Linköpings Universitet
Institutionen för datavetenskap
Peter Jonsson

TDDD20/TDDA32 Konstruktion och analys av algoritmer

Tentamen måndagen den 24:e augusti 2009, 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.

Uppgifterna i tentamen är inte ordnade efter svårighetsgrad.

Lycka till!

Peter

Uppgift 1. (8p)
Vi vet att 3SAT är ett NP-fullständigt problem. Visa att detta problem är NP-fullständigt även om man bara tillåter klausuler av följande två typer:

1. klausuler som endast består av tre icke-negerade variabler $(p \vee q \vee r)$
2. klausuler som endast består av två negerade variabler $(\neg p \vee \neg q)$.

Uppgift 2. (8p)
En *k-färgning* av en oriktad graf $G = (V, E)$ är en total funktion $f : V \to \{1, \ldots, k\}$ med följande egenskap: om $(v, w) \in E$ så är $f(v) \neq f(w)$. Att avgöra om grafer är 3-färgbara eller ej är ett välkänt NP-fullständigt problem.

Antag nu att vi har tillgång till en algoritm A som givet en graf G svarar "ja" om grafen kan 3-färgas och i annat fall svarar "nej". Algoritmen A går i $O(c^n)$ tid (där $c > 1$ är en konstant och n är antalet noder i den givna grafen). Konstruera en algoritm som givet en 3-färgbar graf G konstruerar en konkret 3-färgning av grafens noder. Algoritmen ska gå i $O(p(n) \cdot c^n)$ tid (där $p(n)$ är något polynom) eller bättre. Algoritmen får naturligtvis använda A som subrutin.

Uppgift 3. (8p)
Vi säger att en sträng $s = s_1 s_2 \ldots s_m$ är *dold* i en sträng $t = t_1 t_2 \ldots t_n$ om det finns tal $1 \leq j_1 < \ldots < j_m \leq n$ sådana att $s = t_{j_1} t_{j_2} \ldots t_{j_m}$. Konstruera en algoritm som i $O(m \cdot n)$ tid räknar ut hur många olika sätt strängen s finns dold i strängen t. Exempelvis finns strängen `kost` dold på två sätt i strängen `konstruktion` medan `rim` endast finns dold på ett sätt i `algoritm`.

Uppgift 4. (8p)
Låt $G = (V, E)$ vara en oriktad och sammanhängande graf där varje båge $e \in E$ har en positiv vikt $w(e)$. Vi vet att G innehåller minst ett minsta uppspännande träd och ett sådant kan hittas i polynomisk tid (t.ex. med Kruskals eller Prims algoritm). Konstruera en polynomisk algoritm som kontrollerar om en given graf med positiva bågvikter innehåller minst två olika minsta uppspännande träd.

Uppgift 5. (8p)
Pariteten för ett positivt heltal i är 1 om binärrepresentationen av i innehåller ett udda antal ettor och 0 om den innehåller ett jämnt antal ettor. Detta ger att t.ex. 1 och 8 har paritet 1 medan 3 och 5 har paritet 0. Givet ett heltal N så är en *paritetstabell* en tabell $par[0...N-1]$ sådan att $par[i]$ = pariteten för i. Konstruera en algoritm som skapar sådana tabeller i $O(N)$ tid. Man får anta att grundläggande matematiska operationer som addition, multiplikation och division går i $O(1)$ tid. Observera att problemet går lätt att lösa i $O(N \log N)$ tid; sådana algoritmer kommer att ge 0p.