NTNU
Norges teknisk-naturvitenskapelige universitet

Fakultet for informasjonsteknologi, matematikk og elektroteknikk

Institutt for datateknikk og informasjonsvitenskap

BOKMÅL

LØSNINGSFORSLAG, AVSLUTTENDE EKSAMEN I

TDT4120/IT1105

ALGORITMER OG DATASTRUKTURER

Fredag 8. desember 2006
Kl. 09.00 – 13.00

Faglig kontakt under eksamen:

Magnus Lie Hetland, tlf. 918 51 949

Hjelpemidler:

Alle trykte og håndskrevne hjelpemidler tillatt. Bestemt, enkel kalkulator tillatt.

Sensurdato:

8. januar. Resultater gjøres kjent på http://studweb.ntnu.no og sensurtelefon 815 48 014.

Viktig:

Les hele oppgavesettet før du begynner, disponér tiden og forbered evt. spørsmål til faglærer kommer til eksamenslokalet. Les oppgaveformuleringene grundig. Det er angitt i prosent hvor mye hver deloppgave teller ved sensur. Gjør antagelser der det er nødvendig. Skriv kort og konsist. Skriv fortrinnsvis i rutene på svarskjemaet (dvs. ikke legg ved egne ark med mindre det er tvingende nødvendig). Bruk gjerne blyant (og viskelær). Lange forklaringer som ikke direkte besvarer oppgaven tillegges liten eller ingen vekt.

Svarskjema

1a (6%): Fordi tall kan «hope seg opp» i én bøtte. (Bøttene sorteres med INSERTION-SORT.)
1b (6%): Lengden til den lengste felles subsekvensen til prefiksene X_i og Y_j.
1c (6%): Kjøretiden med heap blir da $\Theta(V^2 \lg V)$, men vi kan få $\Theta(V^2)$ med en enkel tabell.
1d (6%): Uttrykke den som en differanse $x' - x''$, der $x', x'' \geq 0$.
1e (6%): Her kan man argumentere på flere måter. Polynomisk reduksjon fra HAM-CYCLE til TSP er ikke nødvendigvis godt nok (det gjelder jo mellom alle NP-komplette problemer). Man må argumentere for at instansene til HAM-CYCLE kan ses på (mer eller mindre formelt) som instanser av TSP. Et enkelt argument er: «TSP finner en HAM-CYCLE med begrenset kostnad» eller noe tilsvarende. Om studentene bruker optimaliseringsvarianten er det akseptabelt her – dvs. «TSP finner en billigst mulig HAM-CYCLE». Det er her implisitt at kantene har vekter i TSP, og at ikke-eksisterende kanter får uendelig høy vekt. Eventuelt kan man si noe slikt som: «HAM-CYCLE er TSP med alle kantvekter satt til 1 og $k = \|V\|$». Evt. kan man i stedet for 1 si c og i stedet for $\|V\|$ si $c\|V\|$. Svar som viser hovedideen her gir full score (selv om de ikke nødvendigvis har med en diskusjon rundt k eller bruken av uendelige kantvekter, f.eks.)
2a (10%): $D^{(0)} \dots D^{(4)}$: $\begin{pmatrix} 0 & 1 & 8 & \infty \\ \infty & 0 & \infty & 2 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 8 & \infty \\ \infty & 0 & \infty & 2 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 8 & 3 \\ \infty & 0 & \infty & 2 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 8 & 3 \\ \infty & 0 & \infty & 2 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & 3 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 6 & 3 \\ \infty & 0 & 5 & 2 \\ \infty & \infty & 0 & \infty \\ \infty & \infty & 3 & 0 \end{pmatrix}$ Her får man 4 poeng for å ha fylt ut første matrise riktig, 2 poeng til for nr. 2 og 3 og 1 poeng per stk. for de siste to. (Merk at det vil ta *mye* kortere tid å fylle ut disse matrisene hvis man har skjønt algoritmen ordentlig; da er det relativt greit å se hva som faktisk kan endre seg ved å se på strukturen til grafen.)
3a (5%): Nei, fordi *best-case* og *worst-case* kan være forskjellige. Merk her at det med «tett, øvre grense» her menes at man ikke kan finnes noen asymptotisk lavere øvre grense. Noen kan ha misforstått dette, og trodd at det var snakk om en «asymptotisk tett grense» (dvs. ikke bare en øvre grense) – som vil gi svaret «ja». For at det skal gis poeng for et slikt svar må det være begrunnet (jfr. læreboka s. 42) og det gis uansett bare 4 av 5 poeng.
3b (5%): Ja. Hvis worst-case var lavere, ville ikke grensen ha vært tett. Se merknad til 3a. Svaret vil uansett være «ja» men begrunnelsen vil kunne være annerledes. Det vil da være mulig å gi full score på denne deloppgaven (det vil være snakk om en mindre følgefeil).
3c (5%): $T(n) = \Theta(n^2 \lg n)$.

Bruker f.eks. Masterteoremet. $a = 16$, $b = 4$, $f(n) = n^2$, $n^{\log_b a} = n^{\log_4 16} = n^2$. Tilfelle 2 gjelder.
4a (5%): Den teller antall forekomster av tallene $1 \ldots n$ i tabellen A, og legger svaret i B.
4b (3%): $\Theta(n)$
4c (7%): Den returnerer et intervall av lengde < 5 i A som inneholder a, eller som ligger like ved. **Merk**: Algoritmen var egentlig ment å returnere et intervall som inneholdt a, men på grunn av en feil gjør den altså ikke det. Forklaringer som får med seg noe av essensen i atferden til algoritmen vil uansett få god uttelling, og for studenter der det vil gi en bedre totalsum, vil oppgaven bli holdt utenfor sensur.
4d (5%): $\Theta(n)$. Hvert rekursive kall utfører konstant arbeid og reduserer problemstørrelsen med en konstant (dvs. 5). Kan evt. settes opp som rekurrensen $T(n) = T(n–5) + \Theta(1)$, som gir $T(n) = \Theta(n)$ vha. f.eks. iterasjonsmetoden.
5a (5%): Uenig. Problemstørrelsen (antall bits) er $m = \Theta(\lg n)$. Altså er kjøretiden $\Theta(m)$, dvs. lineær.
5b (5%): Finne en reduksjon fra VERTEX-COVER til *Slurm Invaders*, dvs. vise at en polynomisk løsning på *Slurm Invaders* gir oss en polynomisk løsning på VERTEX-COVER.

6a (2%): F.eks. [2, 3, 2].	6b (3%): F.eks. [2, 1, 1, 2].

6c (10%): Forklaring (krever ikke slik utdyping i student-svar): Bruker dynamisk programmering. «Beste blant de k første» ($W[i]$) inneholder enten node k eller ikke. Hvis den gjør det legges vekt w_k til «Beste blant de k–2 første» (node k–1 kan ikke være med). Hvis ikke brukes «Beste blant de k–1 første» direkte.

Pseudokode (mindre detaljerte beskrivelser aksepteres også, hvis det kommer klart frem hva ideen er):

```
HEAVIEST-INDEPENDENT-SUBSET(G=(V, E), w):
W[1] ← w_1
W[2] ← max(W[i–1], w_2)
for i ← 3 … |V|
    W[i] ← max(W[i–1], w_i + W[i–2])
return W[|V|]
```

Her vil det også være nok å ta vare på de 2 forrige verdiene i hver iterasjon.

Kjøretid: $\Theta(n)$