Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

Datum för tentamen	2013-08-28
Sal	TER2
Tid	14.00-19.00
Kurskod	TDDD20
Provkod	TEN1
Kursnamn/benämning	Konstruktion och analys av algoritmer
Institution	*IDA*
Antal uppgifter som ingår i tentamen	5
Antal sidor på tentamen (inkl. försättsbladet)	
Jour/Kursansvarig	Peter Jonsson
Telefon under skrivtid	013-282415, 070-7734389
Besöker salen ca kl.	15.30, 17.30
Kursadministratör (namn + tfnnr + mailadress)	Madeleine Häger Dahlqvist, 013-28 23 60 madeleine.hager.dahlqvist@liu.se
Tillåtna hjälpmedel	Kursbok "Introduction to Algorithms" (Cormen et al.) Ordlista.
Övrigt (exempel när resultat kan ses på webben, betygsgränser, visning, övriga salar tentan går i m.m.)	
Vilken typ av papper ska användas, rutigt eller linjerat	Rutigt
Antal exemplar i påsen	

Linköpings Universitet
Institutionen för datavetenskap
Peter Jonsson

TDDD20 Konstruktion och analys av algoritmer

Tentamen onsdagen den 28:e augusti 2013, kl 14.00–19.00.

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)
Låt $A_1, \ldots, A_m$ vara en uppsättning $n \times n$-matriser och $b_1, \ldots, b_m$ vara en uppsättning n-vektorer. Antag för enkelhets skull att alla förekommande koeffecienter är heltal. Konstruera en polynomisk algoritm som avgör om det existerar en *unik* rationell n-vektor x sådan att $A_i x = b_i$ för alla $1 \leq i \leq m$.

Uppgift 2. (8p)
Maximeringsproblemet MAX DICUT definieras så här:

Indata: Riktad graf $G = (V, E)$ som inte innehåller någon självloop (d.v.s. saknar bågar av typen (v, v)).
Lösning: Två mängder V_1, V_2 sådana att $V = V_1 \cup V_2$ och $V_1 \cap V_2 = \emptyset$.
Mått: Antal bågar som har sin startpunkt i V_1 och sin slutpunkt i V_2.

Konstruera en polynomisk algoritm som approximerar MAX CUT inom 4, d.v.s. hittar en lösning som har ett mått på minst 25% av den optimala lösningen. Algoritmen får vara randomiserad och då är kravet att den *förväntade* storleken på lösningen är minst 25% av optimum.

Uppgift 3. (8p)
Vi säger att en sträng $s = s_1 s_2 \ldots s_m$ är *dold* i en sträng $t = t_1 t_2 \ldots t_n$ om det finns tal $1 \leq j_1 < \ldots < j_m \leq n$ sådana att $s = t_{j_1} t_{j_2} \ldots t_{j_m}$. 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 4. (8p)
En *stig* mellan två noder s, t i en oriktad graf är en väg från s till t som inte passerar någon nod mer än en gång. Betrakta följande problem:

INDATA. En oriktad graf $G = (V, E)$, två noder $s, t \in V$ och ett heltal K.
FRÅGA. Finns det en stig från s till t som innehåller exakt K bågar?

Visa att detta problem är NP-fullständigt.

Uppgift 5. (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.