TIN - Teoretická informatika
2021/ 2022

3. Domácí úloha

Vypracoval: Jan Lorenc (xloren15) Datum: 2. 1. 2022

Teoretická informatika (TIN) – 2021/2022
Úkol 3
(max. zisk 5 bodů – 10 bodů níže odpovídá 1 bodu v hodnocení předmětu)

1. Strýc Rudolf organizuje osmdesátiny dědy Adolfa. Oslava pro početnou rodinu bude v sále s mnoha stoly s různým počtem míst. Příbuzní Rudolfovi hlásí své dary pro dědu, někteří více, než jeden. Mnozí však hlásí stejné dary. Aby se předešlo trapným situacím, plánuje Rudolf rozesadit příbuzné ke stolům tak, aby u žádného neseděli příbuzní se stejným darem. Děda pak každý stůl navštíví ve vhodný okamžik, kdy bude od ostatních stíněn Rudolfovým hlasitým zpěvem. Zároveň je důležité, aby dědův stůl byl plně obsazen. Návrh zasedacího pořádku je tedy *Rudolfův problém*. Rudolf jr. složil zkoušku z IZP a tvrdí, že to nacéčkuje, a bude.

   Zařaďte co nejpřesněji Rudolfův problém do složitostní třídy. Jak dobře může fungovat řešení Rudolfa jr.?

   Nápovědu najdete zde:
   `https://en.wikipedia.org/wiki/List_of_NP-complete_problems` 20 bodů

2. Mějme jazyk $L \in DTIME(n^3)$ nad abecedou Σ. *Superpalindrom jazyka* L je slovo formy $a_1 \cdots a_n a_n \cdots a_1, n \in \mathbb{N}$ takové, že pro všechna $i : 1 \leq i \leq n$, $a_i \cdots a_n a_n \cdots a_i \in L$.
   Nechť M je jazyk slov w, která obsahují podslovo v délky alespoň $|w|/2$, jež je superpalindromem L, tedy $M = \{w = x.v.y \mid v \text{ je superpalindrom } L, |v| \geq |w|/2, x, y \in \Sigma^*\}$.
   Příkladem slova z jazyka M je $aaaa\,bcddcbbb$, za předpokladu, že $bcddcb$, $cddc$, dd i ϵ patří do L.
   Najděte co nejmenší $k, \ell, m \in \mathbb{N}$, aby platila následující tvrzení, a zdůvodněte, že platí:

   (a) $M \in DTIME(n^k)$. (pozor, nejpřímočařejší řešení není optimální)
   (b) $M \in NTIME(n^\ell)$.
   (c) $M \in DSPACE(n^m)$.

   Nemusíte dokazovat, že Vaše volby k, ℓ, m jsou opravdu optimální.

   20 bodů

3. Dokažte, že pro jakékoliv dané $p, m, k, \ell \in \mathbb{N}$ je

   (a) $\mathcal{O}(\log_k(n^p)) = \mathcal{O}(\log_\ell(n^m))$ (zopakujte si pravidla pro počítání s logaritmy)
   (b) $\mathcal{O}(n^k) \subset \mathcal{O}(n^{k+1})$ (ukažte oba vztahy, $\subseteq$ i $\not\supseteq$) 10 bodů

Příklad 1

Problém spadá do složitostní třídy NP.
Zasedací pořádek, počet stolů a míst lze nedeterministicky „uhádnout" a jsme tak schopni sestrojit NTS řešící problém v polynomiálním čase. NP úplnost je pak možné dokázat redukcí na některý NP-úplný problém.

Fungování nedeterministického TS pro Rudolfův problém:
Předpokládejme jazyk L reprezentující Rudolfův problém. NTS M poté pracuje následovně:

1. M ověří, zda jeho vstup je korektní instancí jazyka L, a odmítne, pokud není.
2. M nedeterministicky zvolí počet stolů, míst a následně i zasedací pořádek u nich.
3. M ověří, jestli jsou splněny obě podmínky (nesedí spolu příbuzní se stejným darem a dědův stůl není prázdný). Jsou-li splněny, přijme, jinak odmítne.

Jednotlivé kroky NTS M lze provést v polynomiálním čase, proto i M pracuje v polynomiálním čase.

Idea redukce na NP-úplný problém barvení vrcholů grafu:

- Nechť množina vrcholů grafu je množina příbuzných A, jejíž prvky a_i jsou podmnožiny množiny darů B (zjednodušeně máme příbuzné a každý příbuzný má 0-N darů).
- Hrany grafu lze definovat jako:
  $$H = \{(a_1, a_2) \in A \times A \mid \exists b \in B: b \in a_1 \land b \in a_2\}$$
  (zjednodušeně hrany jsou mezi vrcholy = příbuznými, kteří mají nějaký stejný dar)
- Za barvení vrcholů pak lze považovat zobrazení $f: A \rightarrow C$, kde C je množina stolů, pro které platí:
  $$\forall (a_1, a_2) \in H: \ f(a_1) \neq f(a_2)$$
  (zjednodušeně žádní příbuzní se stejným darem = hrany nesmí sedět u stejného stolu)
- Za barvy lze poté brát jednotlivé stoly a počet jejich výskytů jsou počty míst u patřičných stolů.

Lze nahlédnout, že tato transformace je proveditelná úplným DTS pracujícím v polynomiálním čase se zachováním členství v jazyce.

Řešení Rudolfa jr.:
Cčkový program Rudolfa jr. v polynomiálním čase fungovat nebude, neboť nemůže využít nedeterminismu. Program musí prozkoumat všechny možné zasedací pořádky. Počet těchto kombinací rozesazení však roste exponenciálně, a proto bude jeho řešení spadat do třídy EXP.

Příklad 2

Uvažujme existenci *k*-páskového TS T_L přijímajícího jazyk L, jenž pracuje v čase $O(n^3)$. Takový musí existovat, neboť $L \in DTIME(n^3)$.
Dále diskutujeme, jak by mohl pracovat (*k*+2)-páskový TS T (v případě b) NTS) z pohledu složitosti.
(1. páska vstup, 2. páska pomocná na kontrolu náležitosti volených podslov do L, zbylých k pásek pro simulaci T_L)

<u>a)</u>
Přímočaré neoptimální řešení:

- Vybrání každého podslova *v* (všech kombinací) dané délky: $O(n^2)$
- Každé je testováno na superpalindrom z L:
    - Musí se ověřit nejen dané podslovo *v*, ale i jeho palindromická podslova: O(n)
    - Ověření náležitosti do L strojem T_L: $O(n^3)$

$O(n^2) * O(n) * O(n^3) = O(n^6)$

První bod, tedy počáteční vybírání, lze zjednodušit díky znalosti, že $|v| \geq |w|/2$. Dále víme, že delší superpalindromy obsahují ty kratší (zbytečné hledat delší podslova). Proto stačí lineární průchodem najít pouze podslova délky $|w|/2$ ($|w|/2 + 1/2$ v případě lichého $|w|$), což lze provést v čase O(n).

Výsledná složitost je tedy $O(n) * O(n) * O(n^3) = O(n^5)$, tedy $M \in DTIME(n^5)$ a $k = 5$.

<u>b)</u>
Poněvadž M je rozhodován nedeterministickým TS, můžeme oproti příkladu a) vybírat podslova nedeterministicky, tedy v čase O(1). Vše ostatní je stejné.

Výsledná složitost je tedy $O(n) * O(n^3) = O(n^4)$, tedy $M \in NTIME(n^4)$ a $l = 4$.

<u>c)</u>
Délka vstupu: O(n)   … první páska
Délka pomocné pásky pro simulaci TS T_L nad zvoleným slovem: O(n)   … druhá páska
Prostorová složitost TS T_L: $O(n^3)$ … zbylých *k* pásek pro T_L
(Není známo, jak TS T_L pracuje, proto neznáme jeho prostorovou složitost. Víme však, že ta asymptoticky není nikdy větší než časová díky větě z přednášky: „Je-li časová složitost výpočtu prováděného TS rovna n, pak prostorová složitost tohoto výpočtu není větší než n+1.")

Výsledná složitost je tedy $O(n) + O(n) + O(n^3) = O(n^3)$, tedy $M \in DSPACE(n^3)$ a $m = 3$.

Příklad 3

a)

Důkaz lze provést aplikací logaritmických pravidel k dosažení stejného logaritmu vynásobeného odlišnou konstantou, která již však asymptoticky není významná, tedy složitosti se rovnají.

$$log_k(n^p) = \mathrm{p} * log_k(\mathrm{n}) = p * \frac{\log(\mathrm{n})}{\log(\mathrm{k})} = \frac{\mathrm{p}}{\log(\mathrm{k})} * \log(\mathrm{n}) \rightarrow O(\log(\mathrm{n}))$$

$$log_l(n^m) = \mathrm{m} * log_l(\mathrm{n}) = m * \frac{\log(\mathrm{n})}{\log(\mathrm{l})} = \frac{\mathrm{m}}{\log(\mathrm{l})} * \log(\mathrm{n}) \rightarrow O(\log(\mathrm{n}))$$

b)

Pro dokázání ⊆ stačí vzít n^k, ukázat, že skutečně patří do $O(n^k)$ a že patří i do $O(n^{k+1})$.

Pro dokázání ⊉ naopak vezmeme n^{k+1}, ukážeme, že skutečně patří do $O(n^{k+1})$, ale již nepatří do $O(n^k)$.

Důkaz ⊆:

Je zřejmé, že n^k patří do $O(n^k)$, uvažme například n_0=1 a c=1, poté zřejmě platí:

$$\forall n \geq n_0 \colon f(n) \leq c * n^k$$

Tedy:

$$\forall n \geq 1 \colon n^k \leq n^k$$

Nyní ukážeme, že n^k patří i do $O(n^{k+1})$. Uvažme opět n_0=1 a c=1, poté zřejmě platí:

$$\forall n \geq 1 \colon n^k \leq n^{k+1}$$

Důkaz ⊉:

Podobně jako výše lze ukázat, že n^{k+1} patří do $O(n^{k+1})$. Uvažme opět n_0=1 a c=1, poté zřejmě platí:

$$\forall n \geq 1 \colon n^{\mathrm{k}+1} \leq n^{\mathrm{k}+1}$$

Naopak n^{k+1} již nepatří do $O(n^k)$. Z definice O() stačí ukázat, že:

$$\lim_{n\to\infty} \frac{c * n^k}{f(n)} = 0$$

Stále při c=1 a $f(n)=n^{k+1}$ zde dostáváme:

$$\lim_{n\to\infty} \frac{n^k}{n^{k+1}} = \lim_{n\to\infty} \frac{1}{n} = 0$$

Bylo tedy dokázáno, že $O(n^k) \subseteq O(n^{k+1})$ a $O(n^k) \not\supseteq O(n^{k+1})$, tedy že $O(n^k) \subset O(n^{k+1})$.