MAROŠ ORSÁK

XORSAK02

Teoretická informatika (TIN) – 2020/2021
Úkol 3
(max. zisk 5 bodů – 10 bodů níže odpovídá 1 bodu v hodnocení předmětu)

1. Ferko Amundsen se rozhodl navštívit Grónsko. Vytipoval si n slavných grónských měst, která chce navštívit. Chtěl by přiletět do kteréhokoliv z nich a všechna ostatní z něj projet, tradičně na psím spřežení, tak, aby v žádném městě nebyl dvakrát. Odletět by chtěl z posledního n-tého města na své trase. Aby viděl krásy Grónska, ohodnotil si každou cestu mezi dvěma městy podle toho, jaký na ní uvidí hezký přírodní úkaz (například nekonečnou bílou planinu) nebo zajímavou památku (velmi starý sníh). Cesty, kde se nic zajímavého nenachází, ohodnotil nulou. Ne mezi každými dvěma městy existuje cesta. Ferko by nejraději viděl alespoň polovinu grónských krás, proto si chce naplánovat trasu tak, aby součet skóre cest mezi městy na trase měl hodnotu alespoň poloviny součtu skóre všech krás Grónska. Chtěl by nejdříve zjistit, zda taková trasa vůbec existuje. Formalizujte problém Ferka Amundsena, a ukažte, že je NP-úplný.

   Pomůže Vám NP-úplnost některého z problémů uvedených zde:
   `https://en.wikipedia.org/wiki/NP-completeness#NP-complete_problems`
   v odstavci „*NP-complete problems*“.

   20 bodů

2. Mějme dán jazyk L na abecedou $\Sigma = \{0, 1\}$. Na jeho základě definujme jazyk L' jako

   $$L' = \{w \in \Sigma^* \mid \exists u, v : u \neq v \wedge u \sqsubseteq w \wedge v \sqsubseteq w \wedge u \in L \wedge v \in L\} \ ,$$

   kde $\sqsubseteq$ je relace býti podslovem (L' je jazyk všech slov, která mají dvě různá podslova z L). Snažte se najít co nejmenší k takové, aby platilo

   $$L \in DTIME[n^2] \Rightarrow L' \in DTIME[n^k] \ .$$

   Dokažte, že vztah platí, ale nedokazujte, že vaše k je opravdu nejmenší možné. 15 bodů

3. Dokažte, že jazyk L všech slov nad abecedou $\Sigma = \{0, 1\}$, která obsahují přesně 2^n nepřekrývajících se výskytů nějakého neprázdného slova nad Σ, náleží do PSPACE. Pro úplnost, L můžeme definovat formálně jako

   $$L = \{u_1\, w\, u_2\, w \cdots u_{2^n}\, w\, u_{2^n+1} \mid w \in \Sigma^+, n > 0, u_1, \ldots, u_{2^n+1} \in \Sigma^* \setminus (\Sigma^* w \Sigma^*)\} \ .$$

   Pomůže zopakovat si fakta o třídě PSPACE.

   15 bodů

Při popisech Turingových strojů v rámci řešení tohoto úkolu se omezte na stručný, přesný, a přehledný popis principu jejich funkce. Nevypisujte formálně přechodovou funkci ani nekreslete detailní diagram.

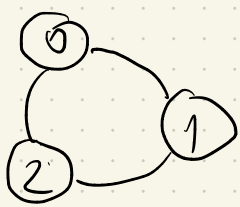
1.

Formulácia problému:

- Problém Ferka Amundsena môžeme charakterizovať jazykom $FA = \{\langle G \rangle \mid G \text{ je neorientovaný graf}\}$, kde $\langle . \rangle$ je vhodné kódovanie (G) napr. také, že:

  - graf zakódujeme ako sekvencia čísel uzlov (bez repetície) nasledovaných sekvencií kódov hrán v podobe dvojíc čísel uzlov (ide o čísla, ktoré sú sekvenciou čísel uzlov), a to bez repetície, všetko vhodne oddelené. Čísla zapíšeme binárne

- napr.:

$\langle$  $\rangle$ = 00|01|10|(00,01)|(01,10)|(10,00)|

Dokážeme, že FA je NP-úplný problém:

I. FA ∈ NP

1) Zkonštruujeme NTS M, ktorý prijíma jazyk FA, a to polynomiálnom čase.
M bude pracovať nasledovne:

a) M overí, či jeho vstup je platná inštancia FA - ak nie odmietne

1. kontrola správneho poradia oddeľovačov
2. prejde kódy uzlov a overí, že nie sú duplikované
3. analogicky overí, že vstupný inštancii nie sú duplikované hrany či hrany medzi ne-exis- tujúcich uzlov

b) "guess": M nedeterministicky uhádne cestu správneho začiatočného miesta do koncového miesta na pomocnú pásku

c) "check": M overí, či zvolil cestu ktorých ohodnotenie je aspoň polovica z celkového ohodnotenia hrán. Ak áno prijme, inak odmietne.

2) Analýza zložitosti M, či pracuje v polynomiálnom čase.

a) Zložitosť:

1. načíta jeden prechod pásky $O(n)$
2. prejde všetkých $O(n)$ kódov uzlov, každý si prepíše na pomocnú pásku $O(n)$ a overí s kódmi uzlov od neho napravo $O(n)$. Jedna kontrola $O(n)$. Celkovo dostaneme:

$$O(n)\cdot(O(n)+O(n)\cdot O(n)) = O(n^3)$$

3. analogicky ako 2. krok

b) Zložitosť:

- vyberáme cestu, ktorá pozostáva z $O(n)$ uzlov, pretože nemôže mať viac než $O(n)$ uzlov

c) Zložitosť:

- M overí, či vybraná cesta tvorí riešenie problému. Pri tomto overovaní sa budú jednotlivé cesty zapisovať na pomocnú pásku, kde sa všetky cesty sčítajú. $O(n)$

• M teda pracuje v polynomiálnom čase.

II. FA je NP-hard:

- dôkaz vykonáme polynomiálnou redukciou z problému Hamiltonovskej dráhy (Hamiltonian Path Problem), ktorý je NP-úplný

Problém HPP je definovaný nasledovným spôsobom:

- problém je s určením či v danom grafe existuje Hamiltonovská cesta,
- Hamiltonovská cesta je cesta v grafe, kde práve navštívim každý vrchol práve raz

Danú redukciu je možné implementovať UDTS, ktorý pracuje v polynomiálnom čase:

- kde jednotlivé problémy HPP je možné previesť na problém FA nasledovne:

- graf ktorý dostaneme bude modifikovaný kde každú hranu medzi uzlami bude 0.
- keďže suma všetkých hrán je 0 tak triviálne splníme podmienku FA s hranou aspoň polovici súčtu všetkých hrán grafu
- touto modifikáciou zmeníme problém FA

Toto neznámkovať, zaujímalo by ma či to ide dokázať aj takto

II. FA je NP-hard:

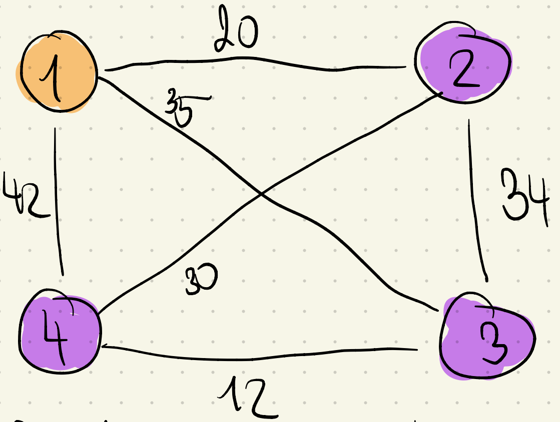
- dôkaz vykonáme polynomiálnou redukciou z problému obchodného cestujúceho (Traveling Salesman Problem), ktorý je NP-úplný

Problém TSP je definovaný nasledujúcim spôsobom:

- problém je nájsť minimálnu vzdialenosť prekonanú obchodného cestujúceho, ktorý chce cestovať po danej množine miest, precestovať každé miesto práve raz a vrátiť sa na začiatok.

1) Idea

- predstavme si, že chceš cestovanie pozostáva zo 4 miest. Cena (vzdialenosť) na cestu z jedného mesta do druhého je uvedená na nasledujúcej matici. Predpokladajme, že cena i-teho uzlu po j-ty uzol je rovnaká ako od j-teho po n-ty.



	1	2	3	4
1	0	20	35	42
2	20	0	34	30
3	30	34	0	12
4	41	30	12	0

Predpokladajme, že cesta začína aj končí v uzle 1. Z uzla č. 1 môže cestujúci pokračovať 2, 3 alebo 4 uzlom. Vo všeobecnosti, môžeme povedať že cesta prechádza z hrán $\langle 1, k \rangle$ kde $k \in V - \{1\}$ (V je množina všetkých vrcholov) a cesty z k do 1. Zároveň, obchodný cestujúci môže prejsť cez mesto iba raz. Kde cestu z k do 1 musí prechádzať cez všetky vrcholy v $V - \{1, k\}$ a táto cesta musí byť najkratšia.

Nech dĺžka naj-kratšej cesty začínajúcej z uzla i a pokrývajúcej všetky uzly presne raz a končí na uzle **I** bude $1(i,S)$. Ak $i=1$, potom funkcia nadobudne formu:

$$1(1, V-\{1\}) = \min_{2\le k\le n}\{c_{1k} + 1(k, V-\{1,k\})\}$$

kde c_{1k} je cena uzla **1** do uzla **k**. Vo všeobecnosti to môžeme zapísať ako:

$$1(i,S) = \min_{j\notin S}\{c_{ij} + 1(j, S-\{j\})\}$$

kde $i \notin S$.

Ak $n=4$, $i=1$ tak potom je $S=\{2,3,4\}$. Takže

$$1(1,\{2,3,4\}) = \min\{[c_{12} + 1(2,\{3,4\})], [c_{13} + 1(3,\{2,4\})], [c_{14} + 1(4,\{2,3\})]\}$$

$\vdots$

vhodnými úpravami sa dostaneme k:

$$1(1,\{2,3,4\}) = \min\{[20,72],[35,62],[42+66]\}$$

$$= \min\{92,97,108\} = \underline{\underline{92}}$$

Ak je element množiny S je 3, potom sú 3 rozličné rozhodnutia pre i. Ak $|S|$ je 2 tak sú 2 rozličné rozhodnutia pre i.
Vo všeobecnosti, pre každú hodnotu $|S|$ je tu $n-1$ rozličných rozhodnutí pre i. Ne-rátame 1 a I, počet rozličných množín veľkosti k môže byť skonštruovaný ${}^{n-2}C_k$. Počet krokov požadovaných pre TSP uzlov n je:

$$\boxed{\sum_{k=0}^{n-2} (n-1)\left({}^{n-2}C_k\right) = (n-1)\sum_{k=0}^{n-2} {}^{n-2}C_k = (n-1)\left[{}^{n-2}C_0 + {}^{n-2}C_1 + \dots + {}^{n-2}C_{n-2}\right]}$$

- vieme, že pomocou úprav dosiahneme výraz $(n-1)2^{n-2}$ a teda zložitosť problému TSP je $O(n \cdot 2^n)$ a je to NP problém.

Redukciu je možné realizovať UDTS, ktorý pracuje v pol. čase.

Každú inštanciu problému TSP je možné previesť na problém FA nasledovne:

- budeme modifikovať funkciu min na max a teda budeme hľadať maximálnu vzdialenosť prekonanú obchodného cestujúceho, ktorý chce cestovať po danej množine miest, precestovať každé mesto práve raz a pri konci nebude sa vrátiť naspäť na štart teda ohodnotenie hrany z konca na štart bude 0.
- týmto vznikne jazyk FA, a teda problém Fedka Amundsena

② Dôkaz konštrukciou TS

1. TS T dostane na vstupe ⟨x⟩
2. Prekopíruje si vstup ⟨x⟩ na 1. pásku a začne predvádzať dané ⟨x⟩.
3. Začne tvoriť podreťazce a postupne si ich všetky ukladať na 2. pásku
4. Začne predvádzať 2. pásku a overiť či špecifický podreťazec nepatrí do L tak skúša ďalší. Ak daný podreťazec patrí L tak si to uloží na 3. pásku a pokračuje krokom 5.
5. Začne predvádzať 2. pásku a overiť či špecifický podreťazec nepatrí do L tak skúša ďalší. Ak daný podreťazec patrí L tak si to uloží na 4. pásku a pokračuje konečným overením či sa dané nájdené podreťazce na 3. páske a

4. pásku rovnajú. Ale nie konečne vytvoril a musel som 2 podreťazce, ktoré sú odlišné a patria do L. Inak znova opakujem krok 5.

Celková zložitosť:

$$O(n) + O(n^2) + O(n^4) + O(n^4) \cdot O(n) = O(n^5)$$

$k = 5$

3. Dokažte, že jazyk L všech slov nad abecedou $\Sigma = \{0, 1\}$, která obsahují přesně 2^n nepřekrývajících se výskytů nějakého neprázdného slova nad Σ, náleží do PSPACE. Pro úplnost, L můžeme definovat formálně jako

$$L = \{u_1 \underline{w}\, u_2 \underline{w} \cdots u_{2^n}\, w\, u_{2^n+1} \mid w \in \Sigma^+, n > 0, u_1, \ldots, u_{2^n+1} \in \Sigma^* \setminus (\Sigma^* w \Sigma^*)\}\,.$$

Pomůže zopakovat si fakta o třídě PSPACE.

15 bodů

Při popisech Turingových strojů v rámci řešení tohoto úkolu se omezte na stručný, přesný, a přehledný popis principu jejich funkce. Nevypisujte formálně přechodovou funkci ani nekreslete detailní diagram.

3. Pre predstavivosť (príklad)

$w = 01$

$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111 \ldots\ldots\}$

$u_1, \ldots, u_{2^n+1} = \{\varepsilon, 0, 00, 10, 11, 000, 100, 110, \ldots\}$

$u_1 w u_2 w \ldots\ldots u_{2^n} w\, u_{2^n+1} = \{\varepsilon 01\, 0\, 01\, 00\, 01 \ldots$

$$PSPACE = \bigcup_{k=0}^{\infty} DSPACE(n^k)$$

S_M – značí priestorovú zložitosť

$s : \mathbb{N} \to \mathbb{N}$ je zobrazenie

$DSPACE[s(n)] = \{L \mid \exists$ k-páskový DTS $M : L = L(M)$ a $S_M \in O(s(n))\}$

$$NPSPACE = \bigcup_{k=0}^{\infty} NSPACE(n^k)$$

$NSPACE[s(n)] = \{L \mid \exists$ k-páskový NTS $M : L = L(M)$ a $S_M \in O(s(n))\}$

Zároveň je dôležité si uvedomiť, že PSPACE a NPSPACE sú ekvivalentné a platí tu Savitchova teorém Veta 7.3.3.

Riešenie Dôkaz konštrukciou:

1. Zkonštruujem NTS M, ktorý urobí "guess" a nájde požadovaný exponent k $O(1)$
2. Následne si NTS M bude generovať všetky podslová na pomocnej páske $O(n^2)$
3. Nakoniec budem počítať 2^k slov w pre každý podreťazec.
4. Ak počet nájdených slov bude rovný 2^k tak k bude uhádnutý číslo exponentu tak prijímam
5. Inak neprijímam.

- Je zjavné, že daný NTS bude mať časovú zložitosť M je polynomiálna. Z toho vyplýva, že priestor musí byť najmenej polynomiálna. Zároveň platí Savitchov teorém, kde vieme že NPSPACE = PSPACE a teda NTS M je dôkazom, že L je triede PSPACE.