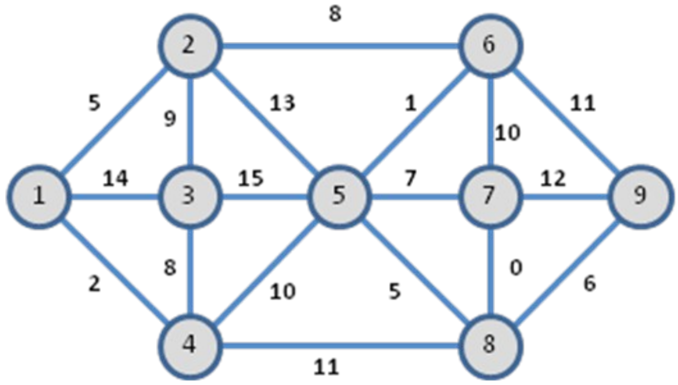
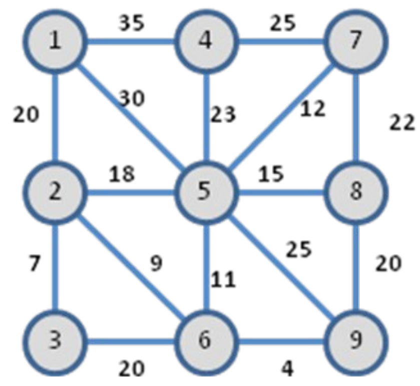
CVIČENÍ 8

Téma: Minimální kostry, hladové algoritmy, minimální cesty.
Cíle: Zvládnout algoritmy určování minimálních koster (Borůvka, Jarník), návrh hladových algoritmů, algoritmy pro hledání minimálních cest (Dijkstra).

Minimální kostry:	
1.	Zjistěte, zda jsou následující tvrzení pravdivá či nikoliv. Pravdivá tvrzení dokažte, nepravdivá vyvraťte co nejjednodušším protipříkladem: a) Jsou-li ohodnocení všech hran v souvislém neorientovaném grafu navzájem různá, je jeho minimální kostra určena jednoznačně. b) Jsou-li ohodnocení všech hran v souvislém neorientovaném grafu navzájem různá, pak mají každé dvě jeho různé kostry různá ohodnocení.
2.	Nechť *h* je hrana s minimálním ohodnocením *w(h)* v neorientovaném grafu *G*. Dokažte, že existuje minimální kostra grafu *G* obsahující hranu *h*. Musí hranu *h* obsahovat každá minimální kostra?
3.	Nechť *h* je hrana s maximálním ohodnocením *w(h)* v nějaké kružnici neorientovaného grafu *G*. Dokažte, že existuje minimální kostra grafu *G,* která neobsahuje hranu *h*. Může existovat minimální kostra, která hranu *h* obsahuje?
4.	Nechť *T je* minimální kostra grafu *G* s ohodnocením hran *w* a označme jako $(w_1, w_2, \ldots, w_{n-1})$ vzestupně uspořádanou posloupnost ohodnocení jejích hran. Dokažte, že i každé jiné minimální kostře *T*' grafu *G* bude odpovídat stejná uspořádaná posloupnost ohodnocení hran.
5.	Předpokládejte, že ohodnocení hran grafu *G* jsou celá čísla z intervalu $\langle 1, \|U\| \rangle$. Jak je možné zrychlit v tomto případě Borůvkův-Kruskalův algoritmus? Je možné ke zrychlení využít informace, že ohodnocení hran jsou celá čísla z intervalu $\langle 1, W \rangle$ pro nějakou konstantu *W*?
6.	Máme daný souvislý neorientovaný graf G = ⟨H,U⟩, pro který jsme určili minimální kostru T. Graf G’ vznikl tak, že jsme ke grafu G přidali jeden uzel a propojili ho pomocí p nových hran s uzly grafu G. Navrhněte postup, jak určit minimální kostru grafu G’. Jaká bude složitost tohoto postupu?
7.	Máme daný souvislý neorientovaný graf G = ⟨H,U⟩, pro který jsme určili minimální kostru T. Graf G’ vznikl tak, že jsme ke grafu G přidali jednu hranu mezi existujícími uzly grafu G. Navrhněte postup, jak určit minimální kostru grafu G’. Jaká bude složitost tohoto postupu?
8.	Máme daný souvislý neorientovaný graf G = ⟨H,U⟩, pro který jsme určili minimální kostru T. Graf G’ vznikl tak, že jsme z grafu G odebrali jednu hranu mezi existujícími uzly grafu G. Navrhněte postup, jak určit minimální kostru grafu G’. Jaká bude složitost tohoto postupu?
9.	Určete minimální kostry následujících dvou grafů pomocí Borůvkova-Kruskalova a Jarníkova-Primova algoritmu.





Hladové algoritmy:

10. Upravte Huffmanův algoritmus tak, aby vytvářel minimální pravidelný strom se zadaným stupněm r.

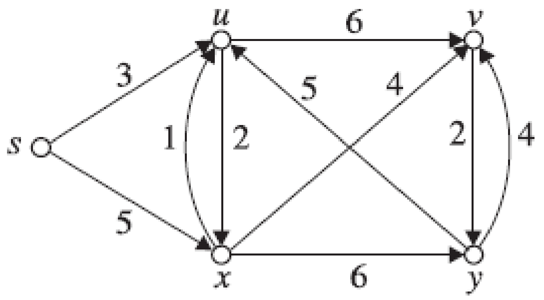
11. Dokažte, že hodnotu E_w vnější w-délky pravidelného stromu vytvořeného Huffmanovým algoritmem lze spočítat jako součet ohodnocení všech jeho vnitřních uzlů.

Hledání minimálních cest:

12. Na rozehřátí

- W-vzdálenost $\boldsymbol{d_w(u, v)}$ je definována jako w-délka nejkratší cesty v grafu s ohodnocením $\boldsymbol{w : H \to R}$, nechť graf G obsahuje záporně ohodnocené hrany, ale **žádný záporně ohodnocený cyklus.** Napište znění trojúhelníkové nerovnosti pro $\boldsymbol{d_w}$ a uveďte, zda v tomto případě platí.
- Předpokládejme, že hrany kružnice jsou ohodnoceny (**kladnými) přirozenými čísly**. Jak se změní vlastnosti týkající se poloměru ***r,*** průměru ***T*** a počtu středů takové kružnice oproti situaci, kdy každá její hrana má délku (ohodnocení) rovnou jedné?

13. **Dijkstrův algoritmus**
Určete alespoň dva různé stromy minimálních cest z uzlu *s* grafu na následujícím obrázku.



14. (Zopakování principu Dijkstrova algoritmu) Pomocí Dijkstrova algoritmu zjistěte pro graf na předchozím obrázku vzdálenosti $\boldsymbol{d_w(x, \ldots)}$ a $\boldsymbol{d_w(v, \ldots)}$.

15. Najděte příklad grafu obsahujícího hrany se záporným ohodnocením, pro který bude výsledek získaný Dijkstrovým algoritmem nesprávný.

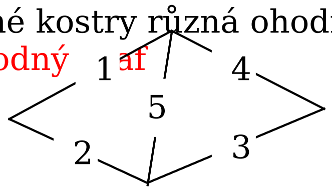
16. Najděte příklad grafu, kde jsou některé (případně i všechny) hrany záporně ohodnocené a přitom Dijkstrův algoritmus správně určí výsledek.

CVIČENÍ 8 – možná řešení

Téma: Minimální kostry, hladové algoritmy, minimální cesty.
Cíle: Zvládnout algoritmy určování minimálních koster (Borůvka, Jarník), návrh hladových algoritmů, algoritmy pro hledání minimálních cest (Dijkstra).

Minimální kostry:

1. Zjistěte, zda jsou následující tvrzení pravdivá či nikoliv. Pravdivá tvrzení dokažte, nepravdivá vyvraťte co nejjednodušším protipříkladem:
   a) Jsou-li ohodnocení všech hran v souvislém neorientovaném grafu navzájem různá, je jeho minimální kostra určena jednoznačně.
   ANO – výběr hran např. Borůvkovým algoritmem je jednoznačný.
   b) Jsou-li ohodnocení všech hran v souvislém neorientovaném grafu navzájem různá, pak mají každé dvě jeho různé kostry různá ohodnocení.
   NE – např. 1+4+5 = 2+3+5 pro vhodný graf se 4 uzly a pěti hranami



2. Nechť h je hrana s minimálním ohodnocením $w(h)$ v neorientovaném grafu G. Dokažte, že existuje minimální kostra grafu G obsahující hranu h. Musí hranu h obsahovat každá minimální kostra?
Hranu dám jako první v pořadí a vyberu ji Borůvkovým algoritmem. Hran s minimálním ohodnocením ovšem může být více, pokud by vytvářely kružnici, tak nemohou být všechny v min. kostře.

3. Nechť h je hrana s maximálním ohodnocením $w(h)$ v nějaké kružnici neorientovaného grafu G. Dokažte, že existuje minimální kostra grafu G, která neobsahuje hranu h. Může existovat minimální kostra, která hranu h obsahuje?
Hranu h z grafu vypustím, zůstane souvislý a vytvořím jeho min. kostru, ta bude min. kostrou i pro původní graf.
Může, pokud všechny hrany dotyčné kružnice mají stejné maximální ohodnocení.

4. Nechť T *je* minimální kostra grafu G s ohodnocením hran w a označme jako $(w_1, w_2, \dots, w_{n-1})$ vzestupně uspořádanou posloupnost ohodnocení jejích hran. Dokažte, že i každé jiné minimální kostře T' grafu G bude odpovídat stejná uspořádaná posloupnost ohodnocení hran.
Plyne z Borůvkova algoritmu, hrany se stejným ohodnocením se budou po seřazení nacházet v kompaktních úsecích, ale pořadí hran uvnitř těchto úseků může být různé.

5. Předpokládejte, že ohodnocení hran grafu G jsou celá čísla z intervalu $\langle 1, |U| \rangle$. Jak je možné zrychlit v tomto případě Borůvkův-Kruskalův algoritmus? Je možné ke zrychlení využít informace, že ohodnocení hran jsou celá čísla z intervalu $\langle 1, W \rangle$ pro nějakou konstantu W?
Pro řazení lze použít nějakou variantu CountingSortu, která má lineární složitost, takže výsledná složitost bude dána implementací operací FIND a UNION.

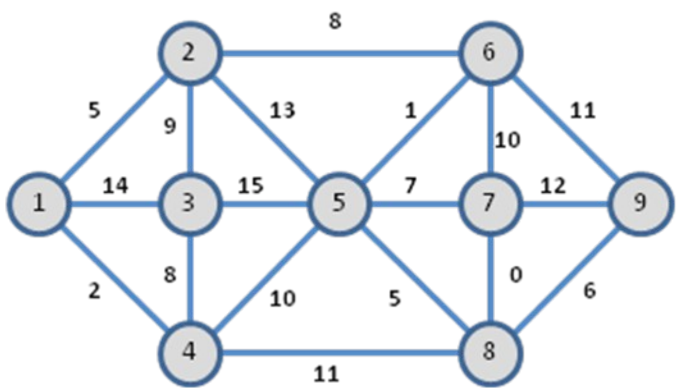
6. Máme daný souvislý neorientovaný graf $\mathbf{G} = \langle \mathbf{H},\mathbf{U} \rangle$, pro který jsme určili minimální kostru **T**. Graf **G’** vznikl tak, že jsme ke grafu **G** přidali jeden uzel a připojili jsme ho pomocí **p** nových hran s uzly grafu **G.** Navrhněte postup, jak určit minimální kostru grafu **G’**. Jaká bude složitost tohoto postupu?
Ke kostře T přidáme tu z nových hran, která má nejmenší ohodnocení. Složitost

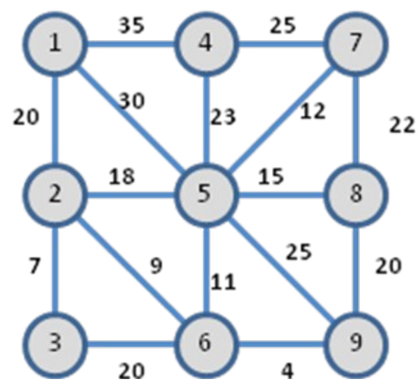
je Θ(p).

7. Máme daný souvislý neorientovaný graf G = ⟨H,U⟩, pro který jsme určili minimální kostru T. Graf G’ vznikl tak, že jsme ke grafu G **přidali jednu hranu** mezi existujícími uzly grafu G. Navrhněte postup, jak určit minimální kostru grafu G’. Jaká bude složitost tohoto postupu?
Nalezneme (jednoznačně určenou) cestu v kostře T mezi krajními uzly přidané hrany, a ta spolu s dotyčnou hranou vytváří kružnici, ze které je třeba vypustit hranu s největším ohodnocením. Jak se tohle udělá např. jenom z reprezentace kostry pomocí odkazu na předchůdce? Jak se případně tato reprezentace změní, aby odpovídala nové kostře? Tyhle otázky zkuste diskutovat.
Složitost by se měla udržet na O(|U|), protože jsou to opakované probírky uzlů a hran kostry.

8. Máme daný souvislý neorientovaný graf G = ⟨H,U⟩, pro který jsme určili minimální kostru T. Graf G’ vznikl tak, že jsme z grafu G **odebrali jednu hranu** mezi existujícími uzly grafu G. Navrhněte postup, jak určit minimální kostru grafu G’. Jaká bude složitost tohoto postupu?
Pokud se odebrala hrana, která nepatří kostře, nic není třeba dělat. (Jak se tohle zjistí z reprezentace pomocí odkazů na předchůdce?) V opačném případě zjistíme rozklad {U1, U2} množiny uzlů daný rozpadem kostry na dva podstromy (jak se to asi udělá?) a nalezneme hranu s krajními uzly v různých třídách rozkladu, která má nejnižší ohodnocení, a tou dva podstromy opět spojíme. Opět – jak se tohle dá udělat s použitím výchozí reprezentace grafu pomocí seznamů sousedů doplněných ohodnoceními hran a kostrou reprezentovanou pomocí odkazů na předchůdce?

9. Určete minimální kostry následujících dvou grafů pomocí Borůvkova-Kruskalova a Jarníkova-Primova algoritmu.





Tohle zkuste naživo …

Hladové algoritmy:

10. Upravte Huffmanův algoritmus tak, aby vytvářel minimální pravidelný strom se zadaným stupněm r.
Především je třeba případně doplnit minimálním počtem fiktivních listů s ohodnocením 0, aby celkový počet listů byl k.(r-1)+1. V algoritmu pak stačí dodat funkci MakeNode součet nejnižších r ohodnocení uzlů a následně na tyto uzly nasměrovat jednotlivé odkazy z vytvořeného rodiče.

11. Dokažte, že hodnotu E_w vnější w-délky pravidelného stromu vytvořeného Huffmanovým algoritmem lze spočítat jako součet ohodnocení všech jeho vnitřních uzlů.
Představíme si cestu od listu x ke kořeni r ve stromu vytvořeném Huffmanovým

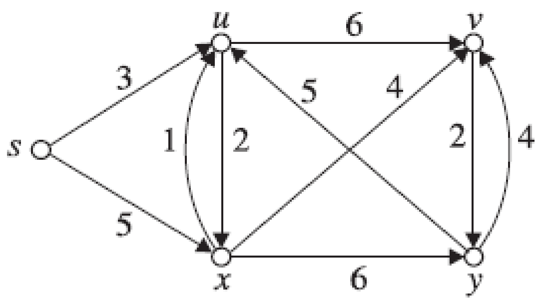
algoritmem. Hodnota w[x] se vyskytne v součtovém ohodnocení všech vnitřních uzlů této cesty včetně kořene, tedy přesně tolikrát, kolik činí hloubka listu x. Součet přes všechny vnitřní uzly stromu tedy odpovídá vnější w-délce.

Hledání minimálních cest:

12. Na rozehřátí

- W-vzdálenost ***d_w(u, v)*** je definována jako w-délka nejkratší cesty v grafu s ohodnocením
w : H -> R, nechť graf G obsahuje záporně ohodnocené hrany, ale **žádný záporně ohodnocený cyklus.** Napište znění trojúhelníkové nerovnosti pro ***d_w*** a uveďte, zda v tomto případě platí.
Platí, plyne to z definice vzdálenosti jako délky nejkratší cesty, a ta je buď konečná nebo neexistuje (pak je vzdálenost +∞).
- Předpokládejme, že hrany kružnice jsou ohodnoceny (**kladnými) přirozenými čísly**. Jak se změní vlastnosti týkající se poloměru ***r,*** průměru ***T*** a počtu středů takové kružnice oproti situaci, kdy každá její hrana má délku (ohodnocení) rovnou jedné?
Průměr může být libovolně větší než poloměr, středem může být jeden nebo všechny uzly.

13. **Dijkstrův algoritmus**
Určete alespoň dva různé stromy minimálních cest z uzlu *s* grafu na následujícím obrázku.

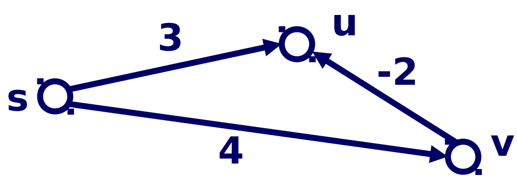


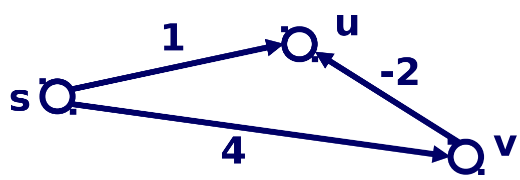
Vzdálenosti: d[u]=3, d[v]=9, d[x]=5, d[y]=11
Kostry minimálních cest z uzlu s jsou čtyři: (s,u), (s,x), (u,v) nebo (x,v), (x,y) nebo (v,y)

14. (Zopakování principu Dijkstrova algoritmu) Pomocí Dijkstrova algoritmu zjistěte pro graf na předchozím obrázku vzdálenosti ***d_w(x, ...)*** a ***d_w(v, ...)***.
Vzdálenosti z uzlu x: d[u]=1, d[v]=4, d[y]=6, d[s]=7
Vzdálenosti z uzlu v: d[y]=2, d[s]=3, d[u]=6, d[x]=8...

15. Najděte příklad grafu obsahujícího hrany se záporným ohodnocením, pro který bude výsledek získaný Dijkstrovým algoritmem nesprávný.
Viz např. graf dole vlevo.

16. Najděte příklad grafu, kde jsou některé (případně i všechny) hrany záporně ohodnocené a přitom Dijkstrův algoritmus správně určí výsledek.
Viz např. graf dole vpravo nebo jakýkoliv kořenový strom při určování vzdáleností (nejen!) od kořene..


3
u
-2
s
4
v


1
u
-2
s
4
v