S

Linköpings Universitet
Institutionen för datavetenskap
Peter Jonsson

TDDD20 Konstruktion och analys av algoritmer

Tentamen onsdagen den 26:e oktober 2015, kl 14.00–19.00.

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 körbara på en processor av standardtyp (vilket utesluter exempelvis kvantdatorer och massiv parallellism). Slumpbitar får användas av algoritmer vid behov och de förutsätts kunna genereras i konstant tid.

Uppgifterna i tentamen är inte ordnade efter svårighetsgrad.

Lycka till!

Peter

Uppgift 1. (8p)
Låt F vara en godtycklig satisfierbar 3SAT-formel. Låt V vara mängden variabler som F innehåller. Man säger att $f : V \to \{0, 1\}$ är en *komplementär modell* om f är en satisfierande tilldelning och den "motsatta" tilldelningen $f'(x) = 1 - f(x)$ också är satisfierande.

Antag att det finns en algoritm A som testar satisfierbarhet för 3SAT-formler i $O(c^n \cdot p(||F||))$ tid där n är antalet variabler i F, $||F||$ är antalet bitar som behövs för att representera F, p är ett polynom och $c > 1$ är en konstant. Konstruera en algoritm som gör följande:

- Om F saknar komplementär modell så svarar algoritmen "nej".
- Om F har en komplementär modell så returnerar algoritmen en komplementär modell f.

Algoritmen ska gå i $O(c^n \cdot q(||F||))$ tid där n är antalet variabler i den givna formeln F, $||F||$ är antalet bitar som behövs för att representera formeln F och q är ett fixt polynom. Algoritmen A får naturligtvis användas som en subrutin i lösningen.

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)
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 DICUT 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 4. (8p)
Vi har i kursen studerat *Horn-klausuler*, d.v.s. klausuler som innehåller högst en icke-negerad literal (t.ex. (p), $(\neg p)$, $(\neg p \vee \neg q \vee \neg r)$ och $(\neg p \vee \neg q \vee r \vee \neg s)$ men inte $(\neg p \vee q \vee r)$.) En *Horn-formel* är en propositionslogisk formel på CNF som endast innehåller Horn-klausuler. Vi vet att satisfierbarhet för Horn-formler kan avgöras i polynomisk tid. Visa att om man tillåter klausuler av typen $(p \vee q)$ att användas tillsammans med Horn-klausuler så blir satisfierbarhetsproblemet NP-fullständigt.

Uppgift 5. (8p)
Låt $G = (V, E)$ vara en oriktad graf. Man säger att G är *k-färgbar* om det finns en funktion $f : V \to \{1, \ldots, k\}$ 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.