Úkol č. 2 do předmětu Složitost

Vojtěch Havlena (xhavle03)

1. **Příklad**

Časová složitost Pro jednoduchost uvažujme, že $n = |V|$. Cyklus na řádcích 2. – 3. se provede celkem n-krát. Zápis položky do pole se provede s konstantní složitostí. Pro kód na řádcích 2. – 3. máme tedy celkem časovou složitost $O(n)$. Kód na řádku 4. (zápis do pole) se provede s konstantní složitostí.

Co se týče složistí práce se zásobníkem, tak uvažuji implementaci zásobníku v podobě jednosměrného vázaného seznamu s ukazatelem na vrchol zásobníku. Vytvoření je tedy možné v konst. čase (přiřazení ukazateli na vrchol zás. hodnotu `NULL`). Přidání prvku je možné také v kons. čase (vytvoření nového prvku, nastavení ukazatele na vrchol, nastavení uk. vrcholu zásobníku na nový prvek). Podobně odebrání prvku je možné také v konst. čase (vrchol nastavím na další prvek, smažu prvek na který ukazoval vrchol před tím). Zjištění, zda je zás. neprázdný lze opět provést v konst. čase (porovnání jestli vrchol zás = `NULL`).

Kód na 5. řádku se tedy provede s časovou složitostí $O(1)$. Vzhledem k tomu, že do zásobníku jsou vkládány vrcholy, které mají stav nastavený na `FRESH` a před samotným vložením do zásobníku jsou nastaveny na `OPEN` a už nikdy nejsou nastaveny zpět na `FRESH`, každý vrchol je vložen do zásobníku nejvýše jednou (v případě nesouvislého grafu tam nemusí být některé vrcholy vloženy vůbec). Dále uvažuji, že vstupní graf je zadán seznamem sousedů a pokud si označím $Adj[v]$ jako seznam vrcholů, které jsou s vrcholem v spojeny hranou, platí:

$$\sum_{v \in V} |Adj[v]| \leq 2|E|$$

Cyklus na řádcích 8. – 11. se tedy provede celkem až $2|E|$-krát (v případě orientovaných grafů až $|E|$-krát) – Kód na řádcích 9. – 11. se provede s konst. složitostí. Nakonec výběr ze zásobníku na ř. 7. a nastavení stavu na ř. 12. se tedy celkem provede až n-krát. Díky tomu, že $|E| \leq n^2$ (rovnost nastává pro orientované úplné grafy), je celková složitost vzhledem k počtu uzlů n: $O(n+n+|E|) = O(n+n+n^2) = O(n^2)$.

Prostorová složitost V algoritmu jsou využity 3 pole (`state`, `d`, `p`) o velikosti počtu vrcholů grafu n. Jak bylo uvedeno výše, do zásobníku je vložen každý vrchol nejvýše jedenkrát, velikost zásobníku je tedy v nejhorším případě n. Celková prost. složitost zásobníku bude $a(n + 1)$, kde 1 je tam pro vrchol zásobníku a a v sobě zahrnuje prost. složitost ukazatele na další prvek a místo pro uložení hodnoty. Celková prostorová složitost tedy bude $O(n)$.

2. Příklad Nejprve důkaz toho, že $EXACT\_LENGTH \in NP$. Pro zadanou instanci (G, X) bude NTS M nedeterministicky tvořit cestu v G (stačí pouze posloupnost navzájem různých vrcholů mezi kterými existuje hrana, nejedná se o multigraf). Těchto vrcholů na dané cestě může být nejvýše $|V| = n$. Generování cesty podrobněji probíhá takto: Nejprve se nedeterministicky zvolí počáteční uzel. Poté se ned. zvolí jestli se bude pokračovat v generování. Pokud ano, ned. se zvolí uzel, do kterého vede z předchozího uzlu hrana (procházení množiny hran, složitost $O(|E|) = O(n^2)$. Tento postup se opakuje maximálně do vygenerování posloupnosti n vrcholů a $n-1$ hran. Dále se zkontroluje, zda se vygenerované uzly neopakují (složitost $O(n^2)$). Pokud se opakují, M zamítne. NTS M nakonec jen sečte váhy hran a porovná s hodnotou X. Pokud se shoduje, přijme, jinak zamítne. Uvedený NTS M tedy pracuje v polynomiálním čase.

Redukci v tomto případě provedu z problému $SUBSET\_SUM$. Předpokládejme, že (S, w, W), kde S je konečná množina prvků, w je váhová funkce $w : S \to \mathbb{Z}$ a W je požadovaná váha. Převod na problém $EXACT\_LENGTH$ provedeme následovně (redukce R). Každá hrana v grafu bude mít váhu jako některý prvek z S. Vzhledem k tomu, že se v problému $EXACT\_LENGTH$ hledá orientovaná cesta (nemůžou se tedy opakovat vrcholy), budeme potřebovat alespoň $|S| = n$ hran (tedy $n+1$ vrcholů), abychom byli schopni popsat váhu celé množiny S. Budeme také uvažovat uspořádání na množině vrcholů. Pomocí tohoto uspořádání budeme mj. definovat množinu hran. Každou podmnožinu S budeme potom uvažovat v seřazené formě a bude pro ni existovat odpovídající cesta v grafu.

Tedy $V = S \cup \{p\}$, kde $p \notin S$. Dále uvažujme libovolné lineární uspořádání $\preceq$ na V, takové že $\forall x \in V : p \preceq x$ (p je nejmenší prvek, vzhledem k tomuto uspořádání, $p = \min_{\preceq} V$). Množinu hran potom sestavíme následovně:

$$E = \{(u, v) \mid u, v \in S, u \preceq v, u \neq v\} \cup \{(p, v')\},$$

kde $v' = \min_{\preceq} S$. Pro vrchol p existuje pouze orientovanou hranu do nejmenšího prvku množiny S (vzhledem k $\preceq$). Co se týče hodnotové funkce, tak ta bude vypadat:

$$d : E \to \mathbb{Z}, \quad (u, v) \mapsto w(v).$$

Tato funkce je dobře definovaná, protože pro všechny vrcholy, mimo p, je definována funkce w, ale do vrcholu p nevede žádná hrana. Nakonec $X = W$.

Velikost výsledného grafu bude $|V| = n + 1$ a

$$|E| = 1 + n + (n-1) + \cdots + 1 = 1 + \frac{n(n-1)}{2} \in O(n^2).$$

Redukci R je tedy možné implementovat deterministickým TS v polynomiálním čase (Za uspořádání $\preceq$ lze například uvažovat pořadí, v jakém byly prvky z S zapsány na pásku). Zbývá dokázat, že $w \in SUBSET\_SUM \Leftrightarrow R(w) \in EXACT\_LENGTH$.

($\Rightarrow$) Pokud $(S, w, W) \in SUBSET\_SUM$, potom existuje $S' \subseteq S$ takové, že $w(S') = W$. Nechť $\preceq$ je zvolené lin. uspořádání a $q = \min_{\preceq} S'$. Dále nechť $q' \in V$ je bezprostřední předchůdce q (tedy $q' \preceq q$ a $\forall q'' \in V : q' \preceq q'' \Rightarrow q \preceq q''$) (takový určitě existuje vzhledem k přidanému vrcholu p, který není v S). Dále si prvky z S' seřadíme do rostoucí posloupnosti $(s_1, s_2, \ldots, s_m)$ vzhledem k $\preceq$. Potom ale v grafu G existuje cesta $\pi = q'e_1s_1e_2 \ldots e_ms_m$ a

$$len(\pi) = \sum_{i=1}^{m} d(e_i) = \sum_{i=1}^{m} w(s_i) = W = X.$$

($\Leftarrow$) Nechť $(G, X) \in EXACT\_LENGTH$. Potom existuje cesta $\pi = v_1e_1 \ldots e_{m-1}v_m$ taková, že $len(\pi) = X$. Vzhledem k tomu, jak je graf G definován, platí, že $S' = \{v_2, \ldots v_m\} \subseteq S$ a

$$\sum_{i=2}^{m} w(v_i) = \sum_{i=1}^{m} d(e_i) = len(\pi) = X = W.$$

Pozn. Implicitně předpokládám, že v případě nesprávně zformovaného vstupu redukce R zamítne.

3. Příklad Nejprve důkaz, že $L_t \in NP$. Existuje DTS, který přijímá L_t (zkontroluje, zda je na vstupu 0, pokud ano přijme, jinak odmítne) – to je možné provést v konstantním čase. Vzhledem k tomu, že DTS je speciální případ NTS, $L_t \in NP$.

Dále ukážeme, že pro každý jazyk $L' \in NP$ platí $L' \leq L_t$. Nechť L' je libovolný jazyk z NP. Vzhledem k tomu, že $P = NP$ (předpoklad), existuje DTS M pracující s polynomiální časovou složitostí $p(n)$ takový, že $L(M) = L'$. Zkonstruujeme redukci R, která pracuje následovně: R na svém vstupu simuluje DTS M. Pokud M přijme, potom R smaže obsah výstupní pásky a zapíše symbol 0. Pokud M odmítne, R na pásku zapíše symbol 1.

Vzhledem k tomu, že M je DTS, je možné i redukci R implementovat pomocí DTS M_R (simulaci lze provádět deterministicky). Co se týče časové složitosti M_R, tak pro simulaci M je potřeba $O(p(n))$ kroků, kde $p(n)$ je polynom. Smazání pásky zabere v nejhorším případě $O(p(n))$ kroků a zápis 0 nebo 1 se provede v konstantním čase. Celková složitost stroje M_R je tedy $O(p(n))$, to znamená polynomiální.

Navíc dostáváme: $w \in L' \Rightarrow R(w) = 0$ a tedy $R(w) \in L_t$. Také $w \notin L' \Rightarrow R(w) = 1$ a tedy $R(w) \notin L_t$. A tím pádem $w \in L' \Leftrightarrow R(w) \in L_t$. Jazyk L_t je tedy NP-úplný.

4. Příklad Předpokládejme, že $P = NP$. Dále pro každý jazyk $L \in NP$, který není prázdný nebo univerzální, existují řetězce w_1 a w_2 takové, že $w_1 \in L$ a $w_2 \notin L$. Potom je možné použít pro důkaz, že L je NP-úplný, důkaz předchozího tvrzení s tím, že řetězec 0 nahradíme za w_1 a řetězec 1 za w_2 (a samozřejmě jazyk L_t za L) – Redukce R pro libovolný $L' \in NP$ a řetězec w simuluje v polynomiálním čase DTS M: $L(M) = L'$ na vstupu w a pokud M přijme, redukce R smaže pásku a zapíše w_1. Pokud odmítne, zapíše w_2 (délka řetězců w_1, w_2 nezáleží na vstupu – jsou pevně zvolené, R lze tedy implementovat DTS s polyn. čas. složitostí). A také platí $w \in L' \Leftrightarrow R(w) \in L$ (viz předchozí důkaz).