Avsluttende eksamen i
TDT4120 Algoritmer og datastrukturer (LF)

Eksamensdato	2. desember 2008
Eksamenstid	0900–1300
Sensurdato	23. desember
Språk/målform	Bokmål
Kontakt under eksamen	Magnus Lie Hetland (tlf. 91851949)
Tillatte hjelpemidler	Ingen trykte/håndskrevne; bestemt, enkel kalkulator

Vennligst les hele oppgavesettet før du begynner, disponer tiden og forbered evt. spørsmål til faglærer kommer til eksamenslokalet. Gjør antagelser der det er nødvendig. Skriv kort og konsist. Lange forklaringer og utledninger som ikke direkte besvarer oppgaven tillegges liten eller ingen vekt.

Algoritmer kan beskrives med tekst, pseudokode eller programkode, etter eget ønske, så lenge det klart fremgår hvordan den beskrevne algoritmen fungerer. Korte, abstrakte forklaringer kan være vel så gode som utførlig pseudokode, så lenge de er presise nok. **Merk**: Noen mindre rettinger er gjort med blå skrift.

Oppgave 1 (47%)

Anta at en probleminstans med størrelse n skal løses algoritmisk.

a) Skriv eksempler på følgende typer kjøretider som funksjon av n, uttrykt med Θ-notasjon.

Logaritmisk (2%)	$\Theta(\lg n)$
Lineær (2%)	$\Theta(n)$
Kvadratisk (2%)	$\Theta(n^2)$
Polynomisk (2%)	$\Theta(n)$ … eller andre polynomiske uttrykk (inkl. n^k eller $\lg n$)
Eksponentiell (2%)	$\Theta(2^n)$ … eller med andre grunntall (evt. f.eks. k)

Merk: Som eksponentiell aksepteres alt som ikke er polynomisk (altså ikke kun eksponentialfunksjoner). Det inkluderer f.eks. $\Theta(n!)$ eller $\Theta(n2^n)$.
Uttrykkene bør være tilstrekkelig forenklede (dvs. ikke $\Theta(2n^2 + n)$ e.l.).

b) Hvorfor kan man ikke bruke Θ-notasjon for å beskrive den generelle kjøretiden til QUICKSORT når det er mulig å bruke denne notasjonen for å beskrive både *best-case*-, *average-case*- og *worst-case*-kjøretidene hver for seg? Svar så kort som mulig.

Svar (8%):

Best-case og worst-case er asymptotisk forskjellige, og begge må omfattes.

Betrakt følgende algoritme:

```
for i = 1 … n
    for j = i … n/100
        print "Hello, World!"

for i = 1 … n
    for j = 1 … lg n
        print "Goodbye, World!"
```

Merk: Du kan anta at en løkke **for** $j = a \ldots b$ ikke utføres hvis $a > b$.

c) Hva blir kjøretiden til algoritmen, som funksjon av n, uttrykt med Θ-notasjon? Begrunn svaret svært kort.

Svar (10%):

$\Theta(n) \cdot \Theta(\text{sum}(i \text{ for } i = 1 \ldots n/100)) + \Theta(n) \cdot \Theta(\lg n) = \Theta(n^2)$

Merk: Uløste rekkesummer gir ikke full uttelling.

d) Hva er løsningen på følgende rekurrens? Oppgi svaret i Θ-notasjon.

$T(1) = 1$
$T(n) = T(n/2) + n$

Svar (8%): $\Theta(n)$

Betrakt følgende algoritme:

MYALGORITHM(n)

```
for i = 1 … n
    print "When will it ever end?"

if n = 1
    return TRUE

for i = 1 … 4
    MYALGORITHM(n/2)
```

e) Hva blir kjøretiden til algoritmen, som funksjon av n, uttrykt med Θ-notasjon? Begrunn svaret svært kort.

Svar (5%): $T(n) = 4T(n/2) + n$

Løses med masterteoremet (tilfelle 1): $\Theta(n^2)$

Du står overfor de tre problemene A, B og C. Alle tre befinner seg i mengden NP. Du vet at A er i mengden P og at B er i mengden NPC. Anta at du skal bruke polynomiske reduksjoner mellom disse problemene til å vise ulike egenskaper.

Merk: Enkelte av egenskapene kan selvfølgelig vises på andre måter. Du kan se bort fra det i denne oppgaven.

f) Fullfør følgende utsagn.

For å bevise at C er i P må C reduseres til A i polynomisk tid. (2%)

For å bevise at C er i NPC må B reduseres til C i polynomisk tid. (2%)

Hvis B kan reduseres til A i polynomisk tid, så er P = NP. (2%)

Oppgave 2 (26%)

a) Anta at du har en binær heap lagret i en tabell, som beskrevet i pensum. Anta at rotnoden ligger på indeks 1. Hvor (det vil si, i hvilken posisjon i tabellen) ligger foreldrenoden til elementet med indeks i?

Svar (6%): $i/2$, rundet ned

b) Hvor mange interne noder har et binærtre med n løvnoder, hvis alle interne noder har to barn?

Svar (7%): $n-1$

Merk: Her vil $n-2$ gi noe uttelling (dvs. interne noder unntatt rota).

c) Hva er forskjellen mellom en maksimal (*maximum*) matching og en perfekt matching?
Merk: Det er her snakk om bipartitt matching.

Svar (5%): Alle noder er med i en perfekt matching

Merk: Dette spørsmålet går noe utenfor det pensum som eksplisitt har blitt satt opp, og blir derfor tatt ut av sensuren der det er til fordel for studenten.

d) Hva er en Hamilton-sykel?

Svar (8%):

En sykel som går gjennom hver node én gang.

Oppgave 3 (17%)

a) Beskriv kort, med egne ord, hvordan RADIX SORT fungerer.

Svar (9%):

Den bruker (stabil) tellesortering på hvert siffer fra minst signifikante og oppover.

I FLOYD-WARSHALL brukes uttrykket $d^{(k)}_{ij}$ til å beskrive løsningen på et delproblem.

b) Hva er den rekursive formelen for $d^{(k)}_{ij}$?

Svar (8%):

$d^{(0)}_{ij} = w_{ij}$

$d^{(k)}_{ij} = \min(d^{(k-1)}_{ij}, d^{(k-1)}_{ik} + d^{(k-1)}_{kj}), k > 0$

Oppgave 4 (10%)

Anta at du har en rettet graf med positive heltallskantvekter. Hvis du skal finne den korteste veien fra u til v så kan det eksistere flere svar, det vil si flere stier som har samme (minimale) lengde.

a) Hvordan vil du effektivt finne den av de korteste stiene fra u til v som består av færrest kanter?

Svar (5%):

DIJKSTRA med nye kantvekter, f.eks. $w'(i, j) = E \cdot w(i, j) + 1$.

(**Forklaring**: Hvis en sti opprinnelig var kortere, vil den fortsatt være det, men om to stier var like lange, vil nå den med færrest kanter bli kortest. Hvis vi ikke hadde heltallsvekter kunne man ha hatt sti-lengder av typen (d, k), der d er summen av kantvekter og k er antallet kanter. Hvis vi behandler dette som to «siffer», der d er det mest signifikante, blir svaret riktig.)

b) Hvordan vil du effektivt finne ut hvor mange korteste stier (det vil si hvor mange stier med minimal lengde) det finnes fra u til v?

Svar (5%): DIJKSTRA med utvidet RELAX. Antall stier til x er $p[x]$. $p[u] = 1$. $p[x] = 0, x \neq u$.

RELAX(x, y):

$d' \leftarrow d[x] + w(x, y)$

if $d' < d[y]$

    $d[y] \leftarrow d'$

    $p[y] \leftarrow p[x]$

else if $d' = d[y]$

    $p[y] \leftarrow p[y] + p[x]$