Linköpings Universitet
Institutionen för datavetenskap
Peter Jonsson

TDDD20 Konstruktion och analys av algoritmer

Tentamen lördagen den 23:e oktober 2010, kl 08.00–13.00.

Hjälpmedel: Kursboken "Introduction to Algorithms" (Cormen, Leiserson, Rivest & Stein). Ordlista.

Poäng: Totalt kan 40 poäng erhållas. För godkänt krävs ca 16 poäng.

Jourhavande lärare: Peter Jonsson, tel 28 24 15 eller 070 773 43 89.

Allmänt: Skriv läsligt. Onödigt komplicerade lösningar kan leda till poängavdrag. Bristfälligt motiverade lösningar leder ofelbart till poängavdrag.

Uppgifterna i tentamen är inte ordnade efter svårighetsgrad.

Lycka till!

Peter

Uppgift 1. (8p)
Problemet 2SAT definieras så här:

Instans: En propositionslogisk formel F på konjunktiv normalform där varje klausul innehåller högst två literaler.
Fråga: Är F satisfierbar?

Visa att 2SAT kan lösas i polynomisk tid.

Uppgift 2. (8p)
Antag att man har en följd av vektorer $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ där alla x_i, y_i är positiva heltal. Vi definierar summan av två vektorer komponentvis, d.v.s. $(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2)$. Uppgiften är att konstruera en algoritm för att, givet en vektor (a, b) (också heltalsvärden), avgöra om det finns ett urval av vektorerna vars summa blir (a, b). Analysera algoritmen med avseende på tidskomplexitet. Algoritmer som provar alla möjligheter eller på annat sätt tar mycket längre tid än nödvändigt ger 0 poäng.

Tips: Problemet kan lösas i tid $O(nL^2)$ tid där

$$L = \max(x_1 + \ldots + x_n, y_1 + \ldots + y_n).$$

Upplysning: Problemet är NP-fullständigt. Detta motsäger naturligtvis inte föregående tips eftersom det endast går åt $O(\log_2 L)$ bitar för att representera talet L^2.

Uppgift 3. (8p)
En instans av *mängdproblemet* består av en ändlig mängd trippler (V_1, C, V_2) där V_1, V_2 är variabler och $C \in \{\subseteq, \mathsf{disj}\}$. Låt $I = \{T_1, \ldots, T_k\}$ vara en godtycklig instans av detta problem över variablerna $X = \{x_1, \ldots, x_m\}$ och låt E vara mängden av alla icke-tomma, ändliga delmängder av de naturliga talen. Vi säger att I har en lösning om och endast om det finns en total funktion f från X till E sådan att

- $f(x_i) \subseteq f(x_j)$ för varje $(x_i, \subseteq, x_j)$ i I; och
- $f(x_i) \cap f(x_j) = \emptyset$ för varje $(x_i, \mathsf{disj}, x_j)$ i I.

Konstruera en polynomisk algoritm som undersöker om en godtycklig instans av mängdproblemet har en lösning.

Linköpings Universitet
Institutionen för datavetenskap
Peter Jonsson

TDDD20 Konstruktion och analys av algoritmer

Tentamen lördagen den 23:e oktober 2010, kl 08.00–13.00.

Hjälpmedel: Kursboken "Introduction to Algorithms" (Cormen, Leiserson, Rivest & Stein). Ordlista.

Poäng: Totalt kan 40 poäng erhållas. För godkänt krävs ca 16 poäng.

Jourhavande lärare: Peter Jonsson, tel 28 24 15 eller 070 773 43 89.

Allmänt: Skriv läsligt. Onödigt komplicerade lösningar kan leda till poängavdrag. Bristfälligt motiverade lösningar leder ofelbart till poängavdrag.

Uppgifterna i tentamen är inte ordnade efter svårighetsgrad.

Lycka till!

Peter

Uppgift 4. (8p)
Maximeringsproblemet Max DiCut definieras så här:

Indata: Riktad graf $G = (V, E)$ som inte innehåller någon självloop (d.v.s. saknar bågar av typen (v, v));
Lösning: Två mängder V_1, V_2 sådana att $V = V_1 \cup V_2$;
Mått: Antal bågar som har sin startpunkt i V_1 och sin slutpunkt i V_2.

Konstruera en polynomisk algoritm som approximerar Max Cut inom 4, d.v.s. hittar en lösning som har ett mått på minst 25% av den optimala lösningen. Algoritmen får vara randomiserad och då är kravet att den *förväntade* storleken på lösningen är minst 25% av optimum.

Uppgift 5. (8p)
Att lösa linjära system med olikheter (d.v.s. givet en rationell $m \times n$-matris A och en rationell m-vektor b hitta en rationell n-vektor x sådan att $Ax \geq b$) är ett problem som är lösbart i polynomisk tid. Visa att problemet blir NP-hårt[1] om man tillåter sig att använda funktionen $\lceil \cdot \rceil$ i olikheterna; som vanligt gäller att $\lceil x \rceil = x$ om x är ett heltal och $\lceil x \rceil$ är närmaste större heltal annars. Exempel på tillåtna olikheter är $\lceil x_i \rceil + x_j + \lceil 2x_k \rceil \geq 1$, $\lceil x_i - 5x_j \rceil \geq 0$ och $x_i + x_j + x_k \geq -1$.

[1] Försök inte visa att problemet är i NP!