

Týden 12

Přednáška

Vraceli jsme se ještě k dřívějším tématům.

Aproximační algoritmy

Přiblížili jsme si elementární základy zachycené 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 $(k \bmod a) = 0$) ...

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).

Partie textu k prostudování

Kapitola 9. Část 10.3. (aproximační algoritmy) a 10.4. (pravděpodobnostní algoritmy).

Cvičení

Příklad 13.1

Zformulujte (přirozený) hltavý aproximační algoritmus pro problém TSP (jdi do nejbližšího dosud nenavštíveného města ...). Proveďte odhad jeho složitosti. Ukažte instanci, v níž algoritmus vydá cestu, která je více než dvakrát delší než optimální cesta. (Vaše instance nemusí splňovat trojúhelníkovou nerovnost.)

Příklad 13.2

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.)

Přesvědčte 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 $x^{10} \bmod 11 = (x^8 \bmod 11) \cdot (x^2 \bmod 11) \bmod 11$.

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

Pak vyplňte podobnou tabulku pro neprvočíslo 15.

x	$x^2 \bmod 15$	$x^4 \bmod 15$	$x^8 \bmod 15$	$x^{14} \bmod 15$
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; na-prosto korektní pravděpodobnostní algoritmus využívá o něco hlubší poznatky z teorie čísel.)

Příklad 13.3

Diskutujte ukázkovou zkouškovou písemku. Speciálně se zaměřte na druhou část, v níž rovněž musíte získat předepsané minimum bodů. (Turingovy stroje, RAMy, polynomiální převeditelnost mezi problémy, konkrétní pozitivní a negativní instance rozhodovacích problémů, aplikace Riceovy věty, základní třídy složitosti a jejich úplné problémy, ...)