Studentnummer _____

Eksamen i
IT1105/TDT4120 Algoritmer og datastrukturer
13. desember 2004 0900-1300

Faglig kontakt: Arne Halaas (*tlf* 416 61 982),
Magnus Lie Hetland (*tlf* 918 51 949)
Tillatte hjelpemidler: Bestemt enkel kalkulator. Ingen trykte eller håndskrevne hjelpemidler.
Merk: Svarene skal skrives på angitt sted på oppgavearket.

Oppgave 1 (20%)

For hver algoritme nedenfor skal du *kort* beskrive (*i*) *hvilket problem* algoritmen løser, (*ii*) *hvilke krav* som eventuelt stilles til probleminstansene og (*iii*) grovt *hvordan algoritmen virker*. Hvert delspørsmål besvares med *maksimalt 50 ord*, gjerne i stikkordsform.

4% *a*) QuickSort

Svar:

4% *b*) Dijkstras algoritme

Svar:

4% *c*) Binærsøk

Svar:

4% *d*) Prims algoritme

Svar:

4% *e*) Warshalls algoritme

Svar:

Oppgave 2 (20%)

I de følgende deloppgavene, kryss av for "ja" eller "nei" og gi en kort begrunnelse for svaret.

Merk: Svar med gal eller manglende begrunnelse gir ingen poeng.

4% *a*) En dårlig algoritme A som sjekker om et heltall n er et primtall har kjøretid $\Theta(n)$. Har denne algoritmen eksponensiell kjøretid?
Hint: Hvordan defineres problemstørrelse?

Svar: [ ] Ja [ ] Nei

Begrunnelse:

4% *b*) Har balanserte, rettede trær færre mulige topologiske ordninger enn andre sammenhengende, rettede asykliske grafer med like mange noder?

Svar: [ ] Ja [ ] Nei

Begrunnelse:

4% *c*) Steinar Hopp vil komme seg over en bred elv. Det er naturligvis begrenset hvor langt han kan hoppe, så han planlegger å hoppe fra stein til stein. Han vil finne den sekvensen av steiner som gir færrest hopp, og mener at dybde-først-søk egner seg bedre enn bredde-først-søk til dette. Er du enig? Angi i begrunnelsen hvordan algoritmene kan brukes, og hvorfor du foretrekker den ene fremfor den andre.

Svar: [ ] Ja [ ] Nei

Begrunnelse:

4% *d*) I asymptotisk notasjon beskriver O og Ω ulike ting. Er det riktig å si at O er et mål på *worst-case*-kjøretiden til en algoritme og at Ω er et mål på *best-case*-kjøretiden?

Svar: [ ] Ja [ ] Nei

Begrunnelse:

4% *e*) Vi kan si at et problem *A* er vanskeligere enn et problem *B* hvis alle instanser av problem *B* også kan ses som instanser av problem *A*, men ikke omvendt. Er topologisk sortering et vanskeligere problem enn vanlig sortering?

Svar: [ ] Ja [ ] Nei

Begrunnelse:

Oppgave 3 (20%)

4% *a*) Her følger en liste med betegnelser på kjøretidsklasser – ordne dem i stigende rekkefølge: *konstant*, *eksponensiell*, *lineær*, *kubisk*, *kvadratisk*, *faktoriell*, *logaritmisk*, *n-log-n*. Ingen begrunnelse er nødvendig.

Svar:

4% *b*) Hva gjør følgende programsnutt og hva blir kjøretiden? Bruk Θ-notasjon og begrunn svaret. Skriv kort.

Algorithm *Azathoth*($A[1\ldots n]$)

```
1. i ← 1
2. j ← n
3. while i < j
4.     if A[i] > A[j]
5.         i ← i + 1
6.     else
7.         j ← j – 1
8. return A[i]
```

Svar:

Begrunnelse:

6% *c*) Hva gjør følgende programsnutt og hva blir kjøretiden? Funksjonen *floor*() runder ned til nærmeste heltall. Bruk Θ-notasjon og begrunn svaret. Skriv kort.

Algorithm *Astaroth*(i, j, $A[1…n]$, $B[1…n]$)

```
1. if i = j
2.     if A[j] > B[j]
3.         return A[j] – B[j]
4.     else
5.         return B[j] – A[j]
6. k ← floor((i + j) / 2)
7. return Astaroth(i, k, A, B) + Astaroth(k + 1, j, A, B)
```

Svar:

Begrunnelse:

6% *d*) En algoritme har et fast sett med instruksjoner som alltid utføres, et sett med instruksjoner der antallet steg er en fast prosent av problemstørrelsen og k rekursive kall på en k-del av problemet. Hva blir kjøretiden til algoritmen? Bruk Θ-notasjon og begrunn svaret med en kort skisse av utregningen din.

Svar:

Begrunnelse:

Oppgave 4 (10%)

Hvis vi fargelegger nodene i en graf og to nabonoder har samme farge kalles dette en konflikt. En graf er k-fargbar hvis den kan fargelegges med k farger uten at det oppstår konflikter.

4% *a*) Finnes det en kjent polynomisk algoritme for å avgjøre om en graf er k-fargbar, for en vilkårlig k? Hvis ja, hvordan virker denne, og hva er kjøretiden (i Θ-notasjon)? Hvis nei, hvorfor?

Svar:

2% *b*) Hvordan vil du løse problemet hvis $k = 2$? Angi øvre og nedre asymptotiske grenser for kjøretiden.

Svar:

4% *c*) Anta at du har en graf som *ikke* er tofargbar. Du ønsker å fargelegge den med to farger slik at antall konflikter blir minst mulig. Gi en kort beskrivelse av en enkel *branch and bound*-løsning på dette problemet.

Svar:

Oppgave 5 (15%)

For hver av algoritmene under, hvilke av de følgende asymptotiske kjøretidsklassene kan brukes til å beskrive algoritmens kjøretid? Du kan ikke anta noe om problem-instansene.

$\Omega(n)$, $\Omega(n \log n)$, $\Omega(n^2)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $\Theta(n)$, $\Theta(n \log n)$, $\Theta(n^2)$

Oppgi de relevante kjøretidsklassene (for eksempel "$\Omega(n^2)$, $O(n)$") eller skriv "Ingen" hvis ingen av klassene kan brukes. Hvis flere klasser kan brukes skal alle oppgis i svaret. Gi en kort begrunnelse til hver deloppgave.

5% *a*) Innsetting av n verdier i et tomt binært søketre.

Svar:

Begrunnelse:

5% *b*) Sortering ved innsetting.

Svar:

Begrunnelse:

5% *c*) MergeSort.

Svar:

Begrunnelse:

Oppgave 6 (15%)

Student Lurvik skal summere n positive flyttall, $x[1]…x[n]$, slik at avrundingsfeilen blir minst mulig. Avhengig av hvordan dette gjøres, kan de ulike tallene delta i et ulikt antall summasjoner. For eksempel vil $x[1]$, $x[2]$, $x[3]$ og $x[5]$ delta i flere summasjoner (tre) enn $x[6]$ (én) i følgende summeringsrekkefølge:

$$(((x[1] + x[3]) + (x[2] + x[5])) + x[6])$$

Merk her at både tall-rekkefølgen og parentes-settingen velges fritt.

Hvor stor avrundingsfeilen for en sum av to flyttall blir er avhengig av hvor stor summen er (større sum gir større feil). Lurvik innser at det kan være lurt å summere slik at de små flyttallene deltar i flest mulig summasjoner (det vil si, de kommer "dypt" i parentes-settingen) mens de store flyttallene deltar i færrest mulig. Hvis $p(i)$ er "parentes-dybden" til flyttall $x[i]$ (antall parenteser utenfor elementet) så definerer Lurvik *feilen* til en mulig løsning som

$$p(1)\cdot x[1] + p(2)\cdot x[2] + \ldots + p(n)\cdot x[\mathrm{n}].$$

Lurvik ønsker nå å finne en parentes-setting som minimaliserer dette feil-uttrykket. Skisser en løsning på problemet. Hvilken designmetode bruker du? Vis, med støtte i pensum, at løsningen er korrekt.

Svar: