Försättsblad till skriftlig tentamen vid Linköpings universitet

(fylls i av ansvarig)

Datum för tentamen	2014-01-10
Sal	TER1
Tid	08.00-13.00
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.	09.30, 11.30
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 fredagen den 10:e januari 2014, 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. Om inte annat anges skall presenterade algoritmer vara körbara på en processor av standardtyp (vilket utesluter exempelvis kvantdatorer och massiv parallellism). Slumpbitar får användas av algoritmer vid behov och de förutsätts kunna genereras i konstant tid.

Uppgifterna i tentamen är inte ordnade efter svårighetsgrad.

Lycka till!

Peter

Uppgift 1. (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 2. (8p)
En instans av *mängdproblemet* består av en ändlig mängd trippler (x_i, C, x_j) där x_i, x_j är variabler och $C \in \{\subseteq, \mathsf{disj}\}$. Låt $I = \{T_1, \ldots, T_k\}$ vara en godtycklig instans av detta problem över variablerna $X = \{x_1, \ldots, x_m\}$ och låt E vara mängden av alla icke-tomma, ändliga delmängder av de naturliga talen. Vi säger att I har en lösning om och endast om det finns en total funktion f från X till E sådan att

- $f(x_i) \subseteq f(x_j)$ för varje $(x_i, \subseteq, x_j)$ i I; och
- $f(x_i) \cap f(x_j) = \emptyset$ för varje $(x_i, \mathsf{disj}, x_j)$ i I.

Konstruera en polynomisk algoritm som undersöker om en godtycklig instans av mängdproblemet har en lösning.

Uppgift 3. (8p)
Låt $(w_1, \ldots, w_n)$ och $(c_1, \ldots, c_n)$ vara två sekvenser av noll-skilda och positiva heltal samt låt W vara ett positivt heltal. Konstruera en algoritm som hittar en mängd $I \subseteq \{1, \ldots, n\}$ med följande egenskaper

1. $\sum_{i \in I} w_i \leq W$ och
2. $\sum_{i \in I} c_i$ är så stor som möjligt.

Algoritmen får högst gå i $O(n \cdot W)$ tid. Detta problem är NP-fullständigt men det motsäger inte denna tidsgräns eftersom det bara krävs $\lceil \log_2 W \rceil$ bitar för att representera talet W.

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)
Det NP-fullständiga problemet NAE3SAT definieras enligt följande:

Indata. En CNF-formel där varje klausul innehåller exakt tre stycken literaler.
Fråga. Finns det en tilldelning av sanningsvärden till propositionerna sådan att minst en men högst två literaler i varje klausul blir sann?

Eftersom detta problem är beräkningsmässigt svårt (om vi utgår från att $P \neq NP$) så behövs en approximationsalgoritm för problemet—denna algoritm skall i polynomisk tid konstruera en tilldelning som gör att så många klausuler som möjligt får en korrekt tilldelning (d.v.s. att minst en men högst två literaler i klausulen blir sann). Konstruera en sådan algoritm som ger *minst* 70% av klausulerna en korrekt tilldelning. Algoritmen får vara randomiserad: Då förväntas den ge minst 70% av klausulerna en korrekt tilldelning.