Teoritenta i DD1352 Algoritmer, datastrukturer och komplexitet 2015-12-18

Inga hjälpmedel är tillåtna. Skriv svaren direkt på blanketten. Bonuspoäng från 2015 kan tillgodoräknas på denna tenta. För betyg E krävs 13 poäng för DD1352. Den som dessutom klarar D-uppgiften får D och den som *dessutom* klarar C-uppgiften får C. Den får godkänt på tentan kan den 8 januari 2016 redovisa extralabben för att få A eller B som tentabetyg.

Lämna in tentan senast 11.00. Ta med dina prylar från platsen och lämna salen, men återvänd klockan 11.15, för då tar rättningen vid. Varje tentand ska rätta en annan (anonym) tentands tenta. Därefter kontrollerar lärarna rättningen och för in resultaten i Rapp ikväll.

1. (6 p) Är följande påståenden sanna eller falska? Ringa in rätt svar! För varje deluppgift ger riktigt svar 1 poäng och ett *övertygande motiverat* riktigt svar 2 poäng.

   a) $n(n \log n + 37(\log n)^3) \in \Omega(n^2 \log n)$.

   sant **falskt**

   Motivering:

   b) Hos en Montecarloalgoritm beror både det beräknade värdet och exekveringstiden på slumpen.

   sant **falskt**

   Motivering:

   c) stopproblemet$\in$EXPSPACE

   sant **falskt**

   Motivering:

2. (3 p) A, B, C och D är beslutsproblem. Anta att B är NP-fullständigt och att det finns polynomiska Karpreduktioner mellan problemen så här (en reduktion av A till B tecknas här $A \to B$):

$$A \quad \to \quad B \quad \leftrightarrow \quad C \quad \to \quad D$$

   Vad vet man då om komplexiteten för A, C och D? Sätt ett kryss i tabellen nedan för det man säkert vet och en ring för det som är möjligt men som man inte vet säkert.

	ligger i NP	är NP-fullständigt	är NP-svårt
A			
C			
D			

3. (2 p)
   a) Vad är den engelska termen för *NP-svår*? ....

   b) Vad är den svenska termen för *undecidable*? ....
   ....

4. (3 p; 1 p på a och 2 p på b)
   a) Definiera begreppet *beslutsproblem.*

   b) Definiera begreppet *NP-svår* utan att använda begreppet *NP-fullständig.*

5. (Uppgift för betyg D, betygskriterium: *förklara principerna för hur man kan hantera problem med hög komplexitet*)
   Ge två (konceptuellt olika) förslag på algoritmkonstruktionsmetoder som vanligen används för att bygga heuristiker för NP-svåra optimeringsproblem.

   1. ....

   2. ....

6. (Uppgift för betyg C, betygskriterium: *konstruera enkla heuristiker*)
   Vi söker i denna uppgift en polynomisk heuristik som ger en så bra lösning till TSP (handelsresandeproblemet) som möjligt, ju kortare tur desto bättre. Algoritmen ska också ge ett mått på hur långt bort från den optimala lösningen som den hittade lösningen som mest är, uttryckt i procent.

   Till ditt förfogande finns följande färdiga polynomiska algoritmer:

$\langle \Pi, s\rangle = \text{Christofides}(n, D)$	Approximationsalgoritm med approximationskvot 3/2.
$\langle \Pi, s\rangle = \text{RandomInsertion}(n, D)$	Probabilistisk heuristik.
$\langle \Pi_2, s\rangle = \text{LocalSearch2Opt}(n, D, \Pi_1)$	Deterministisk heuristik som ger en förbättring av lösningen Π_1 med hjälp av lokal sökning.

   där Π beskriver en lösning som en permutation av städerna, s är lösningens (turens) längd, n är antalet städer och D är en symmetrisk $n \times n$-matris som beskriver avstånden mellan städerna. Anta att avstånden är positiva heltal och uppfyller triangelolikheten.

   Din algoritm ska ta n och D som indata och returnera trippeln $\langle \Pi, s, g\rangle$ som utdata, där g är måttet på hur långt algoritmens lösning som mest är från den optimala.

   $\text{TSP}(n, D) =$

   Kort förklaring till beräkningen av g: