Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

Datum för tentamen	2012-01-12
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)	4
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 282360 madeleine.hager.dahlqvist@liu.se
Tillåtna hjälpmedel	Kursboken "Introduction to Algorithms" av 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 torsdagen den 12:e augusti 2012, 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)
En oriktad graf $G = (V, E)$ är *2-färgningsbar* om och endast om det existerar en total funktion $f : V \to \{0,1\}$ med följande egenskap:

om $(v, w) \in E$ så är $f(v) \neq f(w)$.

Konstruera en polynomisk algoritm som givet en (inte nödvändigtvis sammanhängande) graf och avgör om den är 2-färgningsbar eller inte. Om grafen är 2-färgningsbar ska algoritmen dessutom räkna ut hur många olika 2-färgningar grafen tillåter, d.v.s. hur många olika funktioner f det finns som uppfyller kraven ovan.

Uppgift 2. (8p)
Antag att man har en följd av vektorer $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ där alla x_i, y_i är positiva heltal. Vi definierar summan av två vektorer komponentvis, d.v.s. $(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$. Uppgiften är att konstruera en algoritm för att, givet en vektor (a, b) (också heltalsvärden), avgöra om det finns ett urval av vektorerna vars summa blir (a, b). Analysera algoritmen med avseende på tidskomplexitet. Algoritmer som provar alla möjligheter eller på annat sätt tar mycket längre tid än nödvändigt ger 0 poäng.

Tips: Problemet kan lösas i tid $O(nL^2)$ tid där

$$L = \max(x_1 + \ldots + x_n, y_1 + \ldots + y_n).$$

Upplysning: Problemet är NP-fullständigt. Detta motsäger naturligtvis inte föregående tips eftersom det endast går åt $O(\log_2 L)$ bitar för att representera talet L^2.

Uppgift 3. (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 4. (8p)
Betrakta följande problem:

Indata: En mängd variabler V och en mängd uttryck U av typen $R(x, y, a, b)$ där $x, y \in V$ och $a, b \in \{0, 1, 2\}$.

Fråga: Finns det en total funktion $f : V \to \{0, 1, 2\}$ sådan att för varje $R(x, y, a, b)$ i U så uppfylls minst ett av villkoren $f(x) \neq a$ och $f(y) \neq b$?

Visa att problemet ovan 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.