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.

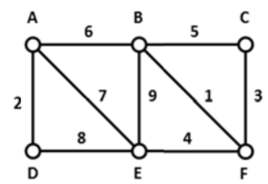
2.
Kolik různých řezů lze vytvořit v grafu s 10 vrcholy?

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?

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.

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_e.



6.
Dokažte, že v každém grafu platí, že každá hrana minimální kostry je tenkou hranou nějakého řezu tohoto grafu. Postupujte sporem.

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.

8.
Předpokládejme, že všechny váhy hran v grafu jsou navzájem různé. Potom hrany minimální kostry tvoří právě množinu L z předchozí úlohy. Dokažte.

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.

10.
Dokažte: Pokud má každý řez grafu jedinou tenkou hranu, pak existuje jediná minimální kostra tohoto grafu.

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.

12.
Zopakujte analýzu předchozí úlohy pro případ, že původně chybně zadaná hrana se nakonec ukazuje být ve skutečnosti dražší.

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.