

**Teoritenta i Algoritmer (datastrukturer) och komplexitet
för KTH 2D1352–1354 och SU 2007-12-15 klockan 10.00–13.00**

Inga hjälpmedel är tillåtna. 11 poäng krävs för betyg E, 14 poäng för betyg D och 17 poäng för betyg C.

1. (8 p) Är följande påståenden sanna eller falska? För varje deluppgift ger riktigt svar 1 poäng och ett övertygande motiverat riktigt svar 2 poäng.
 - a) Funktionen $n! \in O(2^n)$.
 - b) Varje ickedeterministisk polynomiell beslutsalgoritm är också en Monte Carlo-algoritm.
 - c) Följande problem är *oavgörbart* : Att givet en turingmaskin M , en sträng x och ett heltal k avgöra om beräkningen $M(x)$ stannar efter färre än k steg.
 - d) Följande gäller: $NP \subseteq PSPACE$.
2. (3 p) I den här uppgiften skall du beskriva vad en **approximationsalgoritm** är för något. Du skall förklara vad man kräver av en approximationsalgoritm och du skall beskriva vad algoritmens approximationskvot är. Ge också ett exempel på en approximationsalgoritm. Du kan t.ex. beskriva algoritmen för approximation av HÖRNTÄCKNING men andra exempel går också bra.
3. (4 p) När man vill visa att ett problem är NP-svårt använder man som bekant reduktioner. Vi tittar nu speciellt på de två problemen HAMILTONCYKEL och HANDELSRESANDEPROBLEMET (TSP).
 - a) Först vill jag att du visar hur man kan göra en reduktion från HAMILTONCYKEL till HANDELSRESANDEPROBLEMET.
 - b) Att göra en reduktion från HANDELSRESANDEPROBLEMET till HAMILTONCYKEL verkar vara betydligt svårare. Förklara ändå hur man vet att en sådan reduktion är möjlig att göra även om man inte vet riktigt hur den skulle se ut explicit.

Var god vänd!

4. (2 p)

Nedan ser du en algoritm med namnet borttaget. Försök identifiera vad det är för algoritm och vad den gör för något. Tala sedan om ifall det är ett exempel på en girig algoritm, en dynamisk programmering-algoritm eller en divide and conquer-algoritm.

```
???( $G = \langle V, E \rangle, W$ )
(1)   $n \leftarrow \dim(W)$ 
(2)  for  $i \leftarrow 1$  to  $n$ 
(3)    for  $j \leftarrow 1$  to  $n$ 
(4)       $d[i, j, 0] \leftarrow w_{ij}$ 
(5)  for  $k \leftarrow 1$  to  $n$ 
(6)    for  $i \leftarrow 1$  to  $n$ 
(7)      for  $j \leftarrow 1$  to  $n$ 
(8)         $d[i, j, k] \leftarrow \min(d[i, j, k-1], d[i, k, k-1] + d[k, j, k-1])$ 
(9)  return  $d$ 
```

5. (3 p)

Problemet K-FÄRGNING av graf är som bekant NP-fullständigt om K är godtyckligt. Men om $K=2$ är problemet faktiskt avgörbart i polynomiell tid. Visa hur!