

**Teoritentan i Algoritmer (datastrukturer) och komplexitet  
för KTH DD1352/2D1352/DD2354/2D1354 och SU 2009-05-25  
klockan 9.00–11.00 med efterföljande kamraträttning**

Inga hjälpmedel är tillåtna. Skriv svaren direkt på blanketten. Skriv **inte** namn eller personnummer på tentan.

Bonuspoäng från läsåret 2008/2009 kan tillgodoräknas på denna tenta. 14 poäng krävs för betyg E (godkänt), 17 poäng för betyg D och 20 poäng för betyg C. Lämna in tentan när du är klar, men tidigast 9.45. Lämna sedan salen, men återvänd klockan 11.10, för då tar rättningen vid. Varje tentand ska rätta en annan (anonym) tentands tenta. Därefter kontrollerar Viggo rättningen och för in resultaten i res ikväll.

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) Med hjälp av *Graham scan* kan man multiplicera envariabelpolynom av höga grad-tal effektivt.

**sant      falskt**

Motivering:

b) En algoritm med tidskomplexiteten  $O((\log n)^{\log n})$  går i polynomisk tid.

**sant      falskt**

Motivering:

c) Generella handelsresandeproblemet ligger inte i *APX*, det vill säga, det kan inte approximeras inom någon konstant om inte  $P=NP$ .

**sant      falskt**

Motivering:

d) Inom komplexitetsteorin försöker man för varje problem hitta en så låg undre gräns och en så hög övre gräns som möjligt för tidskomplexiteten.

**sant      falskt**

Motivering:

2. (2 p) Stryk över dom felaktiga alternativen nedan så att en korrekt definition av begreppet *oavgörbart* uppstår.

Ett  $\left\{ \begin{array}{l} \text{beslutsproblem} \\ \text{optimeringsproblem} \\ \text{konstruktionsproblem} \end{array} \right.$  är *oavgörbart* om det  $\left\{ \begin{array}{l} \text{alltid} \\ \text{inte} \\ \text{oftast inte} \end{array} \right.$  finns någon algoritm som kan lösa problemet för  $\left\{ \begin{array}{l} \text{alla} \\ \text{något} \\ \text{de flesta} \end{array} \right.$  indata i  $\left\{ \begin{array}{l} \text{polynomisk} \\ \text{exponentiell} \\ \text{ändlig} \end{array} \right.$  tid.

3. (5 p)

a) När är det *inte* lämpligt att implementera en funktion som förberäknad?  
Stryk dom felaktiga alternativen!

När den har sidoeffekter.      När dess definitionsområde är litet jämfört med antalet anrop.  
När den ska skrivas i Java.      När dess definitionsområde är stort jämfört med antalet anrop.  
När den ska skrivas i C++.      När den anropas flera gånger med samma parametervärde.

b) På vilket sätt påminner en förberäknad funktion om dynamisk programmering?

4. (5 p) I detta problem ska du konstruera heuristiker för problemet Maximal klick (Max Clique). Du behöver inte analysera heuristikerna.

a) Beskriv med ord eller pseudokod en enkel konstruktionsheuristik för Maximal klick. Beskriv särskilt vilket kriterium du använder för att välja vilket hörn som ska behandlas.

b) Anta att en klick hittats med metoden i a). Beskriv med ord hur lösningen heuristiskt kan förbättras med *lokal sökning*.