

## Teoritent i DD1352 Algoritmer, datastrukturer och komplexitet 2016-12-19

Inga hjälpmedel är tillåtna. Skriv svaren direkt på blanketten. Bonuspoäng från 2016 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 som får godkänt på tentan kan den 10 januari 2017 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. Klockan 11.15 börjar den obligatoriska kamraträttningen i sal F2. 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)  $3^n \in \omega(2^n)$ .

**sant**      **falskt**

Motivering:

- b) En Lasvegasalgoritm är en probabilistisk algoritm som går i polynomisk tid.

**sant**      **falskt**

Motivering:

- c) Mästarsatsen lämpar sig för att beräkna tidskomplexiteten för dynamiskprogrammeringsalgoritmer.

**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  $A \rightarrow B$ ):

$$\begin{array}{ccc} A & \leftarrow & B \\ \downarrow & & \uparrow \\ C & \leftarrow & D \end{array}$$

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 *totalsökning*? .....

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

4. (3 p; 1 p på a och 2 p på b)

a) Definiera begreppet *oavgörbart problem*.

b) Definiera komplexitetsklassen *NP*. (Ge bara en definition.)

5. (Uppgift för betyg D, betygsriterium: *förklara principerna för hur man kan hantera problem med hög komplexitet*)

Ge två konceptuellt olika förslag på hur man kan angripa NP-svåra beslutsproblem som inte är förtäckta optimeringsproblem (som vanligt förutsatt att  $P \neq NP$ ).

1. ....

2. ....

6. (Uppgift för betyg C, betygsriterium: *konstruera enkla heuristiker*)

I *Kappsäcksproblemet* är indata ett tal  $W$  och  $n$  prylar, där pryl  $i$  har vikten  $w_i$  och värdet  $v_i$ . Problemet är att bestämma vilka prylar som ska packas ner i kappsäcken så att summan av deras vikt är högst  $W$  och summan av deras värde är maximal.

Vi söker i denna uppgift en heuristik som försöker hitta en bra lösning till Kappsäcksproblemet, ju större sammanlagt värde desto bättre. Heuristiken ska använda sig av *lokal sökning*. Beskriv din algoritm med pseudokod på relativt hög nivå.