4. DÚ Z ADS2

Virtuální cvičení – NP-úplné problémy

Termín **úterý 16. 1. 08:00**. Všimněte si času – v pondělní noci můžete pracovat jak dlouho chcete, ale nezapomeňte to odeslat. Úkoly odevzdávejte emailem na adresu `bohm(at)iuuk.mff.cuni.cz`. Do předmětu napište **ADS2 HW4**.

Prosím odevzdávejte digitálně psané dokumenty, buď ve formátu plaintext nebo v PDF. Při jakýchkoli problémech nebo otázkách se nebojte napsat email a vyřešíme to. Kdybyste s nějakým úkolem zápolili, taky napište.

Nejprve si přečtěte kapitoly `19.1`, `19.2` a `19.3` z knížky *Průvodce labyrintem algoritmů*. Budeme se odkazovat na *katalog NP-úplných problémů*, který najdete v kapitole `19.3`.

PŘÍKLAD PRVNÍ (2 body.) **Chybné úvahy.** Následující dva argumenty jsou chybné; vaším úkolem je ukázat na první místo, kde se stala chyba, a vysvětlit ji.

Ověřování jedním bitem. Ukážeme rychlejší důkaz, že SPLNITELNOST leží uvnitř NP. Bude mi dokonce stačit jeden bit informace – a to, jestli je klauzule splnitelná nebo ne. Polynomiální algoritmus je potom snadný – prostě pro vstupy, kde tento bit je nastavený na jedna, odpoví, že klauzule je splnitelná. Zřejmě je pravda, že pro každou splnitelnou formuli existuje správná nápověda a algoritmus odpoví správně – takže SPLNITELNOST je v NP.

Všechno jde v čase $O(n^c)$. Budeme dokazovat, že kdyby $P = NP$, tak všechny problémy v NP jdou řešit v čase dokonce $O(n^c)$ pro nějakou konstantu c.

Předpokládejme tedy $P = NP$. Pak SPLNITELNOST, což je klasický NP-úplný problém, jde vyřešit v čase $O(n^k)$ pro nějakou konstantu k. No a protože splnitelnost je NP-těžký problém, tak na něj jde převést libovolný jiný problém v polynomiálním čase (čili v $O(n^l)$ pro nějaké l.) Tedy každý další problém jde vyřešit v čase $O(n^{kl})$. Zvolíme tedy konstantu $c = kl$ a pro ni tvrzení platí.

PŘÍKLAD DRUHÝ (2 body.) Vyberte si libovolný problém z katalogu a dokažte, že je NP-úplný převodem z jiného problému v katalogu. *(Vyjímka: nevybírejte si ty, co jsou řešené v knížce, a nevybírejte si* PROBLÉM BATOHU*, viz níže. Je to hlavně cvičení pro vás, nemá smysl ho nějak obcházet.)*

PŘÍKLAD TŘETÍ (2 body.) Vyberte si libovolný problém z katalogu a ukažte, že je převoditelný na SAT přímo, bez použití Cookovy věty.

PŘÍKLAD ČTVRTÝ (2 body.) Dokažte, že PROBLÉM BATOHU je polynomiálně řešitelný, pokud čísla na vstupu jsou zadaná v unární soustavě, a NP-úplný, pokud jsou zadaná klasicky binárně.

PŘÍKLAD PÁTÝ (2 body.) Mějme problém ROZVRHOVÁNÍ:

Vstup: Množina *úkolů*, každý s nějakou délkou zpracování p_i, a také počet počítačů m a globální deadline D.

Úkol: Rozvrhnout všechny úkoly na m počítačů tak, aby se úkoly nedělily, každý počítač zpracovával jen nejvýše jeden úkol v jednom okamžiku, a aby se všechny úkoly stihly do času D. Pokud to nejde, odpovědět záporně.

Dokažte, že problém ROZVRHOVÁNÍ je NP-úplný. Abychom to neměli úplně zadarmo, zkuste redukovat z jiného problému, než DVA LOUPEŽNÍCI. (Rozmyslete si, proč je to z nich zadarmo.)