

**Teoritent i DD1352/DD2352 Algoritmer (datastrukturer) och komplexitet**  
**2016-08-17 kl 9.00-11.00**

Inga hjälpmedel är tillåtna. Skriv svaren direkt på blanketten. Examinator: Viggo Kann, 08-7906292.

Bonuspoäng från läsåret 2015/2016 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. Den som nästa vecka vill gå upp på munta för högre betyg (godkänt på skriftliga tentan krävs bl.a.) sätter ett kryss här:

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

1. (8 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)  $n^2 \in \omega(n \log n)$ .

**sant      falskt**

Motivering:

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

**sant      falskt**

Motivering:

c)  $\text{EXPTIME} \subset \text{P}$

**sant      falskt**

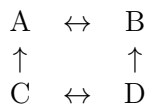
Motivering:

d) Om approximationsalgoritmen A approximerar ett minimeringsproblem inom faktorn 4 och optimala värdet för en viss probleminstans är 400 så vet man att A inte kan returnera en lösning med värde större än 1600.

**sant      falskt**

Motivering:

2. (3 p) A, B, C och D 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  $A \rightarrow B$ ):



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. a) (2 p) Definiera begreppet *Turingreduktion*.

b) (1 p) Definiera komplexitetsklassen *P*.

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 optimeringsproblem där den optimala lösningen inte kan beräknas tillräckligt effektivt.

*Förslag 1:*

*Förslag 2:*

5. (Uppgift för betyg C, betygskriterium: *konstruera enkla heuristiker eller totalsökningsalgoritmer*)

Formulera pseudokod för en exponentiell algoritm som löser problemet Hamiltonsk krets (Hamiltonian circuit), där indata är en oriktad graf med  $n$  hörn.