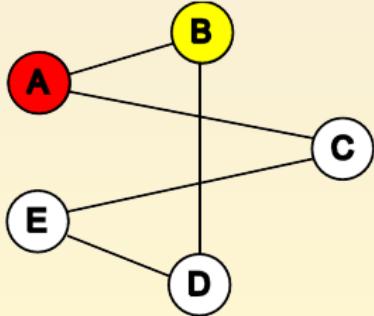
vrchol	A	B	C	D	E
saturace	-	1	1	0	0
stupeň neob.	-	1	1	2	2



Barvení grafu – příklad



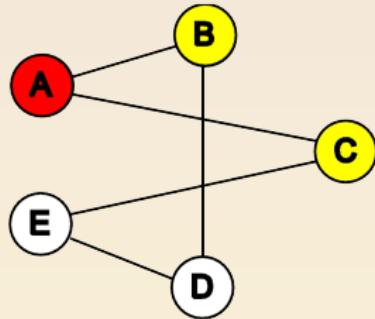
vrchol	A	B	C	D	E
saturace	-	1	1	0	0
stupeň neob.	-	1	1	2	2



vrchol	A	B	C	D	E
saturace	-	-	1	0	0
stupeň neob.	-	-	1	2	2



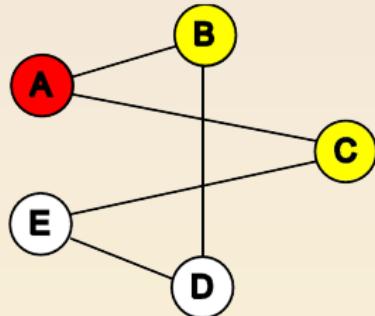
Barvení grafu – příklad



vrchol	A	B	C	D	E
saturace	-	-	-	1	1
stupeň neob.	-	-	-	1	1

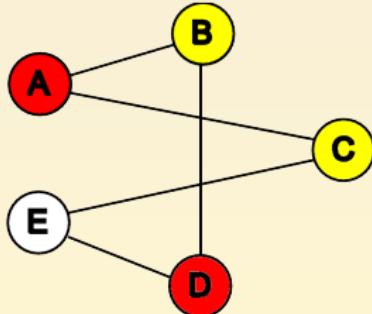


Barvení grafu – příklad

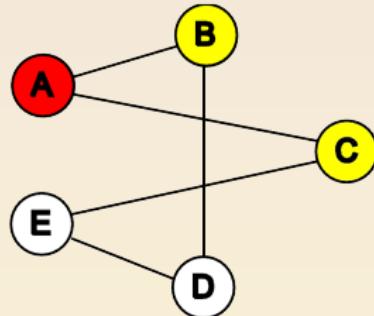


vrchol
saturace
stupeň neob.

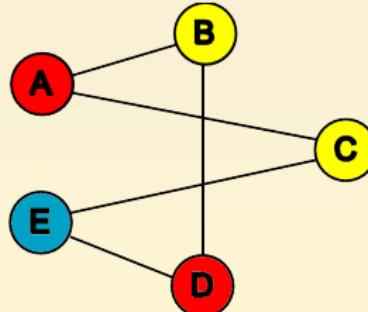
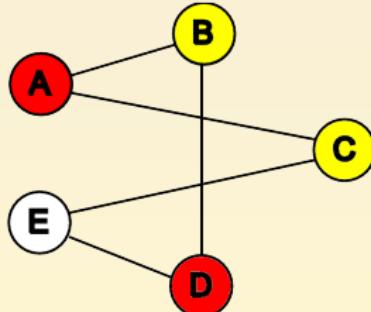
	A	B	C	D	E
saturace	-	-	-	1	1
stupeň neob.	-	-	-	1	1



Barvení grafu – příklad



vrchol	A	B	C	D	E
saturace	-	-	-	1	1
stupeň neob.	-	-	-	1	1



Proobarvení libovolné mapy tak, aby dvě sousední země nebyly obarveny stejnou barvou, potřebujeme nejvýše čtyři barvy.

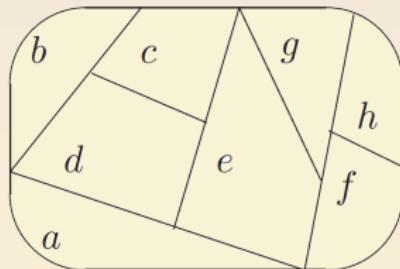
Postup:

- Jednotlivé oblasti na mapě prohlásíme ze vrcholy grafu.
- Oblasti, které spolu sousedí, spojíme hranami. Pozor, alespoň jedna z oblastí musí sousedit celým úsekem hranice!
- Na graf poté aplikujeme algoritmus barvení.



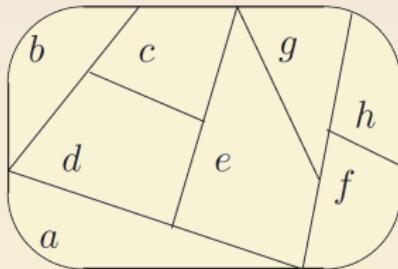
Problém barvení mapy – příklad

Mějme následující mapu:

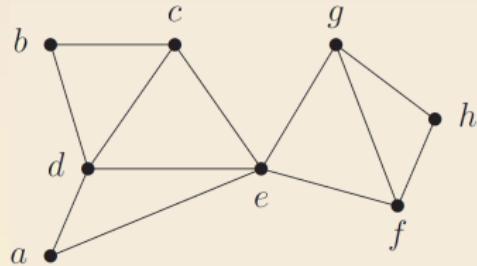


Problém barvení mapy – příklad

Mějme následující mapu:

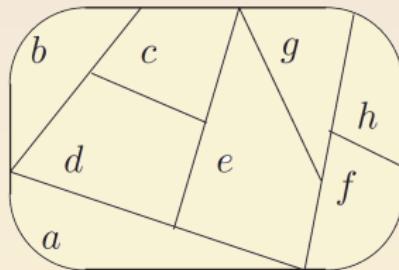


Převedeme jí na graf:

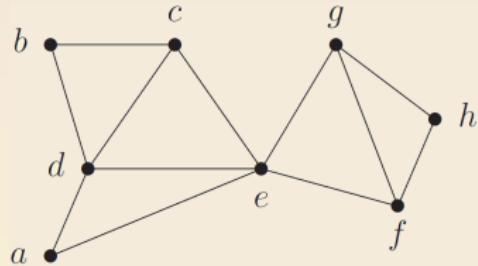


Problém barvení mapy – příklad

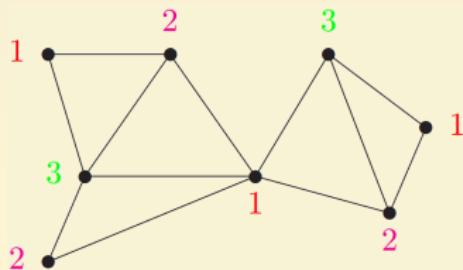
Mějme následující mapu:



Převedeme jí na graf:

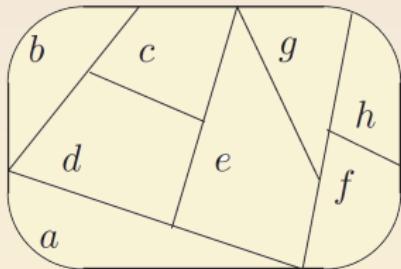


Algoritmus barvení grafů:

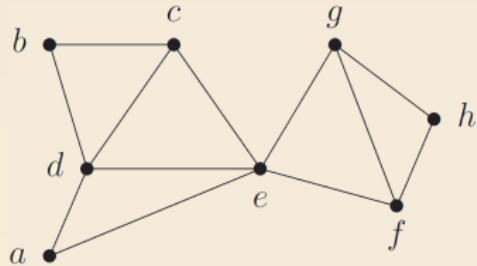


Problém barvení mapy – příklad

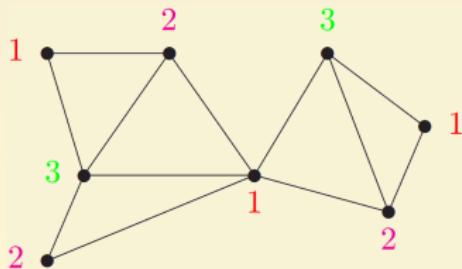
Mějme následující mapu:



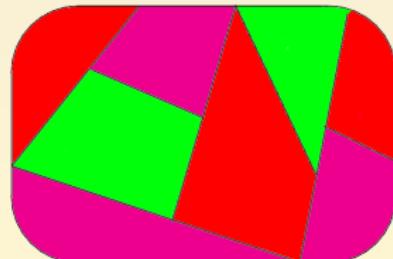
Převedeme jí na graf:



Algoritmus barvení grafů:

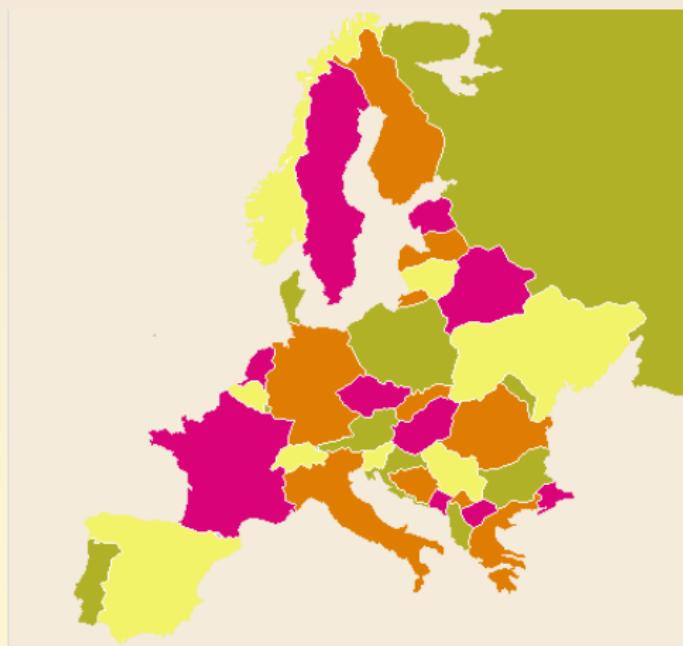


Podle grafu vybarvíme mapu:



Barvení mapy v praxi

Politická mapa Evropy:

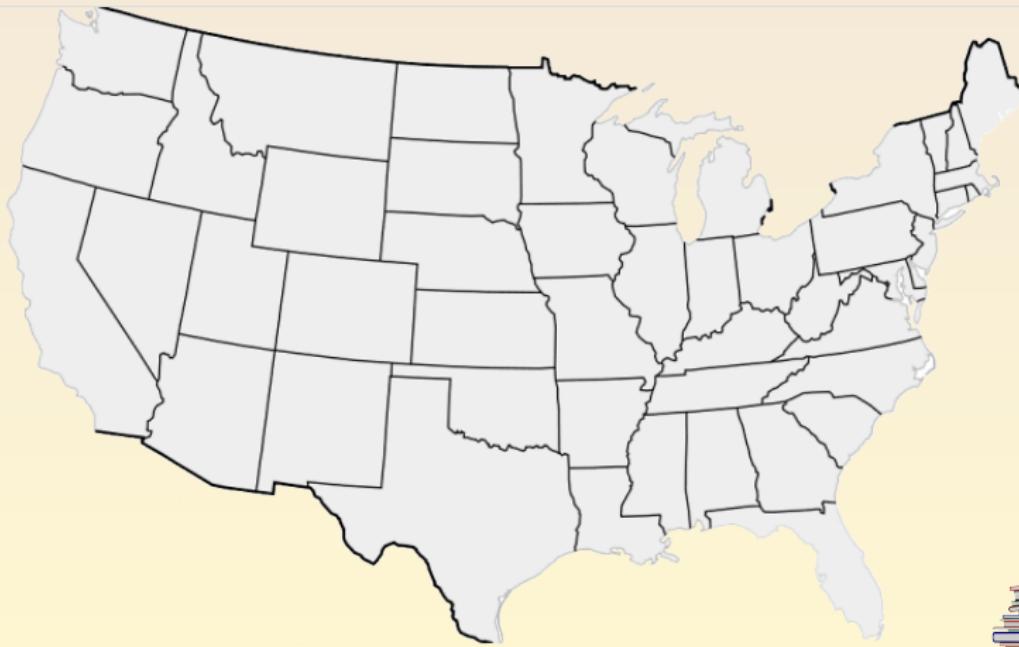


Bonusová úloha



Bonusová úloha

Vybarvěte mapu USA nejnižším možným počtem barev tak, aby žádné dva sousední státy neměly stejnou barvu.



Bonusová úloha – vzorové řešení

