1.10.27 Nezávislé množiny. Je dán prostý neorientovaný graf $G = (V, E)$ bez smyček. Množina vrcholů $N \subseteq V$ se nazývá *nezávislá množina* v G, jestliže žádná hrana grafu G nemá oba krajní vrcholy v N. Jinými slovy, indukovaný podgraf množinou N je diskrétní graf.

Úloha: Je dán prostý neorientovaný graf G bez smyček a číslo k.
Otázka: Existuje v G nezávislá množina o k vrcholech?

1.10.28 Tvrzení. Platí

$$\text{problém klik} \lhd_p \text{nezávislé množiny.}$$

1.10.29 Převod problému klik na nezávislé množiny. Je dán prostý neorientovaný graf bez smyček $G = (V, E)$. Definujeme opačný graf $G^{op} = (V, E^{op})$ takto:

$$\{u, v\} \in E^{op} \text{ iff } u \neq v \text{ a } \{u, v\} \notin E.$$

Platí: Množina $A \subseteq V$ je klika v grafu G právě tehdy, když je maximální nezávislou množinou v grafu G^{op}. (Jinými slovy, A je nezávislá množina a přidáním libovolného vrcholu už nebude nezávislá.)

To, že se jedná o polynomíální redukci vyplývá z faktu, že všech hran v grafu G i doplňkovém grafu G^{op} je $\frac{n(n-1)}{2}$, kde n je počet vrcholů.

1.10.30 Důsledek. Protože úloha o nezávislých množinách je ve třídě $\mathcal{NP}$, jedná se o $\mathcal{NP}$ úplnou úlohu.

1.10.31 Vrcholové pokrytí. Je dán prostý neorientovaný graf bez smyček $G = (V, E)$. Podmnožina vrcholů $B \subseteq V$ se nazývá *vrcholové pokrytí* G, jestliže každá hrana grafu G má alespoň jeden krajní vrchol v množině B.

Poznamenejme, že celá množina vrcholů V je vrcholovým pokrytím, problém je najít vrcholové pokrytí o co nejmenším počtu vrcholů.

Úloha: Je dán prostý neorientovaný graf G bez smyček a číslo k.
Otázka: Existuje v grafu G vrcholové pokrytí o k vrcholech?

1.10.32 Tvrzení. Platí

$$\text{nezávislé množiny} \lhd_p \text{vrcholové pokrytí.}$$

1.10.33 Nástin převodu nezávislých množin na vrcholové pokrytí. Platí: Je-li množina N nezávislá množina grafu G, pak množina $V \setminus N$ je vrcholovým pokrytím grafu G. A naopak, je-li B vrcholové pokrytí grafu G, pak množina $V \setminus B$ je nezávislá množina v G.

Proto: Je dán prostý neorientovaný graf G bez smyček a číslo k. Pak v G existuje nezávislá množina o k vrcholech právě tehdy, když v G existuje vrcholové pokrytí o $n - k$ vrcholech, kde $n = |V|$ je počet vrcholů grafu G.

1.10.34 Důsledek. Protože problém vrcholového pokrytí je ve třídě $\mathcal{NP}$, jedná se o $\mathcal{NP}$ úplnou úlohu.

1.10.35 Existence hamiltonovského cyklu. Je dán orientovaný graf G.

Otázka: Existuje v grafu G hamiltonovský cyklus? (Jinými slovy, existuje v grafu G cyklus procházející všemi vrcholy?)

1.10.36 Tvrzení. Platí

$$\text{vrcholové pokrytí} \ \lhd_p \ \text{existence hamiltonovského cyklu.}$$

1.10.37 Základní myšlenka převodu. Převod je založen na využití speciálního grafu H o 4 vrcholech a 6 orientovaných hranách. Graf H má tuto vlastnost: Má-li být graf součástí hamiltonovského cyklu, pak jsou jen dva základní způsoby průchodu grafem H, buď se projdou všechny vrcholy za sebou, nebo při dvojím průchodu vždy dva a dva.

Předpokládejme, že je dán neorientovaný prostý graf $G = (V, E)$ bez smyček a číslo k. Je možno vytvořit orientovaný graf G' takový, že v G existuje vrcholové pokrytí o k vrcholech právě tehdy, když v G' existuje hamiltonovský cyklus.

Graf G' se, zhruba řečeno, vytvoří takto: Za každou hranu grafu G do G' dáme kopii grafu H. Kromě takto získaných vrcholů přidáme ještě vrcholy $1, 2, \ldots, k$. Celkově tedy počet vrcholů grafu G je $4\,|E| + k$. Hrany grafu G' jsou jednak hrany všech kopií grafu H, jednak hrany vedoucí mezi nimi a dále hrany do a z vrcholů $1, 2, \ldots, k$. Celkově je hran grafu G' také uměrně počtu hran grafu G plus dvojnásobek počtu vrcholů grafu G.

1.10.38 Důsledek. Protože problém existence hamiltonovského cyklu je ve třídě $\mathcal{NP}$, jedná se o $\mathcal{NP}$ úplnou úlohu.

1.10.39 Tvrzení. Platí

$$\text{existence hamiltonovské kružnice} \ \lhd_p \ \text{problém obchodního cestujícícho.}$$

Převod zmíněný v tvrzení je velmi jednoduchý a je ponechán studentům jako domácí úkol.

1.10.40 Důsledek. Protože problém obchodního cestujícího je ve třídě $\mathcal{NP}$, jedná se o $\mathcal{NP}$ úplnou úlohu.

1.10.41 Heuristiky. Jestliže je třeba řešit problém, který je $\mathcal{NP}$ úplný, musíme pro větší instance opustit myšlenku přesného nebo optimálního řešení a smířit se s tím, že získáme „dostatečně přesné“ nebo „dostatečně kvalitní“ řešení. K tomu se používají heuristické algoritmy pracující v polynomiálním čase. Algoritmům, kde umíme zaručit „jak daleko“ je nalezené řešení od optimálního, se také říká aproximační algoritmy.

1.10.42 Tvrzení. Kdyby existovala konstanta r a polynomiální algoritmus $\mathcal{A}$ takový, že pro každou instanci obchodního cestujícího I najde trasu délky $D \leq r\,OPT(I)$, kde $OPT(I)$ je délka optimální trasy instance I, pak

$$\mathcal{P} = \mathcal{NP}.$$

1.10.43 Zdůvodnění tvrzení 1.10.42. Za předpokladu tvrzení 1.10.42 bychom uměli polynomiálně vyřešit problém existence hamiltonovské kružnice. Naznačíme odpovídající převod.

Je dán neorientovaný graf $G = (V, E)$, $V = \{1, 2, \ldots, n\}$, a ptáme se, zda v něm exituje hamiltonovská kružnice. Zkonstruujeme instanci obchodního cestujícího takto: Pro města $\{1, 2, \ldots, n\}$ položíme

$$d(i,j) = \begin{cases} 1, & \{i,j\} \in E \\ r\,n + 1, & \{i,j\} \notin E \end{cases}$$

Trasa v instanci popsané výše může mít délku n, jestliže je tvořena všemi hranami délky 1. V tomto případě jsou všechny hrany hranami grafu G a trasa představuje hamiltonovskou kružnici. Nebo musí trasa mít délku alespoň $n - 1 + n\,r + 1 = n\,r + n$. To je v případě, že aspoň jedna spojnice v trase není tvořena hranou grafu G.

Tedy jestliže algoritmus $\mathcal{A}$ najde trasu délky jiné než n, pak v grafu G neexistuje hamiltonovská kružnice. Takto bychom polynomiálním algoritmem byli schopni rozhodnout existenci hamiltonovské kružnice. Protože existence hamiltonovské kružnice je $\mathcal{NP}$ úplný problém, platilo by $\mathcal{P} = \mathcal{NP}$.

1.10.44 Trojúhelníková nerovnost. Řekneme, že instance obchodního cestujícího splňuje trojúhelníkovou nerovnost, jestliže pro každá tři města i, j, k platí:

$$d(i,j) \le d(i,k) + d(k,j).$$

1.10.45 Tvrzení. Jestliže instance I obchodního cestujícího splňuje trojúhelníkovou nerovnost, pak existuje polynomiální algoritmus $\mathcal{A}$, který pro I najde trasu délky D, kde $D \le 2\,OPT(I)$ ($OPT(I)$ je délka optimální trasy v I).

1.10.46 Slovní popis algoritmu z tvrzení 1.10.45. Instanci I považujeme za úplný graf G s množinou vrcholů $V = \{1, 2, \ldots, n\}$ a ohodnocením d.

1. V grafu G najdeme minimální kostru (V, K).
2. Kostru (V, K) prohledáme do hloubky z libovolného vrcholu.
3. Trasu T vytvoříme tak, že vrcholy procházíme ve stejném pořadí jako při prvním navštívení během prohledávání grafu. T je výstupem algoritmu.

Zřejmě platí, že délka kostry K je menší než $OPT(I)$. Ano, vynecháme-li z optimální trasy některou hranu, dostaneme kostru grafu G. Protože K je minimální kostra, musí být délka K menší než $OPT(I)$ (předpokládáme, že vzdálenosti měst jsou kladné). Vzhledem k platnosti trojúhelnikové nerovnosti, je délka T menší nebo rovna dvojnásobku délky kostry K.

1.10.47 Christofidesův algoritmus. Jestliže instance I obchodního cestujícího splňuje trojúhelníkovou nerovnost, pak následující algoritmus najde trasu T délky D takovou, že $D \le \frac{3}{2}\,OPT(I)$.

Instanci I považujeme ze úplný graf G s množinou vrcholů $V = \{1, 2, \ldots, n\}$ a ohodnocením d.

1. V grafu G najdeme minimální kostru (V, K).
2. Vytvoříme úplný graf H na množině všech vrcholů, které v kostře (V, K) mají lichý stupeň.
3. V grafu H najdeme nejlevnější perfektní párování P.
4. Hrany P přidáme k hranám K minimální kostry. Graf $(V, P \cup K)$ je eulerovský graf. V grafu $(V, P \cup K)$ sestrojíme uzavřený eulerovský tah.
5. Trasu T získáme z eulerovského tahu tak, že vrcholy navštívíme v pořadí, ve kterém jsme do nich poprvé vstoupili při tvorbě eulerovského tahu.

Platí, že délka takto vzniklé trasy je maximálně $\frac{3}{2}$ krát větší než délka optimální trasy.

1.10.48 Poznámka. Odhad délky trasy, kterou jsme získali v 1.10.46, i odhad pro trasu získanou Christofidesovým algoritmem není možné zlepšit.