Linköpings Universitet
Institutionen för datavetenskap
Peter Jonsson

TDDD20/TDDA32 Konstruktion och analys av algoritmer

Tentamen fredagen den 23:e oktober 2009, kl 08.00–13.00.

Hjälpmedel: Kursboken "Introduction to Algorithms" (Cormen, Leiserson, Rivest & Stein). Ordlista.

Poäng: Totalt kan 40 poäng erhållas. För godkänt krävs ca 16 poäng.

Jourhavande lärare: Peter Jonsson, tel 28 24 15 eller 070 773 43 89.

Allmänt: Skriv läsligt. Onödigt komplicerade lösningar kan leda till poängavdrag. Bristfälligt motiverade lösningar leder ofelbart till poängavdrag.

Uppgifterna i tentamen är inte ordnade efter svårighetsgrad.

Lycka till!

Peter

Uppgift 1. (8p)
Problemet 2SAT definieras så här:

Instans: En propositionslogisk formel F på konjunktiv normalform där varje klausul innehåller högst två literaler.
Fråga: Är F satisfierbar?

Visa att 2SAT kan lösas i polynomisk tid.

Uppgift 2. (8p)
Vi säger att en sträng $s = s_1 s_2 \dots s_m$ är *dold* i en sträng $t = t_1 t_2 \dots t_n$ om det finns tal $1 \leq j_1 < \dots < j_m \leq n$ sådana att $s = t_{j_1} t_{j_2} \dots t_{j_m}$. Konstruera en algoritm som i $O(m \cdot n)$ tid räknar ut hur många olika sätt strängen s finns dold i strängen t. Exempelvis finns strängen `kost` dold på två sätt i strängen `konstruktion` medan `rim` endast finns dold på ett sätt i `algoritm`.

Uppgift 3. (8p)
Låt $G = (V, E)$ vara en oriktad och sammanhängande graf där varje båge $e \in E$ har en positiv vikt $w(e)$. Vi vet att G innehåller minst ett minsta uppspännande träd och ett sådant kan hittas i polynomisk tid (t.ex. med Kruskals eller Prims algoritm). Konstruera en polynomisk algoritm som kontrollerar om en given graf med positiva bågvikter innehåller minst två olika minsta uppspännande träd.

Uppgift 4. (8p)
Maximeringsproblemet MAX CUT definieras så här:

Indata: Oriktad graf $G = (V, E)$;
Lösning: Två mängder V_1, V_2 sådana att $V = V_1 \cup V_2$;
Mått: Antal bågar som har en ändpunkt i V_1 och en ändpunkt i V_2.

Konstruera en polynomisk algoritm som approximerar MAX CUT inom 2, d.v.s. hittar en lösning som har ett mått på minst 50% av den optimala lösningen. Algoritmen får vara randomiserad och då är kravet att den *förväntade* storleken på lösningen är 50% av optimum.

Uppgift 5. (8p)
Att lösa linjära system med olikheter (d.v.s. givet en rationell $m \times n$-matris A och en rationell m-vektor b hitta en rationell n-vektor x sådan att $Ax \geq b$) är ett problem som är lösbart i polynomisk tid. Visa att problemet blir NP-hårt[1] om man tillåter sig att använda funktionen $\lceil \cdot \rceil$ i olikheterna; som vanligt gäller att $\lceil x \rceil = x$ om x är ett heltal och $\lceil x \rceil$ är närmaste större heltal annars. Exempel på tillåtna olikheter är $\lceil x_i \rceil + x_j + \lceil 2x_k \rceil \geq 1$, $\lceil x_i - 5x_j \rceil \geq 0$ och $x_i + x_j + x_k \geq -1$.

[1] Försök inte visa att problemet är i NP!