

Základy kombinatoriky a teorie grafů

Cvičení #11 – P vs. NP

Příklady

Jsou to jenom přepsané příklady z Průvodce labyrintem algoritmů, pořádné vysvětlení a často i hinty nebo řešení najdete tam.

1. Tahle úloha se trochu špatně zadává přesně, takže to bude delší. Kdyby vám to přesto nebylo jasné, napište mi mail s dotazy :). Turingův stroj má jednu nekonečnou pásku (pozice jsou, řekněme, očíslované přirozenými čísly), na níž se nacházejí nuly a jedničky. Vstup začíná na pozici nula, někde končí a dál pokračují samé nuly.¹ Často je potřeba na vstupu zadat přirozené číslo n předem neznámé velikosti. Jak to udělat, pokud můžete použít $2\lfloor \log_2(n) \rfloor + c$ bitů, kde c je nějaká konstanta? Uměli byste to s $\lfloor \log_2(n) \rfloor + o(\log(n))$ bity?

Správné řešení není napsat prostě na pásku binární kód čísla, protože pak nelze poznat, kde končí číslo a kde začínají nuly značící prázdné pozice. Správné řešení není ani napsat binární kód a zakončit ho jedničkou – Turingův stroj nemá v konečném čase možnost poznat, jestli poslední jednička, kterou přečetl, je ta ukončovací, anebo jestli ještě někde napravo od aktuální pozice budou další data.

2. Převeďte SAT \rightarrow 3-SAT. Jinak řečeno, najděte polynomiální algoritmus, který na vstupu dostane CNF formulí φ (např. $(x_1 \vee x_7 \vee \neg x_9 \log x_1 2) \wedge (x_1 \vee \neg x_7) \wedge \dots$) a vyrobí jinou CNF formulí ψ (klidně s jinými proměnnými), v níž má každá klauzule nejvíše tři literály (tj. $(x_1 \vee \neg x_2 \vee x_3)$ je povolené, $(x_1 \vee x_2 \vee x_3 \vee \neg x_4)$ není) a která je splnitelná právě tehdy, když φ je (tj. existuje přiřazení proměnných φ takové, že s ním je φ pravda, právě když existuje přiřazení proměnných ψ s nimiž je ψ pravda).
3. Převeďte 3-SAT na nezávislou množinu. Tj. najděte polynomiální algoritmus, který na vstupu dostane 3-CNF formulí φ a vyrobí graf G a číslo k takové, že G obsahuje nezávislou množinu velikosti (alespoň) k , právě když φ je splnitelná.
4. Převeďte nezávislou množinu na SAT. Tj. najděte polynomiální algoritmus, který na vstupu dostane graf G a číslo k a vyrobí CNF formulí φ , která je splnitelná, právě když G obsahuje nezávislou množinu velikosti k .
5. Problém kliky dostane na vstupu graf G a číslo k a ptá se, zda G obsahuje kliku (úplný graf) velikosti alespoň k . Převeďte problém kliky na problém nezávislé množiny a naopak.
6. Předpokládejte, že máte k dispozici černou skříňku, která umí vyřešit problém kliky v čase 1. Vyřešte v polynomiálním čase s pomocí (opakování) volání této skříňky problém, kdy na vstupu dostanete graf G a máte zjistit velikost největší kliky v G .
7. 3,3-SAT je další omezení 3-SATu, kde je navíc slíbené, že každá proměnná se vyskytuje v nejvíše třech klauzulích (at' už v negaci nebo gaci). Převeďte 3-SAT na 3,3-SAT.
8. E3,E3-SAT je omezení 3,3-SATu, kde navíc slibujeme, že každá klauzule má právě tři literály a každá proměnná se vyskytuje v právě třech klauzulích. Dokažte, že E3,E3-SAT je polynomiálně řešitelný, a to proto, že každá taková formule je ve skutečnosti splnitelná.

¹Tohle je jedna možnost, jak zavést Turingův stroj, tady to píšu jen kvůli motivaci problému.

Henty

Stránky znamenají strany v Průvodci labyrintem algoritmů, konkrétně ve verzi na stránce <http://pruvodce.ucw.cz/static/pruvodce.pdf>.

1. Bylo by fajn mít k dispozici větší abecedu než $\{0, 1\}$, třeba $\{0, 1, \clubsuit\}$, kde \clubsuit znamená „tady končí číslo“. Obecně můžete abecedu s 2^k znaky zapsat pomocí $\{0, 1\}$ tak, že každému znaku původní abecedy bude odpovídat k znaků té binární. Zlepšit to pak můžete tím, že nejdřív zapíšete pomocí téhle myšlenky počet bitů čísla a potom to samotné číslo už jen základní abecedou.
2. str. 435
3. str. 436
4. str. 437
5. str. 438
6. str. 438
7. Vyrobte si bipartitní graf, kde vlevo jsou proměnné, vpravo klauzule a použijte Hallovu větu.