Teoritenta i DD1352/DD2352 Algoritmer (datastrukturer) och komplexitet
2015-08-19 kl 9.00-11.00

Inga hjälpmedel är tillåtna. Skriv svaren direkt på blanketten.

Bonuspoäng från läsåret 2014/2015 kan tillgodoräknas på denna tenta. För betyg E krävs 13 poäng för DD1352 och 12 poäng för DD2352. Den som dessutom klarar D-uppgiften får D och den som *dessutom* klarar C-uppgiften får C.

Namn: .... *Personnr:* ....

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) $2^n \in \Omega(n^2)$.

   sant **falskt**

   Motivering:

   b) Den förväntade tidskomplexiteten för Random Quicksort är $O(n)$.

   sant **falskt**

   Motivering:

   c) NP$\subseteq$PSPACE

   sant **falskt**

   Motivering:

2. (4 p) A, B, C, D och E är beslutsproblem. Anta att B är NP-fullständigt och att det finns kända polynomiska Karpreduktioner mellan problemen så här (en reduktion av A till B tecknas här A$\rightarrow$B):

$$\begin{array}{ccccc} A & \leftarrow & B & \leftrightarrow & C \\ & & \uparrow & & \uparrow \\ & & D & \leftarrow & E \end{array}$$

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

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

3. (4 p; 2 p på varje deluppgift)
   a) Definiera begreppet *Karpreduktion*.

   b) Definiera begreppet *oavgörbart problem*.

4. (Uppgift för betyg D, betygskriterium: *förklara principerna för hur man kan hantera problem med hög komplexitet*)
   Ge två olika förslag på hur man kan angripa NP-svåra beslutsproblem som inte är förtäckta optimeringsproblem (som vanligt förutsatt att P$\neq$NP).

   Förslag 1:

   Förslag 2:

5. (Uppgift för betyg C, betygskriterium: *konstruera enkla heuristiker*)
   Optimeringsproblemet *Minimal delmängdssumma* tar som indata ett mål M och en lista med positiva heltal $t_1, t_2, \ldots, t_n$. I denna uppgift antar vi att $t_1 \leq t_2 \leq \ldots \leq t_n$.

   En tillåten lösning till problemet är en delmängd av talen i listan som har summa större än eller lika med M. Vi söker en lösning som har så liten summa som möjligt.

   Uppgiften är att konstruera en heuristik för Minimal delmängdssumma som först konstruerar en lösning och därefter använder lokalsökning för att förbättra lösningen.

   Beskriv algoritmen i text eller pseudokod.

   MinDelmängdssummeHeuristik(M, $(t_1, t_2, \ldots, t_n)$) =

   // Konstruktion:

   // Lokalsökning: