Minimální kostra

Při úvahách o minimální kostře budeme vždy předpokládat, že pracujeme se souvislým grafem G = (V, E). Kdyby byl graf nesouvislý, hledali bychom minimální kostry jeho komponent, které jsou ale samy souvislými grafy, takže zobecnění úlohy na grafy nesouvislé nepřináší v tomto případě žádný nový pohled na celou problematiku a budeme se mu vyhýbat.

D1:
Řez grafu G = (V, E) je neuspořádaná dvojice disjunktních neprázdných množin vrcholů $S = (S_1, S_2)$, jejichž sjednocení je celá množina V. Symbolicky: $S_1 \cup S_2 = V$, $S_1 \cap S_2 = \emptyset$, $S_1 \neq \emptyset$, $S_2 \neq \emptyset$.

Řekneme, že hrana (a,b) **spojuje** řez $S = (S_1, S_2)$, pokud její koncové vrcholy leží v různých množinách řezu. To jest, pokud platí $(a \in S_1 \wedge b \in S_2) \vee (a \in S_2 \wedge b \in S_1)$.

1.
Na přednášce je pojem řezu definován odlišně:
D2:
Řez grafu je množina hran $F \subseteq E$, taková, že platí: $\exists\, U \subset V$: $F = \{ \{u,v\} \in E \mid u \in U, v \notin U \}$.
Rozmyslete si, že platí následující vztahy mezi oběma definicemi řezu D1 a D2:
a. Množina všech hran, které spojují řez $S = (S_1, S_2)$ podle definice D1, je také řezem podle definice D2.
b. Řez F v definici D2 určuje jednoznačně dvojici množin vrcholů (U, V — U). Tato dvojice je zároveň řezem podle definice D1. Množina hran spojujících řez S = (U, V — U) je tudíž množinou F v definici D2.

2.
Kolik různých řezů lze vytvořit v grafu s 10 vrcholy?
Řešení Každá neprázdná a různá od V(G) podmnožina vrcholů $W \subseteq V(G)$ grafu G s *n* vrcholy definuje řez S = (W, V(G) — W). Uvedených množin W je celkem $2^n - 2$. Protože množiny W a V(G) — W definují stejný řez, musíme uvedené číslo dělit dvěma, takže máme $2^{(n-1)} - 1$. V grafu s 10 vrcholy tak lze nalézt 511 různých řezů.

3.
Je pravda, že pro každý řez S a každou kostruT gafu G platí, že mezi všemi hranami spojujícími S existuje alespoň jedna, která náleží také T?
Řešení Věc je jasná vícemeně na první pohled, uděláme ale pro úplnost také o něco důkladnější úvahu. Zvolme libovolné dva uzly $a \in S_1$ a $b \in S_2$, $a \neq b$. Protože G je souvislý graf, existuje cesta z a do b, to jest existuje posloupnost navzájem různých vrcholů $a_1=a, a_2, a_3, ..., a_p = b$, $(1 < p \leq |V(G)|)$ tak, že (a_i, a_{i+1}) je hrana G pro $i = 1,2, ..., p-1$.
Protože b neleží v S_1, musí v uvedené posloupnosti existovat uzel, a_k $(1 \leq k < p)$, který leží v S_1 a zároveň pro všechna $i > k$ platí, že a_i neleží v S_1. Proto také uzel a_{k+1} leží v S_2 a tudíž i hrana (a_k, a_{k+1}) spojuje S. Ukázali jsme, že na každé cestě spojující dva uzly z různých množin řezu leží alespoň jedna spojující hrana tohoto řezu. Nyní uvažme konkrétní řez S a konkrétní kostru T. Zvolme dva uzly x, y z různých množin řezu S. Kostra T pokrývá všechny uzly grafu a je souvislá, proto v ní existuje cesta C mezi x a y. Podle toho, co jsem ukázali výše, musí tato cesta obsahovat spojující hranu řezu S. Protože celá cesta C leží v T, je příslušná spojující hrana součásti T, což jsme měli ukázat.

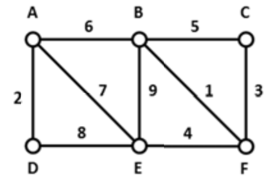
4.
Předpokládejme, že G má *n* vrcholů, je dána kostra T a řez S. Označme symbolem sh(G, T, S) číslo udávající počet hran, které jsou zároveň součástí T a spojují S. Symbolem sh(G) označme maximum ze všech čísel sh(G, T , S) když T resp. S probíhá všechny kostry resp. řezy G. Pomocí hodnoty *n* vyjádřete číslo sh(G). Pomůcka: Obarvěte kostru grafu dvěma barvami.
Řešení Zvolme libovolnou kostru T. K ní zkonstruujeme takový řez S, že každá hrana kostry T bude spojovat řez S. Kostru lze obavit dvěma barvami, dejme tomu námořnickou modří a svinibrodskou zelení.

Připomínáme, že obarvení lze najít tak, že zvolíme libovolný uzel r kostry T a každému jejímu uzlu přiřadíme vzdálenost (v kostře!) od uzlu r. Potom uzly v liché vzdálenosti obarvíme jednou barvou a uzly v sudé vzdálenosti (včetně 0) obarvíme druhou barvou. Uzly jedné barva prohlásíme za množinu S_1 a uzly druhé barvy prohlásíme za množinu S_2. Tyto množiny jsou disjunktní, a jejich sjednocení je V(G), takže definují řez S = (S_1, S_2). Koncové uzly každé hrany kostry T mají různou barvu (v obarvení grafu nikdy nesousedí dva uzly stejné barvy), tudíž tyto uzly leží v různých množinách řezu S, tudíž každá hrana kostry T spojuje řez S. Všechny kostry grafu G mají stejný počet rovný $n - 1$, takže uzavíráme: sh(G) = $n - 1$.

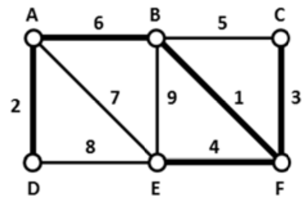
Pro daný hranově ohodnocený (vážený) graf G = (V, E) a pro daný řez S = (S_1, S_2) označíme jako **tenkou hranu** každou hranu, která spojuje řez S a má minimální váhu ze všech hran spojujících S.

5.

V uvedeném grafu G_1 najděte minimální kostru T. Pro každou hranu $e \in$ T zkonstruujte řez S_e grafu G_1 takový, že e bude tenkou hranou pro G a S.



Řešení Minimální kostru vidíme na obrázku. S hledáním řezů začneme u nejdražší hrany kostry. To je hrana (A, B). tato hrana dělí kostru na dvě části. Část T1 obsahuje vrcholy A, D, část T2 obsahuje vrcholy B, C, E, F. Tyto dvě části (= množiny vrcholů) definují řez S. Pohledem do obrázku ověříme, že hrany spojující takto vytvořený řez jsou (A, B), (A, E), (D, E) a že všechny mají cenu stejnou nebo větší než hrana (A, B), tudíž hrana (A,B) je tenkou hranou S. Analogicky dále budeme postupovat pro další hrany a získáme tak seznam řezů:



Hrana (E, F) -- řez daný množinami uzlů {A, D, E} a {B, C, F}.
Hrana (C, F) -- řez daný množinami uzlů {C} a {A, B, D E, F}.
Hrana (A, D) -- řez daný množinami uzlů {A, B, C, F} a {D, E}.
Hrana (B, F) -- řez daný množinami uzlů {A, B, D, E} a {C, F}.
Uvedené řezy nejsou dány jednoznačně, najděte ještě další možnosti.

6.

Dokažte, že v každém grafu platí, že každá hrana minimální kostry je tenkou hranou nějakého řezu tohoto grafu. Postupujte sporem.

Řešení Zvolme v minimální kostře T konkrétní hranu e. Nyní zkonstruujeme vhodný řez S. Když z T odstaníme hranu e a ponecháme její koncové uzly, rozpadne se T na dva stromy T_1 a T_2, jejichž sjednocení obsahuje všechny uzly G. Protože T_1 a T_2 jsou evidentně disjunktní, plyne z toho, že množiny uzlů V(T_1) a V(T_2) definují řez, označme ho S_T. Hrana e spojuje řez S_T. Předpokládejme nyní pro dosažení sporu, že hrana e není tenkou hranou řezu S_T, to jest že existuje hrana e_1 spojující S_T, která má menší váhu než e. Protože e_1 spojuje S_T, spojuje také oba stromy disjunktní stromy T_1 a T_2, takže sjednocení $T_1 \cup T_2 \cup e_1$ je kostra grafu G, označme ji T_3. Kostra T_3 vznikla z T tak, že jsme odebrali hranu e a nahradili ji hranou e_1. Protože e_1 má nižší cenu než e, je celková ceny kostry T_3 nižší než cena kostry T. O kostře T jsme ale předpokládali, že je minimální, tudíž naopak cena T_3 nemůže být nižší než cena T, což je hledaný spor.

7.

Hrany minimální kostry jsou podmnožinou množiny L, která je definovaná jako sjednocení tenkých hran všech řezů daného ohodnoceného grafu. Dokažte. Najděte příklad grafu a jeho ohodnocení, při kterém má L více hran než (minimální) kostra grafu.

Řešení Nejjednodušší možností je volit kružnici C_3 a všechny její hrany ohodnotit stejně. Pak je každá hrana tenkou hranou nějakého řezu, celkem má L 3 hrany, zatímco kostra má jen dvě hrany. Obecně postačí například, když v grafu jsou alespoň dvě hrany se stejným minimálním ohodnocením ležící na nějaké kružnici grafu.

8.

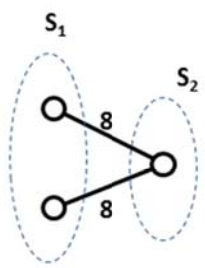
Předpokládejme, že všechny váhy hran v grafu jsou navzájem různé. Potom hrany minimální kostry tvoří právě množinu L z předchozí úlohy. Dokažte.

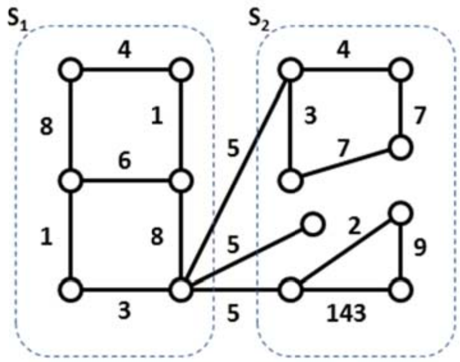
Řešení Protože již víme, že každá hrana minimální kostry je tenkou hranou některého řezu, zbývá dokázat, že okolností daných touto úlohou každá tenká hrana náleží minimální kostře. Budeme tedy postupovat sporem. Předpokládejme že existuje řez S = (S_1, S_2) a jeho tenká hrana *e*, která nenáleží žádné minimální kostře. Zvolme libovolnou z těchto koster a označme ji T. Kdyby v T neexistovala hrana spojující S (tj. hrana mezi S_1 a S_2), pak by v T neexistovala žádná cesta mezi S_1 a S_2 (protože jiné hrany než spojující mezi oběma množinami nevedou) , a pak by T nebyla kostrou, protože v kostře existuje cesta mezi libovolnými dvěma uzly grafu. Víme tedy že v T existuje nějaká hrana e_1 spojující S_1 a S_2. Dále víme, že $e \neq e_1$, protože hrana *e* neleží v T a hrana e_1 v T leží. Navíc víme, že váha(e_1) a váha (*e*) jsou různé. Hrana *e* je tenkou hranou řezu S, tudíž váha e_1 musí být větší, protože podle předpokladu žádné dvě hrany nemají stejnou váhu. Nyní s pomocí hrany *e* zkonstruujeme novou kostru, jejíž váha bude menší než váha T. Protože však předpokládáme, že T je minimální kostra, zároveň tak nalezneme spor s úvodním předpokladem a tvrzení bude dokázáno. T rozdělíme na dva disjunktní stromy T_1 a T_2 tak, že z T odstraníme hranu e1. Z dosavadní konstrukce ihned plyne, že $V(T_1) = S_1$ a $V(T_2) = S_2$. Nyní zbývá spojit stromy T_1 a T_2 vhodnou spojující hranou a použijeme právě hranu *e*. Strom $T_1 \cup T_2 \cup e = T_3$ je opět kostra a díky tomu, že T_3 vznikla z T vyměnou hrany e_1 za hranu *e* s nižší váhou, má T_3 menší váhu než původní minimální kostra T, což je hledaný spor.

9.

Najděte příklad grafu G a jeho ohodnocení tak, že G má jedinou minimální kostru a přitom v G existuje alespoň jeden řez S s alespoň dvěma tenkými hranami.

Řešení Všechny tenké hrany musí být součástí minimální kostry. Nejjednodušší varianta je na levém obrázku, všeobecná povaha řešení snad o něco lépe vystoupí z ukázky napravo.





10.

Dokažte: Pokud má každý řez grafu jedinou tenkou hranu, pak existuje jediná minimální kostra tohoto grafu.

Řešení Samostatně :-).

11.

Nalezli jsme a odevzdali zákazníkovi minimální kostru jím dodaného grafu s mnoha miliony vrcholů. Odpoledne zákazník volá, že v zadání je chyba a že hrana mezi vrcholy 2075154 a 11439446 je ve skutečnosti o 17 % levnější než v jak bylo uvedeno v původní specifikaci.

Veškerá data grafu i kostry jsou dosud na našem disku. Máme určit v lineárním čase vzhledem k počtu vrcholů, to jest ještě během téhož rozhovoru, který platí naše firma(!), zda tato změna ovlivní tvar a cenu minimální kostry, a pokud ano, vydat co možná nejkratší opravu, která se dá opět tlumočit zpět telefonem.

Řešení

Předpokládejme, že graf má *n* vrcholů.

Mohou nastat dva případy: Zlevněná hrana buď je nebo není součástí minimální kostry. Počet hran v kostře je $n - 1$, náležení do kostry lze ověřit v lineárním čase vzhledem k počtu vrcholů prostým průchodem hranami kostry.
Pokud zlevněná hrana je součástí minimální kostry, kostra se nezmění, zlevněná hrana v ní zůstane a pouze celková cena kostry se sníží o rozdíl mezi původní a novou cenou zlevněné hrany.
Pokud zlevněná hrana není součástí původní minimální kostry, je úloha nepatrně složitější. Označme původní kostru T_1, zlevněnou hranu $e = (x, y)$. Musíme rozhodnout, zda máme T_1 ponechat beze změny (například v případě, kdy e je zdaleka nejdražší hranou v grafu i po svém zlevnění) nebo zda hranu e do kostry T_1 zahrnout místo nějaké jiné její dosavadní hrany hrany f. V T_1 existuje mezi x a y jediná cesta, označme ji $P_1(x, y)$. Přidáním hrany e do T_1 se cesta $P_1(x, y)$ změní v kružnici $C_1(x, y)$ (a T_1 přestane být kostrou). Hrana f, kterou máme odstranit (čímž se z T_1 opět stane kostra), je pak zřejmě nejdražší hranou v $C_1(x, y)$.
Úlohou naší implementace je, abychom dokázali kružnici $C_1(x, y)$ rekonstruovat v čase $O(n)$. Nalezení hrany f pak zřejmě proběhne v čase $O(n)$, stačí jen projít všechny hrany $C_1(x, y)$, nahrazení hrany f hranou e by mělo proběhnout v konstantním čase.
Kostru můžeme reprezentovat jako orientovaný kořenový strom, kdy každý uzel kromě kořene obsahuje odkaz na svého bezprostředního souseda (předchůdce) blížšího ke kořeni stromu. Vytvoříme posloupnost P_x všech vrcholů na cestě z vrcholu x do kořene a posloupnost P_y všech vrcholů na cestě z vrcholu y do kořene. To stihneme v čase $O(n)$, díky tomu, že přístup k předchůdci daného uzlu na cestě do kořene máme v čase konstantním.
Dále mohou nastat tři případy. (1) Vrchol x neleží v P_y ani vrchol y neleží v P_x, to jest cesta $P_1(x,y)$ vede přes kořen stromu. (2) Vrchol x je prvkem P_y, to jest x leží na cestě z y do kořene. (3) Vrchol y je prvkem P_x, to jest y leží na cestě z x do kořene .
V případě (1) je kružnice $C_1(x, y)$ dána vrcholy ve sjednocení P_x a P_y. V případě (2) je kružnice $C_1(x, y)$ dána všemi vrcholy v P_y mezi vrcholy y a x včetně. Případ (3) je analogický případu (2).
Prozkoumat P_x a P_y, který z daných případů nastává a uznačit příslušné vrcholy kružnice $C_1(x, y)$, lze zřejmě v čase $O(n)$. Tím je analýza úlohy hotova zbývá napsat příslušný efektivní kód.

12.
Zopakujte analýzu předchozí úlohy pro případ, že původně chybně zadaná hrana se nakonec ukazuje být ve skutečnosti dražší.
Řešení
Pokud uvedená hrana -- označme ji e -- není součástí minimální kostry T_1 (což stejně jako výše rozhodneme v čase $O(n)$, minimální kostra se ovšem nebude nijak měnit.
V opačném případě opět musíme zkoumat, zda po odstranění e z T_1 dokážeme za tuto hranu najít lepší náhradu.
Odstraněním hrany z T_1 se tato kostra rozpadá na dva disjunktní stromy ST_1, ST_2 , jejichž vrcholy pokrývají dohromady celý graf. Sestavit nyní minimální kostru znamená nejlacinější hranu spojující tyto dva stromy, to jest najít nejlacinější hranu řezu $S = (V(ST_1), V(ST_2))$. Počet hran spojujících řez S může být vysoký, až roven $|V(ST_1)| \times |V(ST_2)|$. Pokud například $|V(ST_1)| = |V(ST_2)| = n/2$, pak musíme prozkoumat případně až $n/2 \times n/2 = n^2/4 \in O(n^2)$ hran (když hrany spojující S definují úplný bipartitní graf) , a nezdá se, že bychom tuto horní hranici mohli v obecném případě nějak zlepšit, pokud nebudeme při tvorbě kostry T_1 používat další pomocné struktury registrující jednotlivé řezy.

13.
Předpokládejme, že graf je zadán maticí vah jednotlivých hran. Význačná hodnota v této matici (např. nekonečno, minimální/maximální hodnota číselného typu, NaN apod) indikuje, že mezi příslušnými vrcholy hrana neexistuje. Modifikujte Jarníkův-Primův algoritmus tak, aby nezávisel na počtu hran v grafu a měl složitost $\Theta(n^2)$, kde n je počet vrcholů grafu.
Řešení
1. krok: Zvolíme libovolný vrchol x_1 a označíme ho jako použitý, do fronty vložíme všechny jeho sousedy společně s váhami hran mezi nimi a x_1. To vše má celkem složitost $\Theta(n)$.

k-tý krok pro $1 < k \leq n$: Projdeme linárně všechny vrcholy ve frontě a najdeme ten, jenž je v ní uveden s nejmenší váhou hrany, označme jej x_k a označíme x_k jako použitý.To má složitost $O(n)$.
Dále pozkoumáme všechny sousedy x_k. Pokud je soused x_j nepoužitý, zařadíme jej do fronty s váhou hrany (x_j, x_k). Pokud je použitý aktualizujeme jeho váhu ve frontě podle toho, zda je nebo není menší než jeho dosavadní váha registrovaná ve frontě. Tato část proběhne zcela analogicky jako v klasické variantě algoritmu. Procházíme při ní řádek matice vah odpovídající vrcholu x_k a jen musíme mít navíc zajištěno, že dokážeme sousední vrchol x_j nalézt ve frontě v konstantním čase.To lze implementovat například tak, že pořadí vrcholů ve frontě nikdy nijak neměníme a nezáleží na něm a stačí jen mít u každého vrcholu zaveden další příznak, zda je ve frontě či ne. Procházet frontu pak znamená procházet všechny vrcholy a přeskakovat ty, které nemají nastaven příznak náležení do fronty. Takové procházení frontou má složitost $\Theta(n)$. Díky tomu, že během zkoumání sousedů vrcholu x_k musíme vždy projít celý řádek matice vah, má tento celý k-tý krok složitost $\Theta(n)$.
Dohromady tedy musíme včetně prvého vykonat n kroků, každý se složitostí $\Theta(n)$, celková složitost je pak $\Theta(n^2)$.
Poznámka. V literatuře typicky se tato úloha objevuje s požadavkem modifikovat původní algoritmus tak, aby měl složitost $O(n^2)$. Uvažte, jestli je možné při použití pouze matice vah (a výše zmíněných příznaků nebo funkčně ekvivalentních struktur) v některých případech dosáhnout asymptotické složitosti nižší než $\Theta(n^2)$. Zdá se, že to nejde, že musíme "sáhnout" na každý prvek matice vah, která samotná má n^2 prvků.