Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2019-03-22
Sal (1)	G32(22)
Tid	14-19
Utb. kod	TDDD20
Modul	TEN1
Utb. kodnamn/benämning **Modulnamn/benämning**	Konstruktion och analys av algoritmer Skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	5
Jour/Kursansvarig Ange vem som besöker salen	Peter Jonsson
Telefon under skrivtiden	013-282415, 070-7734389
Besöker salen ca klockan	15.30, 17.30
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Veronica Kindeland Gunnarsson 013-285634 veronica.kindeland.gunnarsson@liu.se
Tillåtna hjälpmedel	Kursboken "Introduction to Algorithms" (Cormen et al.) Ordlista
Övrigt	
Antal exemplar i påsen	

Linköpings Universitet
Institutionen för datavetenskap
Peter Jonsson

TDDD20 Konstruktion och analys av algoritmer

Tentamen fredagen den 22:e mars 2019, 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 013 282415, 070 7734389.

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)
Man 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 2. (8p)
Indata i denna uppgift består av en mängd A innehållande positiva heltal samt ett positivt heltal s. Konstruera en algoritm som kontrollerar om det finns två tal $t, u \in A$ sådana att $s = t + u$. Algoritmen ska gå strikt snabbare än i $O(n^2)$ tid där $n = |A|$, d.v.s. det ska finnas ett $\epsilon > 0$ sådant att algoritmen går i högst $O(n^{2-\epsilon})$ tid.

Uppgift 3. (8p)
Låt F vara en godtycklig SAT-formel (dvs. en propositionslogisk formel på konjunktiv normalform). En variabel x i F är *monoton* om den endast förekommer negerad eller onegerad i formeln. Betrakta följande formel:

$$(x \vee \neg y \vee \neg z) \wedge (x \vee y \vee z) \wedge (y \vee \neg z) \wedge (\neg u).$$

Här är x och u monotona: x förekommer endast onegerad medan u förekommer endast negerad. Däremot är varken y eller z är monotona. Visa att följande problem kan lösas i polynomisk tid.

INDATA: En SAT-formel F med n variabler varav högst $\lfloor \log_2(n) \rfloor$ variabler *inte* är monotona.

FRÅGA: Är F satisfierbar?

Uppgift 4. (8p)
Betrakta system av villkor där varje villkor är av typen $2 \leq x_i \cdot x_j \cdot x_k \leq 4$ och x_i, x_j, x_k är distinkta variabler. Konstruera en algoritm som tilldelar varje variabel ett heltal och som garanterar att denna tilldelning uppfyller minst hälften av villkoren. Algoritmen får vara randomiserad och då är kravet att det förväntade antalet uppfyllda villkor är minst 1/2. I båda fallen ska algoritmen gå i polynomisk tid.

Uppgift 5. (8p)
Visa att följande problem är NP-fullständigt.

INDATA: Oriktad graf $G = (V, E)$.

FRÅGA: Innehåller G en oberoende mängd av storlek $|V|/2$ eller större?