Otázky pro ADS2 – ZS 2016/17, zkoušející Luděk Kučera

Hlavním cílem přednášky není až tak naučit studenty konkrétní algoritmy, ale naučit je o přednesených algoritmech uvažovat, aby totéž pak byli schopni dělat o algoritmech, které sami vymyslí a naprogramují.

Proto pokud v dalším není uvedeno jinak, je u všech algoritmů třeba umět dokázat, proč dává správný výsledek, že se zastaví a za jak dlouho, tedy dokázat vlastnosti algoritmu. Samotná znalost výpočetního postupu stačí u složitých algoritmů nanejvýš na známku „dobře“ a u jednodušších algoritmů nestačí vůbec.

Hlavní otázky a na co se u nich zaměřit

1. **Vyhledávání v textu**:

Algoritmus Knuth-Morris-Pratt a jeho zobecnění Aho-Corasicková.

Důležité je především znát

a) hlavní invariant: stav výpočtu je dán nejdelším prefixem vzoru (či vzorů u AC), který je suffixem přečteného textu (u KMP je stav roven délce tohoto prefixu)

b) umět dokázat lineární časový odhad pro počet provedených operací

2. Toky v sítích:

Dinitzův algoritmus

Důležité je umět dokázat, že v každé další fázi výpočtu se délka nejkratší zlepšující cesty prodlouží o alespoň 1.

Také je třeba bez přemýšlení vědět, zda hrana sítě, jejíž rezerva poklesla na 0 se zase v dalším průběhu výpočtu může objevit s nenulovou rezervou (a popřípadě kolikrát nejvýše se to může stát)

Goldbergův algoritmus

Je třeba umět dokázat, že algoritmus nalezne optimální tok a odvodit jeho časovou výpočetní složitost. Klíčové je zde lemma o tom, že z každého vrcholu s kladným přebytkem vede zpět do zdroje cesta v síti rezerv (neboli reziduální síti) složená z hran s kladnou rezervou.

3. Diskrétní Fourierova transformace

Algoritmus FFT (Fast Fourier Transform)

Je nezbytné vědět, že základem algoritmu je rozdělit úlohu na *čtyři* podúlohy poloviční velikosti, z nichž dvě a dvě jsou si rovny (a umět tu rovnost dokázat)

A nezapomeňte na inverzní transformaci (včetně důkazu, že je opravdu inverzí k té přímé)

4. Sčítání binárních čísel

Na přednášce byl popsán obvod, který sečte dvě n-bitová čísla a má hloubku $\log_2 n$, viz též Algovize. Jiné podobné obvody se dají nalézt v literatuře. Je třeba znát jeden takový obvod (nejspíš ten z přednášky) a umět zdůvodnit, že opravdu sčítá.

5. Třídící sítě

Na přednášce byl ukázán obvod pro třídění čísel podle velikosti, nazývaný *Bitonic-sort* (bitonické třídění). V textech pro ADS2 mých kolegů můžete najít obdobný obvod, pracující na jiném principu, nazývaný *Merge-sort*. Jeden z nich byste měli znát. Kromě popisu rekurzivní konstrukce obvodu je třeba umět dokázat, že opravdu třídí. U bitonického třídění je k tomu třeba znát lemma o rozdělení bitonické posloupnosti na dvě části (horní a dolní půlka čísel).

6. Voroného diagram a Delanauy triangulace

Fortunův algoritmus

Pokud byste se naučili jiný algoritmus pro Voroného diagram, bude to v pořádku, ale nedoporučuji to.

Zde není požadována znalost důkazu správnosti ani časový odhad, ale jen popis algoritmu: měli byste znát 3D pohled na problém (kužely) a vědět, co jsou místní a kružnicové události a co se při nich děje a také mít základní představu o kalendáři událostí.

7. Rabin-Millerův pravděpodobnostní test prvočíselnosti a RSA šifra

Zde pochopitelně není požadován důkaz správnosti RM testu (kdyby se ho někdo naučil, dostane jedničku s pěti hvězdičkami a téma a vedoucího diplomové práce – ale raději to nezkoušejte). Stejně tak u RSA jen základní představa o tom, proč to funguje. Pochopitelně je ale vyžadována taková znalost algoritmů, abyste je v případě potřeby byli schopni naprogramovat.

8. NP-úplnost

Umět alespoň 3 redukce (převody) mezi rozhodovacími problémy (například redukce mezi problémem splnitelnosti booleovských formulí, problémem tříbarevnosti grafu a problémem existence nezávislé množiny dané velikosti v grafu – viz .ppt prezentace na mých webových stránkách)

Znát definici třídy NP a nedeterministického algoritmu (neboli úlohy, kde je možné kladné řešení – existuje-li – uhádnout a poté ověřit)

Být schopný(á) vysvětlit základní myšlenku Cook-Levinovy věty a její význam (věta o tom, že každý problém ve třídě NP lze převést na problém splnitelnosti booleovských formulí (nebo na jiný NP-úplný problém)

Vedlejší otázky (kladené např. pro rozhodnutí mezi dvěma známkami za hlavní otázku)

- Vyhledávání v textu algoritmem Rabin-Karp

- Umět ukázat, že doba výpočtu Ford-Fulkersonova algoritmu pro toky v sítích nelze shora odhadnout jen ze znalosti počtu vrcholů a hran sítě

- Konvexní obal konečné množiny v rovině (umět dokázat, že pracuje v lineárním čase)

- Vědět několik výsledků o aproximačních algoritmech od polynomiálních aproximačních schémat po neaproximovatelné úlohy (pokud $P \neq NP$)

- Dynamické programování