Linköpings Universitet
Institutionen för datavetenskap
Peter Jonsson

TDDD20 Konstruktion och analys av algoritmer

Tentamen fredagen den 1:e november 2013, 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. Om inte annat anges skall presenterade algoritmer vara deterministiska och körbara på en processor av standardtyp (vilket utesluter exempelvis kvantdatorer och massiv parallellism). Vidare får presenterade algoritmer använda högst polynomiskt mycket minne mätt i indatas storlek.

Uppgifterna i tentamen är inte ordnade efter svårighetsgrad.

Lycka till!

Peter

Uppgift 1. (8p)
Betrakta följande beräkningproblem:

Indata: En mängd intervall[1] $\mathcal{I} = \{I_1, \ldots, I_n\}$ på den rationella tallinjen där varje intervall I_j, $1 \leq j \leq n$, har en positiv och noll-skild vikt $w(I_j)$.
Utdata: En delmängd $\mathcal{I}'$ av $\mathcal{I}$ som uppfyller följande krav:

1. om $I, J \in \mathcal{I}'$ och $I \neq J$ så gäller att I och J är disjunkta, dvs $I \cap J = \emptyset$, samt
2. $\mathcal{I}'$ har största möjliga totalvikt, dvs. $\sum_{I \in \mathcal{I}'} w(I)$ är så stor som möjligt.

Konstruera en algoritm som löser detta problem i polynomisk tid.

Uppgift 2. (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 $V = V_1 \cup V_2$ och $V_1 \cap V_2 = \emptyset$.
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 3. (8p)
Låt $G = (V, E)$ vara en oriktad graf. Man säger att G är *k-färgbar* om det finns en funktion $f : V \rightarrow \{1, \ldots, k\}$ sådan att om $(v, w) \in E$ så gäller $f(v) \neq f(w)$ (det vill säga att sammanbundna noder ges olika färger). En *clique* i G är en delmängd $V' \subseteq V$ sådan att för alla $v, w \in V'$ (där $v \neq w$) så gäller det att $(v, w) \in E$ (det vill säga att varje nod i V' är sammanbunden med alla andra noder i V').

Låt X vara mängden av alla 5-färgbara grafer. Konstruera en algoritm som givet en godtycklig graf $G \in X$ hittar en clique av maximal storlek. Algoritmen måste gå i polynomisk tid.

---

[1] Ett intervall I representeras med ett par av tal (I_s, I_f) där $I_s < I_f$. Antag att både I_s och I_f är positiva heltal.

Uppgift 4. (8p)
En *ekvivalens* mellan två booleska literaler ℓ_1, ℓ_2 (vilket skrivs $\ell_1 \equiv \ell_2$) är sann om ℓ_1 och ℓ_2 tilldelas samma logiska värde. Exempelvis gäller att $x \equiv \neg y$ är sann om (1) x är sann och y är falsk, eller (2) x är falsk och y är sann. Däremot gäller inte $x \equiv \neg y$ om x är sann och y är sann.
Betrakta följande problem:

Indata: En uppsättning ekivalenser mellan par av literaler (där varje literal antingen är en negerad eller icke-negerad boolesk variabel) samt ett positivt heltal K.
Fråga: Går det att ge alla variabler booleska sanningsvärden så att minst K ekvivalenser är satisfierade?

Ett exempel på indata är

$$(x \equiv y, \neg y \equiv \neg z, z \equiv \neg w, w \equiv x, w \equiv x), \ K = 3$$

Man ser att ekvivalensen $w \equiv x$ förekommer två gånger i instansen; sådana flerdubblingar är tillåtna. Visa att problemet är NP-fullständigt.

Tips: Reducera från det NP-fullständiga problemet MAX-2-CNF (d.v.s. MAX-CNF där varje klausul innehåller högst två literaler). Notera att vi i kursen har kallat problemet MAX-2-CNF för MAX 2SAT.

Uppgift 5. (8p)
Binomialkoeffecienterna $C(n, k)$ definieras för positiva heltal $0 \leq k \leq n$ enligt följande rekursiva formel:

$$C(n, k) = C(n-1, k-1) + C(n-1, k) \text{ då } 1 \leq k \leq n-1$$

$$C(n, 0) = C(n, n) = 1 \text{ i övriga fall (dvs. när } k = 0 \text{ eller } k = n)$$

Konstruera en algoritm som beräknar $C(n, k)$ i $O(n \cdot k)$ tid.