



Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

Datum för tentamen	2012-08-24
Sal	TER1
Tid	14-19
Kurskod	TDDD20
Provkod	TEN1
Kursnamn/benämning	Konstruktion och analys av algoritmer
Institution	<i>IDA</i>
Antal uppgifter som ingår i tentamen	5
Antal sidor på tentamen (inkl. försättsbladet)	4
Jour/Kursansvarig	Peter Jonsson
Telefon under skrivtid	013282415, 0707734389
Besöker salen ca kl.	15.30, 17.30
Kursadministratör (namn + tfnr + mailadress)	Madeleine Häger Dahlqvist 013282360, 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 fredagen den 24: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)

Låt A_1, \dots, A_m vara en uppsättning $n \times n$ -matriser och b_1, \dots, b_m vara en uppsättning n -vektorer. Antag för enkelhets skull att alla förekommande koefficienter 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)

Betrakta följande grafproblem:

Indata: Sammanhängande graf G med n noder och m bågar.

Utdata: Alla par av noder x, y sådana att om man tar bort x och y ur G så blir den resulterande grafen ej sammanhängande.

Konstruera en algoritm som löser detta problem i $O(n \cdot m)$ tid. Observera att man kan kontrollera om en graf är sammanhängande i $O(m)$ tid. Alltså kan problemet trivialt lösas i $O(n^2 \cdot m)$ tid.

Uppgift 3. (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 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)

I en sekvens innehållande n positiva heltal så vill vi identifiera tal som förekommer mer än $\lceil n/3 \rceil$ gånger (det kan finnas maximalt två sådana tal i en given sekvens). Konstruera en algoritm som löser detta problem i $O(n)$ tid. Var noggrann med att bevisa algoritmens korrekthet! Tips: Notera att det inte finns någon övre gräns på talens storlek. Detta innebär, bland annat, att man inte kan basera sin algoritm på Radix-Sort eller någon liknande "snabb" sorteringsalgoritm.