

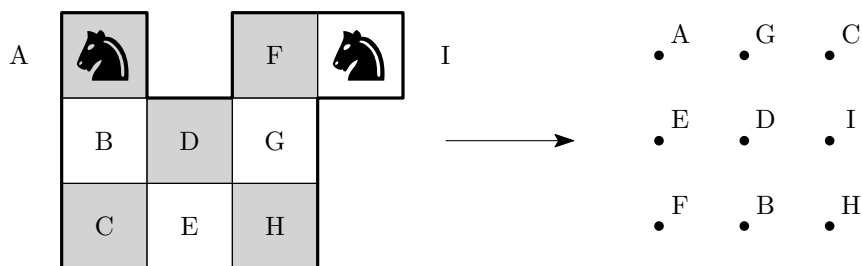
1 Teorie grafů

U úloh s leteckými linkami jsme znázorňovali města pomocí koleček a jednotlivé linky pomocí spojnic či šipek mezi nimi. Takovým schématům se v matematice říká **grafy**, přičemž městům se říká **vrcholy** a linkám mezi nimi **hrany**. Hrany, jak jsme již viděli mohou či nemusí být jednosměrné. Pokud jsou, říkáme jim **orientované hrany** a grafu s orientovanými hranami říkáme **orientovaný graf**.

Situací, v nichž lze takové obrázky kreslit, je bezpočet. Kromě různých dopravních plánů lze například pomocí grafů znázorňovat sociální interakce, třeba přátelství na Facebooku. Nebo, jak bylo naznačeno v hodině o hrách, lze grafů užít k analýze her pomocí V a P pozic. Vrcholy odpovídaly stavům hry a hrany odpovídaly možným tahům (tedy přesunům z jednoho stavu do druhého).

Povězme si tedy o grafech a jejich aplikacích více.

Úloha: K danému výřezu šachovnice nakreslete graf, v němž vrcholy reprezentují políčka a hrany spojují ty dvojice políček, mezi nimiž může šachový jezdec skočit jedním tahem.



Je-li nám v každém kroku dovoleno táhnout libovolným z jezdců na neobsazené pole, rozhodněte (s využitím nakresleného grafu) zda:

- lze prvního jezdce dopravit na výchozí pole druhého.
- lze oba jezdce vyměnit.

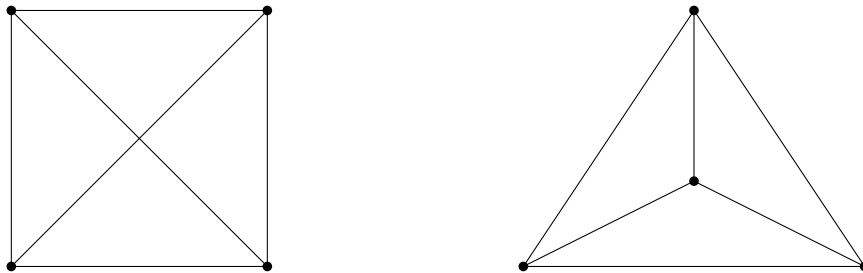
Jako příklad grafu z reálného života uveďme graf přátelství uživatelů sítě facebook. Vrcholy reprezentují uživatele a hrany spojují „přátelé“. Asi nepřekvapí, že tento graf má přes miliardu vrcholů, přičemž z každého z nich vychází průměrně několik stovek hran. Co ale následující fakta, překvapují vás? Co z nich dovedete usoudit o tom, jak se tvoří přátelství mezi lidmi?

- Po zanedbání cca 0,4 % účtů platí, že každý dva uživatelé se na sebe mohou proklikat skrze své známé, známé jejich známých atd..
- Toto proklikání lze pro učinit v průměru pomocí tří až čtyřech prostředníků. V 97 % případů pomocí nejvýše pěti.
- Pro uživatele s nadprůměrným počtem známých typicky platí, že jejich známí mají v průměru také nadprůměrný počet známých. To samé platí opačně o lidech s podprůměrným počtem známých.
- Pro 83,6 % uživatelů platí, že mají méně známých než většina jejich známých. Jak je to možné?

- Pro 92,7 % uživatelů platí, že mají méně známých než, kolik mají průměrně jejich známí. Není zvláštní, že toto procento není stejné jako v předchozím bodě?

1.1 Kreslení grafů

Problém: Najděte společné znaky následujících grafů a rozhodněte se, zda byste je považovali spíše za stejné či za různé.



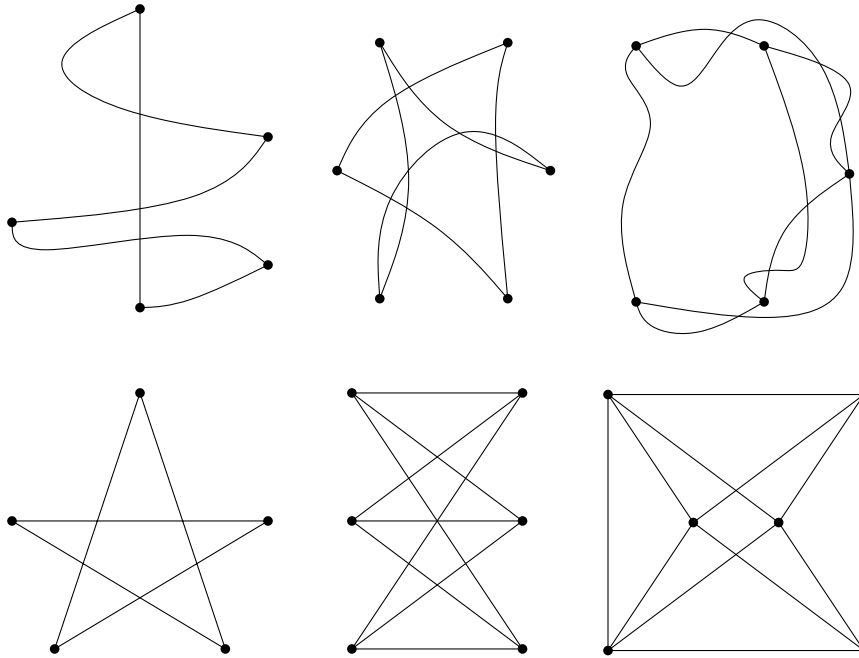
Způsob, jakým se matematici vyrovnali s předchozím problémem je následující. Dohodneme se, že grafem budeme nazývat výčet vrcholů opatřený i výčet dvojic vrcholů spojených hranami, a nebudeme jej tak rovnou spojovat s žádným obrázkem. V tuto chvíli jsou oba obrázky stejným grafem daným výčtem vrcholů $\{1, 2, 3, 4\}$ a hran

$$\{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}.$$

Pro jejich odlišení zavedeme pojem **nakreslení grafu**, tedy znázornění vrcholů jako bodů v rovině a hran jako křivek spojujících odpovídající dvojice bodů. Vidíme tedy, že tentýž graf může mít různá nakreslení.

Například při návrhu procesorů či elektrických obvodů je důležité hledat takové nakreslení grafu, v němž se žádné dvě spojnice nekříží. Grafům, jež je možné takto bez křížení nakreslit budeme říkat **rovinné grafy**. Někdy zase může být praktické, aby nekřížící se křivky byly dokonce úsečkami.

Úloha: Nakreslete následující grafy tak, aby se žádné jejich hrany nekřížily. Zvládnete to i tak, aby hrany byly úsečky?



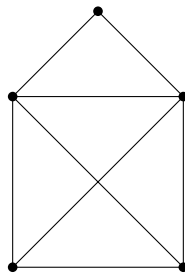
Úloha: Nakreslete rovinný graf, který nelze nakreslit bez křížení pouze pomocí úseček.

Podotkneme ještě, že podobné otázky si lze klást i o kreslení na jiné povrchy než jen rovina, například povrch koule, válce či třeba pneumatiky. Těmito problémy a jejich vícerozměrnými analogiemi se zabývá takzvaná **topologická teorie grafů a hypergrafů**.

1.2 Jednotažky

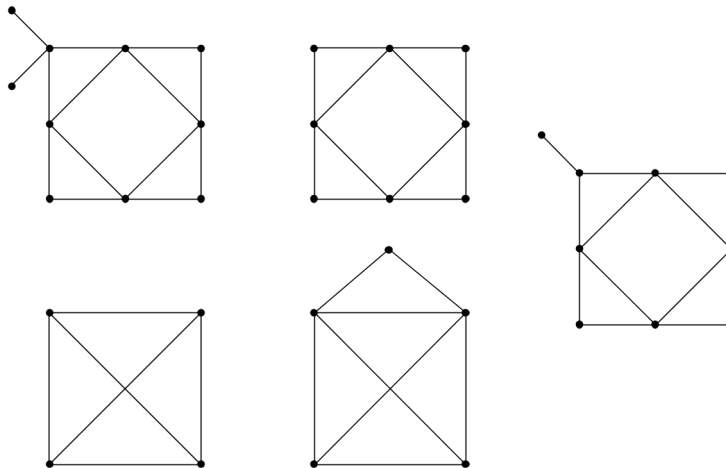
Začneme úlohou.

Úloha: Nakreslete „domeček“ jedním tahem.



V řeči teorie grafů se požaduje najít posloupnost navazujících hran, v níž se každá hrana vyskytne přesně jednou. Naším cílem nyní bude se na situaci podívat o něco obecněji. Grafům, které lze projít jedním tahem budeme říkat **jednotažky**. Jednotažky, které lze dokonce projít tak, aby cesta začínala a končila ve stejném vrcholu, nazveme **kruhovými jednotažky**.

Úloha: Poznejte jednotažky. Určete, které z nich jsou kruhové, a které počáteční či koncové vrcholy připadají v úvahu.



Úloha: Městem Königsberg (dnešní Kaliningrad) protéká řeka Pregole a uprostřed ní se nachází ostrov. Dříve mezi břehy vedly mosty tak jako na obrázku. Rozhnděte, zda lze při jedné procházce projít každý most právě jednou. Můžete si vybrat, odkud procházku začnete a kde skončíte.



Poznámka: Úlohu výše vyřešil roku 1735 Leonhard Euler. Byla to první úloha, která byla vyřešena pomocí postupů a pojmů, které se později začaly nazývat teorií grafů.

Jak jste možná již poznali výběr počátečních vrcholů souvisí s tím, kolik hran z daného vrcholu vychází. Toto číslo nazveme **stupněm vrcholu**.

Úloha: Na základě vašich výsledků rozhodněte, které z následujících hypotéz jsou pravdivé.

- (a) Je-li jednotažka kruhová, lze zvolit počáteční vrchol jakkoliv.
- (b) Ke každému grafu lze přikreslit nejvýše dvě hrany tak, aby se stal jednotažkou.
- (c) Je-li graf jednotažkou, avšak nikoliv kruhovou, lze koncové vrcholy vybrat pouze jedním způsobem.
- (d) Je-li graf nekruhovou jednotažkou, pak po smazání koncových vrcholů (a z nich vycházejících hran) získáme kruhovou jednotažku.

(e) Z nekruhové jednotažky lze přidáním jedné hrany vyrobit kruhovou jednotažku.

Úloha: Nakreslete graf o šesti vrcholech a šesti hranách, v němž má každý vrchol stupeň dva. Nalezněte dvě řešení! Jsou obě kruhové jednotažky?

Úloha: Zformulujte hypotézy a hledejte argumenty i protiargumenty.

- (a) Co musí platit o stupních vrcholů, aby byl graf kruhovou jednotažkou?
- (b) Vrcholy jakých stupňů mohou být koncovými pro nekruhovou jednotažku? Co lze říci o zbylých vrcholech?
- (c) Předchozí úloha ukazuje, že nelze rozhodnout o tom, zda je graf jednotažka, pouze pohledem na stupně jeho vrcholů. Co se s tím dá dělat?

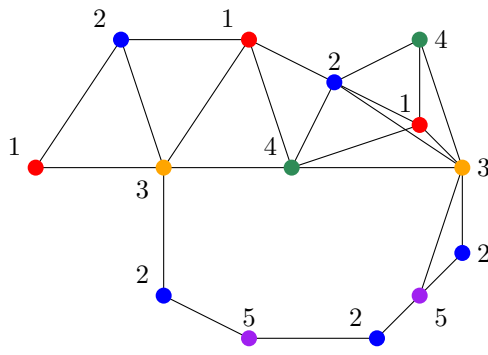
1.3 Barvení grafů

Na území České republiky je rozmístěno přes 150 radiových vysílačů, z nichž ty nejvýznamnější jsou znázorněny na mapě níže. Každý z nich má přiřazené frekvenční pásmo, na němž vysílá signál. Pokud by se ovšem stalo, že nějaké dva blízké vysílače vysílají na podobné frekvenci, došlo by k interferenci radiových vln a znehodnocení signálu. Je tedy nutné, aby každé dva blízké vysílače využívaly jiného frekvenčního pásma. To samozřejmě není problém, je-li k dispozici dostatek pásem, ovšem v praxi dochází k omezením. Určení minimálního počtu potřebných frekvenčních pásem je příkladem dalšího typu problémů, jimiž se zabývá teorie grafů.

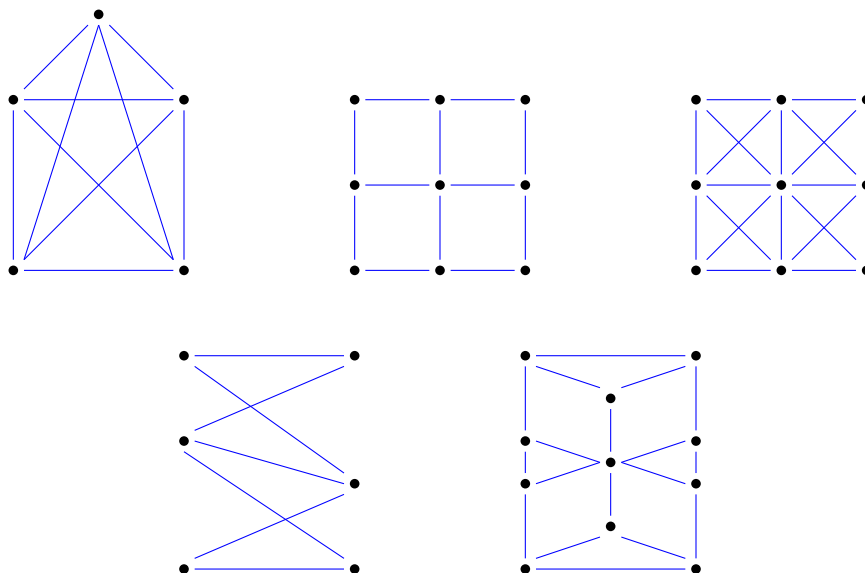


Vrcholy našeho grafu budou vysílače a hranou spojíme ty, jež jsou dostatečně blízké na to, aby docházelo k interferenci. Naším úkolem bude vrcholům grafu přiřadit barvy (frekvenční pásma) tak, aby žádné dva sousední vrcholy neměly stejnou barvu. Každému takovému přiřazení barev budeme říkat prostě obarvení a hlavní otázkou bude, kolik nejméně barev je k obarvení potřeba.

Úloha: V následujícím obrázku je graf obarven pěti barvami. Rozhodněte, zda jej lze obarvit úsporněji, tedy pomocí nižšího počtu barev.

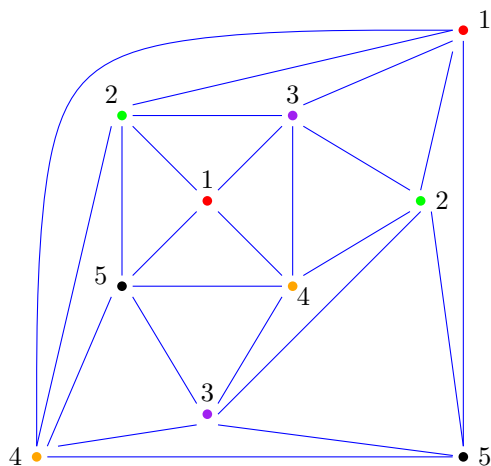


Úloha: Obarvěte následující grafy pomocí co nejnižšího počtu barev.



Na předchozích příkladech je patrné, že žádný pevný počet barev nemůže stačit k obarvení všech možných grafů. Pokud ale zúžíme svoji pozornost jenom na některé typy grafů, situace se může změnit. Nejznámější výsledek v této oblasti je takzvaná Věta o čtyřech barvách. Ta říká, že každý rovinný graf lze obarvit pomocí čtyř barev, neboli je čtyř-obarvitelný. Ačkoliv byla tato hypotéza vznesena již v polovině devatenáctého století, její pravdivost se podařilo spolehlivě dokázat až v posledních desetiletích. Není bez zajímavosti, že součástí důkazu je i užití počítače k ověření čtyř-obarvitelnosti 663 konkrétních grafů.

Úloha: Následující graf byl jistou, byť krátkou, dobu považován za příklad rovinného grafu, který nelze obarvit čtyřmi barvami. Na obrázku je obarven pěti barvami. Ukažte, že lze jednu barvu „ušetřit“.



Grafy, jejichž obarvení je potřeba nalézt v praktických situacích mají často příliš mnoho vrcholů na to, aby bylo možné jejich nejúspornější obarvení najít ručně. Je proto potřeba

vyvinout algoritmus, který si i s velkými grafy poradí v rozumném výpočetním čase. Vývoj takového algoritmu fungujícího pro zcela obecné grafy se ovšem ukazuje jako nesmírně těžký.

Úloha: Uvažme následující algoritmus přiřazující vrcholům grafu přirozená čísla reprezentující barvy.

1. Zvolme libovolný dosud neočíslovaný vrchol.
2. Přiřadíme mu nejmenší přirozené číslo, které se nevyskytuje mezi čísly přiřazenými jeho sousedům.
3. Postup opakujeme, dokud nejsou čísla přiřazená všem vrcholům

Rozhodně, zda je výsledkem algoritmu vždy nejúspěšnější možné obarvení grafu.

2 Grafové algoritmy

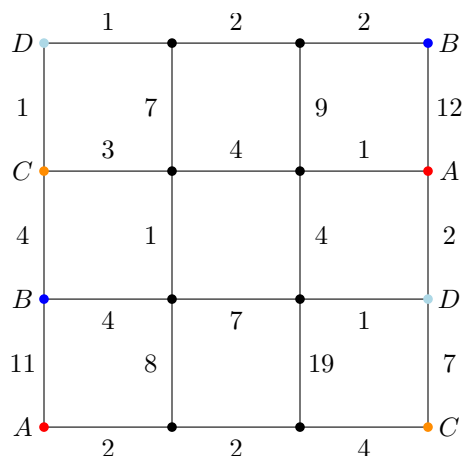
Poslední zastávka při naší exkurzi do teorie grafů bude ryze infromatická. Ukážeme si dvě s praxí spojené úlohy, jejichž podstatu nejlépe zachytíme pomocí grafů.

2.1 Hledání nejkratší cesty

Úlohu o hledání nejkratšího, nejrychlejšího či nejlevnějšího spojení mezi vrcholy grafu řeší miliony počítačových programů dennodenně. Asi nás nepřekvapí, že algoritmů pro výpočet nejkratší cesty využívají například mobilní aplikace pro hledání optimálního dopravního spojení, ať už autem, letadlem nebo v MHD. Méně přímočarými aplikacemi mohou být programy hledající například, jak z dané pozice nejrychleji složit Rubikovu kostku či jak nejrychleji vynutit mat v šachové koncovce.

Zformulujme nyní naši úlohu přesněji. Je dán graf, jehož každá hrana má přiřazenu svoji „délku“, již vyjadřujeme kladným reálným číslem. V grafu máme vyznačeny dva vrcholy a naším úkolem je najít cestu mezi nimi, za jejíž průchod „zaplatíme“ v součtu co nejméně.

Úloha: V následujících grafech nalezněte nejkratší cestu mezi dvojicemi stejně označených vrcholů.



Nyní předložíme několik návrhů, jak by mohl počítač při hledání nejkratší cesty postupovat. Vaším úkolem u každého z nich bude rozhodnout, zda takový postup skutečně vždy dosáhne nejkratší možné cesty. Je-li nejkratších cest více, stačí, aby algoritmus našel nějakou z nich, ne nutně všechny.

Ve všech algoritmech budeme o jednom z označených vrcholů mluvit jako o koncovém a o jednom jako o počátečním.

Algoritmus 1:

1. Do počátečního položíme žeton P a do koncového žeton K.
2. Přesuneme žeton po nejlevnější hraně vedoucí z jeho současného vrcholu do vrcholu, který ještě nenavštívil.

3. Provedeme totéž s žetonem K.
4. Opakujeme kroky 2) a 3) dokud se jeden z žetonů nepřesune na pole, na němž již dříve (či v tu samou chvíli) byl ten druhý. Nazvěme toto pole S.
5. Víme, po jaké cestě se do S dostal žeton P i žeton K. Spojení těchto dvou cest je nejkratší cesta mezi vybranými vrcholy.

Algoritmus 2:

1. Vybereme jednu libovolnou hranu grafu. Vrcholy, jež spojuje, nazvěme X a Y.
2. Pro každého souseda X zjistíme, zda je i sousedem Y. Tím identifikujeme všechny společné sousedy X a Y.
3. Pro každého společného souseda Z vrcholů X a Y, zkontrolujeme, zda je kratší přímá cesta z X do Y, nežli cesta X-Z-Y. Pokud nikoliv, hranu X-Y z grafu odstraníme.
4. První tři kroky algoritmu postupně aplikujeme na všechny hrany v grafu.
5. Předchozím postupem jsme smazali všechny hrany vedoucí k neefektivním cestám. Mezi dvěma vyznačenými vrcholy zůstávají tedy pouze cesty (může jich být více) s nejkratší možnou délkou. Vybereme z nich kteroukoliv.

Algoritmus 3:

Tento algoritmus nalezne nejkratší cestu z počátečního vrcholu P do všech ostatních, tedy i do cílového. Vrcholy, do nichž jsme našli již nějakou cestu budeme značit červeně a vrcholy, do nichž jsme již našli nejkratší cestu budeme značit modře. Pro každý červený i modrý vrchol si budeme pamatovat dosud nejkratší nalezenou cestu do něj vedoucí jakož i její délku. Této cestě budeme říkat aktuálně nejkratší cesta a její délce budeme říkat aktuální vzdálenost.

1. Vrchol P označíme modře (nejkratší cestu do něj známe, má délku nula) a všechny jeho sousedy červeně, přičemž jim zaktualizujeme aktuálně nejkratší cestu i aktuální vzdálenost.
2. Z červených vrcholů vybereme ten, který má **nejnižší** aktuální vzdálenost. Označme jej V.
3. Přebarvíme V namodro a všechny jeho nevybarvené sousedy vybarvíme červeně.
4. Vyberme červeného souseda V a označme jej S.
5. Zkontrolujeme, zda aktuálně nejkratší cesta do S je kratší než cesta P-V-S. Pokud ne, změňme ji.

6. Krok 5 opakujeme pro každého červeného souseda V.
7. Kroky 2-6 opakujeme dokud nejsou všechny vrcholy modré. Známe pak nejkratší cesty do všech vrcholů a jsme hotovi.

Algoritmus 4:

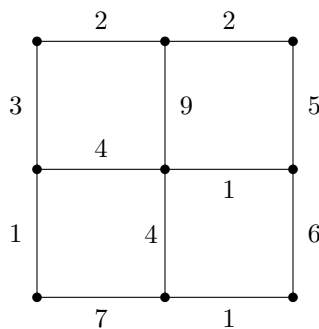
Budeme postupovat téměř stejně jako v algoritmu 3. Jedinou změnou bude, že tučné slovo nejnižší nahradíme slovem libovolný.

2.2 Hledání minimální kostry

Když se ve dvacátých letech 20. století elektrifikovala jižní Morava, vyvstala otázka, jak zavést proud do každého z měst za co nejnižších nákladů. Přesněji za minimální celkové délky elektrického vedení. Podotkněme, že pro zavedení proudu mezi dvěma městy nevadí, pokud vedení prochází také přes jiná města. Různé trasy elektrického vedení se ovšem mohou rozdělovat a spojovat pouze ve městech. Na obrázku naleznete příklad takového elektrického vedení.



Úloha: Nalezněte nejúspěšnější síť v následujícím grafu.



V řeči teorie grafů lze předchozí úlohu zformulovat takto. Je dán graf, jehož každá hrana je opatřena kladným reálným číslem. V tomto kontextu budeme toto číslo nazývat **váhou hrany**. Cílem je najít podmnožinu hran zajišťující spojení mezi každou dvojicí vrcholů a přitom s minimálním možným součtem vah (tzv. minimální kostru). Pro vysvětlení dodejme, že kostrou grafu se nazývá každá podmnožina hran zajišťující přesně jednu cestu mezi každou dvojicí měst.

U následujících algoritmů opět rozhodněte, zda opravdu vedou k nalezení minimální kostry.

U všech algoritmů budeme postupně hrany do hledané kostry vybírat a aktuálně vybraným hranám budeme říkat modré.

Algoritmus 1:

1. Pomocí algoritmu pro hledání nejkratší cesty určíme nejkratší vzdálenost mezi každými dvěma městy.
2. Vybereme dvojici měst s největší vzdáleností. Hrany tvořící nejkratší cestu mezi nimi obarvíme modře.
3. Vybereme vrchol V , z něhož nevede žádná modrá hrana.
4. Určíme vzdálenost vrcholu V od každého z vrcholů napojeného na nějakou modrou hranu, vybereme z nich ten nejbližší a nazveme jej S .
5. Všechny hrany na nejkratší cestě mezi V a S obarvíme modře.
6. Kroky 3-5 opakujeme, dokud není každé město napojeno na nějakou modrou hranu.

Algoritmus 2:

Množiny vrcholů mezi nimiž se lze přepravit pouze pomocí modrých hran budeme nazývat komponenty. Na začátku, tedy před obarvením modrých hran, je každý vrchol samostatnou komponentou.

1. Nalezneme nejkratší hranu spojující vrcholy z různých komponent.
2. Hranu obarvíme modře a komponenty, jež spojuje, sloučíme do jedné.
3. Kroky 1-2 opakujeme, dokud nezbyde jediná komponenta.

Algoritmus 3:

1. Vybereme libovolný dosud nevybraný vrchol V .

2. Mezi nemodkými hranami z něj vycházejícími vybere tu s nejnižší vahou a obarvíme ji modře.
3. Kroky 1-2 opakujeme, dokud nebyl takto vybrán každý vrchol.

Algoritmus 4:

1. Napíšeme si seznam hran v pořadí podle jejich váhy (nejnižší váhy jako první).
2. Nyní postupně procházíme hrany v seznamu.
3. Pokud hrana spojuje města, mezi nimiž se nelze dopravit po modrých hranách, obarvíme ji modře.
4. Po projití seznamu tvoří modré hrany minimální kostru.