Viggo Kann, KTH CSC

Teoritenta i Algoritmer (datastrukturer) och komplexitet för KTH DD1352/2D1352/DD2354/2D1354 och SU 2011-06-08 klockan 14.00–16.00

Inga hjälpmedel är tillåtna. Skriv svaren direkt på blanketten!

Endast bonuspoäng från läsåret 2010/2011 kan tillgodoräknas på denna tenta. 13 poäng krävs för betyg E (godkänt), 15 poäng för betyg D och 18 poäng för betyg C. Tentan rättas av Viggo (ingen kamraträttning) senare idag. Anmälan till eventuell munta nästa vecka görs med e-brev till viggo@nada.kth.se senast idag.

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) Det existerar inga problem som är så svåra att inte någon algoritm kan lösa dom i ändlig tid.

   sant **falskt**

   Motivering:

   b) En lista L består av n heltal. Alla tal i listan ligger mellan 1 och n. Det går att sortera dessa n tal i tid $O(n)$.

   sant **falskt**

   Motivering:

   c) Anta att beslutsproblemet A kan reduceras till beslutsproblemet B med en polynomisk Karpreduktion. Om B är NP-fullständigt så vet vi att A är NP-svårt.

   sant **falskt**

   Motivering:

   d) Anta att beslutsproblemet A kan reduceras till beslutsproblemet B med en polynomisk Karpreduktion. Om B är tillhör NP så vet vi att A tillhör NP.

   sant **falskt**

   Motivering:

2. (3 p) Ange för var och en av nedanstående algoritmer vilken algoritmkonstruktionsmetod den bygger på och namnet på det problem den löser.

   Mergesort .....

   Dijkstras algoritm .....

   Bellman-Fords algoritm .....

3. (4 p) Definiera *polynomisk tillväxt* och *exponentiell tillväxt*. Om ett lands befolkning växer med 3% per år, är då befolkningsökningen polynomisk eller exponentiell? Motivera!

4. (5 p) Arrayen $V[1..n]$ innehåller n stycken objekt, där varje objekt beskriver ett spelkort. Anta att vi har en hjälparray $H[1..n]$ och en funktion $Patiens(H[1..n])$ som simulerar läggningen av en patiens då spelkorten ligger i den ordning som anges av $H[1..n]$ och returnerar *sant* om patiensen går ut och *falskt* om patiensen inte går ut. Konstruera en totalsökningsalgoritm som testar alla permutationer av spelkorten i $V[1..n]$ och räknar ut antalet olika permutationer som gör att patiensen går ut (genom att anropa funktionen $Patiens$). Beskriv algoritmen med pseudokod!