Složitost

2. ledna 2013

Obsah

1 Úvod

K porovnávání algoritmů slouží složitost. Závisí na velikosti vstupu. Pokud bychom chtěli měřit velikost vstupu, rigorózně se používá počet bitů nutných k zakódování. Pokud by se používala jiná než dvojková soustava, tak to je až na konstantu stejné číslo – není zajímavé (až na jednoprvkové abecedy).

Časová složitost je funkce udávající počet kroků algoritmu v závislosti na velikosti vstupu, obdobně u paměťové. Nezajímá nás přesný tvar, jen asymptota. U polynomiálních to roste ještě „rozumě“, u exponenciálních a podobně je to už nepoužitelné.

Typy odhadů:

- O – horní odhad.
- ω – ostrý dolní odhad.
- o – ostrý horní odhad.
- Ω – dolní odhad.
- Θ – přesný odhad.

2 Hladové algoritmy

Algoritmus hledá globální minimum tak, že vždy lokálně vybere maximum. Mezi ně patří například minimální kostra grafu (Borůvkův). Nebo množina úkolů (jednotkové délky), u nich lhůty a penalizace.

2.1 Matroidy

Matroid je (X, S), kde $S \subseteq \mathcal{P}(X)$, splňující:

- $|X| < \omega, X \neq \emptyset$
- $A \in S \wedge B \subseteq A \Rightarrow B \in S$
- $\forall A, B \in S; |A| < |B| \Rightarrow \exists x \in B \backslash A; A \cup \{x\} \in S$

Matroidy například maticové (nezávislé jsou nezávislé sloupce), grafový (nezávislé množiny hran jsou takové, které tvoří les).

Lemma 1 *Les obsahující k hran sestává z $n - k$ stromů.*

Důkaz:
Dokáže se indukcí podle k (od 0).

☻

Nechť A, B jsou lesy a $|A| < |B|$. Tedy A má víc stromů, tedy můžu vzít něco z B, co vede mezi stromy z A.

Lemma 2 *Všechny maximální nezávislé (co do inkluze) jsou stejně veliké.*

Důkaz:
Spor s výměnnou vlastností – do menší bych mohl něco přidat.

☻

2.1.1 Matroidový problém

Máme kladné váhy pro prvky X. Chceme najít maximální nezávislou množinu. Lze udělat hladově.

Mnoho problémů se dá takto převádět na matroidy.

Lemma 3 *Pokud neexistuje jednoprvková nezávislá množina obsahující x, pak neexistuje žádná nezávislá obsahující x.*

Důkaz:
Triviální důsledek dědičnosti.

☻

Důsledek 1 *Prvky přeskočené před prvním výběrem nebudou v žádné optimální množině.*

Důsledek 2 *Nechť S je setříděno podle vah a nechť je první prvek, který je nezávislý. Potom existuje optimální řešení, které obsahuje x. Tedy první výběr nezablokuje cestu k optimu.*

Lemma 4 *Když už víme, že x tam bude, můžeme ze všeho x vyhodit (a vyhodit ty nezávislé, které neobsahují x) a najít optimum zbytku.*

Důkaz:
Třeba dokázat, že po kontrakci vznikne matroid a že to bude obsahovat přesně to, co má.

☻

Věta 1 *Hladový algoritmus dává pro matroidy správný výsledek.*

Důkaz:
Jen iteruji předchozí lemmata.

☻

3 2-souvislost

Graf je 2-souvislý, když nemá artikulaci. Hledám pomocí DFS – nějak si všímáme zpětné hrany.

4 PSpace-úplnost

Budeme používat polynomiální převoditelnost. Problém A je tedy ***PSpace-úplný***, pokud každý PSpace problém na něj lze převést a sám je také PSpace.

Polynomiální hierarchie je podmnožina PSpace.

Ukázkou takového problému jsou kvantifikované booleovské formule. To má v sobě proměnné, spojky a kvantifikátory. Díky výrokové logice lze požadovat, aby byly v prenexním tvaru – mají kvantifikátory vepředu (trik je, že neomezujeme, kolik je kvantifikátorů). V polynomiálním prostoru můžeme vyzkoušet všechna ohodnocení těch formulí (projdeme všechny možnosti, rozepsat formuli nemůžeme, to by bylo moc dlouhé).

Nyní chceme kvantifikovanou formuli pro každý turingův stroj s polynomiálně omezenou pamětí. Budeme tvrdit, že existuje cesta z počáteční do koncové konfigurace.

Uděláme si graf konfigurací. Bude tam $2^{O(S(n))}$ vrcholů. Počet hran bude něco podobného. Ověření, že mezi dvěma vrcholy existuje hrana je v pohodě, to jde popsat jednou formulí. Chceme ověřit, že existuje cesta z počáteční do přijímající konfigurace. Uděláme trik. Budeme konstruovat formuli, že se odněkud někam dá dostat pomocí max 2^k kroků. To je totéž, jako že existuje nějaký vrchol, do kterého se dá dostat za max 2^{k-1} a z něj také. To, bohužel, nefunguje, je to moc dlouhé, proto to nejde rychle vygenerovat.

Přepíšeme dvě poloviny tak, že použijeme nějaké provšechnítko a implikaci (když to je dosazení prvním způsobem nebo druhým způsobem, tak zkontroluj formuli). Tohle narůstá jen lineárně s počtem kroků do logaritmické hloubky, tedy budeme mít polynomiálně dlouhou formuli.

Podobně to vypadá při hledání vítězné strategie v nějaké hře s otevřenou informací.

Kdyby nebyl rozdíl mezi PSpace a Polynomiální hyerarchií, tak ta na některé úrovni kolabuje.