Týden 14

Přednáška-pondělí

Aproximační algoritmy

Vrátili jsme se podrobněji k elementárním základům zachyceným v sekci 10.3.

Pravděpodobnostní algoritmy

Přiblížili jsme si elementární základy zachycené v sekci 10.4.
Speciálně jsme se věnovali problému prvočíselnosti.
Uvědomili jsme si, že algoritmus

> Máš-li testovat prvočíselnost zadaného (např. několikasetmístného) k, projdi všechna $a, 1 < a < k$ a zjišťuj, zda $Divides(a, k)$ (tedy zda $(k \mod a) = 0$) ...

je exponenciální (ve velikosti zápisu k). (A to i při přímočarých vylepšeních, při nichž zkoumáme jen lichá $a \leq \sqrt{k}$ apod.)
Také jsme si všimli, že pravděpodobnostní algoritmus

> Vygeneruj náhodné a (řekněme liché $a, 1 < a \leq \sqrt{k}$); jestliže $Divides(a, k)$, return NE, jinak return ANO.

nám moc nepomůže. Vydá-li (nějaký) jeho běh NE, tak sice víme jistě, že k není prvočíslo, ale vydá-li ANO pro dané k třeba při miliónkrát opakovaném provedení, nemůžeme si vůbec být jisti, že k je prvočíslem. (Např. pro velké číslo $m = pq$, kde p, q jsou prvočísla, je náhodná trefa jednoho z dělitelů p, q téměř nemožná.)
Pak jsme naznačili, že (malá) Fermatova věta, je základem podstatně lepšího algoritmu (k čemuž se ještě vrátíme na cvičení).

Přednáška-čtvrtek

Čas přednášky bude věnován shrnutí látky především s ohledem na zkoušku. Lze to vidět jako určitou hromadnou konzultaci; předpokládá se přitom především aktivní přístup posluchačů, konkrétní dotazy apod.

Partie textu k prostudování

Sekce 10.3. (Aproximační algoritmy). Sekce 10.4. (Pravděpodobnostní algoritmy).

Cvičení

Prezentace referátů

Referát č. 25 (Savitchova věta)
Popište konstrukci v důkazu Savitchovy věty. (K nastudování můžete např. využít podklad k referátu č. 8 na `http://www.cs.vsb.cz/jancar/VYCSLOZ/vycsloz.htm`.)
Můžete se ovšem omezit na tento speciální případ:

> Je-li problém P rozhodován nedeterministickým Turingovým strojem s prostorovou složitostí n, pak je také rozhodován deterministickým Turingovým strojem s polynomiální prostorovou složitostí.

Máte tedy vysvětlit, jak lze k tzv. lineárně omezenému automatu (linear bounded automaton) M,

> tj. k nedeterministickému Turingovu stroji M, který při výpočtu na vstupním w, $|w| = n$, nenavštíví jiná políčka než ta, na nichž je zapsán vstup (a má tedy prostorovou složitost n)

navrhnout (deterministický) algoritmus A, který pro zadané w zjistí, zda M má přijímající výpočet pro w (tedy zda $w \in L(M)$). Algoritmu A přitom musí stačit polynomiálně omezená paměť.
(Připomenutí. Počet konfigurací délky n stroje M je omezen hodnotou c^n, kde konstantu c lze snadno spočítat z velikosti (stavové množiny a abecedy) stroje M. Délka nejkratšího přijímajícího výpočtu M nad w, $|w| = n$, (pokud takový existuje) je tedy také omezena oním c^n.)
(Bylo by dobré ukázat, že ta prostorová složitost A se dá omezit kvadraticky, je v $O(n^2)$, a naznačit, proč A lze přímočaře implementovat deterministickým Turingovým strojem s prostorovou složitostí $O(n^2)$.)

Referát č. 26 (Problém QBF; Quantified Boolean Formulas)
Uvažujme problém

> *Název*: QBF (*problém pravdivosti kvantifikovaných booleovských formulí*)
>
> *Vstup*: formule $(\exists x_1)(\forall x_2)(\exists x_3)(\forall x_4)\ldots(\exists x_{2n-1})(\forall x_{2n})\mathcal{F}(x_1, x_2, \ldots, x_{2n})$, kde $\mathcal{F}(x_1, x_2, \ldots, x_{2n})$ je booleovská formule v konjunktivní normální formě.
> *Otázka*: je daná formule pravdivá ?

Navrhněte algoritmus, který řeší problém QBF a má prostorovou složitost omezenou polynomem. (Tím ukážete, že QBF je v PSPACE.)
Návod. Řekneme, že formule $\mathcal{F}(x_1, x_2, \ldots, x_{2n})$ je OK pro posloupnost booleovských hodnot $b_1, b_2, \ldots, b_i$, kde $0 \leq i \leq 2n$, jestliže

buď $i = 2n$ a $\mathcal{F}(b_1, b_2, \ldots, b_{2n}) = true$,

nebo $i < 2n$, i je liché a $\mathcal{F}$ je OK jak pro $b_1, b_2, \ldots, b_i, true$, tak pro $b_1, b_2, \ldots, b_i, false$,

nebo $i < 2n$, i je sudé a $\mathcal{F}$ je OK pro alespoň jednu z posloupností $b_1, b_2, \ldots, b_i, true$ a $b_1, b_2, \ldots, b_i, false$.

Ověřte nejprve, že formule $(\exists x_1)(\forall x_2)(\exists x_3)(\forall x_4)\ldots(\exists x_{2n-1})(\forall x_{2n})\mathcal{F}(x_1, x_2, \ldots, x_{2n})$ je pravdivá právě tehdy, když $\mathcal{F}$ je OK pro prázdnou posloupnost.
Pak sestavte kýžený algoritmus (a prokažte, že jeho prostorová [tedy paměťová] složitost je polynomiální).

Referát č. 27 (Oblázková hra v PSPACE)
Uvažujme problém, jehož instancí je orientovaný graf s vybraným vrcholem v a dále k 'oblázků'. Můžeme v jakémkoli pořadí provádět následující elementární kroky:

- na vrchol x můžeme položit oblázek, pokud v daný okamžik leží oblázky na všech vrcholech, z nichž vede hrana do x,
- oblázek položený na vrchol můžeme odebrat (a znovu použít později).

Otázkou je, zda existuje posloupnost kroků, při níž položíme oblázek na zadaný vrchol v.
Prokažte, že problém je v PSPACE.
(Jednou z motivací problému je problém přidělování paměti při výpočtu; stačí daný počet registrů k provedení určeného výpočtu ?)

Příklady

Příklad 14.1

Následující tvrzení je známo jako „Malá Fermatova věta“.
Tvrzení. Jestliže p je prvočíslo, tak pro každé $a, 0 < a < p$, platí

$$a^{p-1} \equiv 1 \pmod{p}$$

.
(Když p není prvočíslo, tak to neplatí, jak byste se měli být schopni sami snadno přesvědčit [např. zbude-li čas na cvičení]).
Přesvěčdte se, že tvrzení platí pro $p = 11$. Přitom si uvědomte, jak je užitečné tzv. opakované umocňování. Můžete postupovat vyplněním následující tabulky; přitom využijte, že $x^{10} = x^8 \cdot x^2$, tedy $x^{10} \mod 11 = (x^8 \mod 11) \cdot (x^2 \mod 11) \mod 11$.

x	$x^2 \mod 11$	$x^4 \mod 11$	$x^8 \mod 11$	$x^{10} \mod 11$
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Pak vyplňte podobnou tabulku pro neprvočíslo 15.

x	$x^2 \mod 15$	$x^4 \mod 15$	$x^8 \mod 15$	$x^{14} \mod 15$
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				

Uvedená pozorování nabízejí zvážit jistý (polynomiální) pravděpodobnostní algoritmus k testování prvočíselnosti (velkých čísel). Jak vypadá tento algoritmus?

(Poznámka. Ten algoritmus „téměř“ funguje, „ošálí“ jej ale tzv. Carmichaelova čísla; naprosto korektní pravděpodobnostní algoritmus využívá o něco hlubší poznatky z teorie čísel.)

Příklad 14.2

Pokračujte případně v diskusi zkouškové písemky, především na základě konkrétních podnětů studentů.