Eksamen i fag
SIF8010 Algoritmer og Datastrukturer
Mandag 16. Desember 2002, kl 0900-1500

Faglig kontakt under eksamen: Arne Halaas, tlf. 41661982.
Hjelpemidler: Alle kalkulatortyper tillatt. Alle trykte og håndskrevne hjelpemidler tillatt.
VIKTIG: Alle svar anbefales skrevet inn på vedlagte Svarark. Tilleggsark kan benyttes om nødvendig. Før på "Student.nr." på hvert svarark.

Oppgave 1 (15 %)

Du skal her vise hvordan Floyd-Warshalls algoritme arbeider ved å fylle inn verdier i de 5 matrisene som følger etter startmatrisen:

1	2	3	4	5
0	2	9	∞	-3
∞	0	∞	1	7
∞	4	0	∞	∞
-2	∞	-5	0	∞
∞	∞	∞	5	0

Oppgave 2 (40 %)

Her skal du foreslå så effektive løsninger som mulig på 4 forskjellige algoritmeproblemer:

a) Foreslå en algoritme for å finne et svar på følgende ja/nei spørsmål:

Gitt de n heltallene $H_1, H_2, \ldots, H_n$. Er det mulig å finne et utvalg av eksakt k (< n) H-verdier slik at summen av disse verdiene er mindre enn en skranke T ?

Analyser kjøretiden for den beste algoritmen du finner.

b) Beskriv en liten modifikasjon av Quicksort som gjør at algoritmen får kjøretid O(n log n) også i verste tilfelle (worst case).

c) I en urettet graf G=(V,E) uttrykker nodene kjøreretninger (passeringsmåter) i et komplisert veikryss, mens kantene uttrykker at kjøreretninger er i konflikt med hverandre, dvs at kjøreretningene ikke kan benyttes samtidig fordi kollisjoner kan oppstå. Det er ikke kanter mellom noder som ikke er i konflikt med hverandre.
Anta at du vil finne ut om kjøreretningene kan deles i 2 grupper, de som skal ha rødt lys mens de øvrige har grønt, og omvendt.
Skisser kort en algoritme som avgjør om nodene i en slik konfliktgraf G kan deles inn i 2 grupper, der kjøreretninger (noder) i samme gruppe kan kjøre samtidig.
Algoritmen skal kun svare ja eller nei.

Analyser kjøretiden.

d) En tallsekvens $\{v_i\}$ som er slik at $v_1 < v_2 < \ \ldots\ < v_t > v_{t+1} > v_{t+2} > \ \ldots\ > v_n$ kaller vi her en opp&ned-sekvens. Den har topp-punktet v_t i posisjon t.
Vi skal her finne 2 effektive algoritmer, A1 og A2, som skal gjøre følgende:

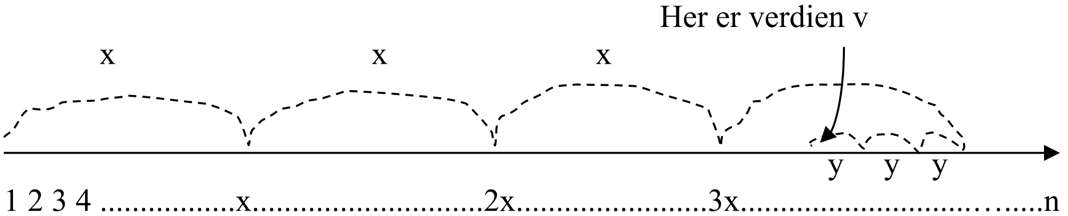
A1 skal avgjøre (ja/nei) om en gitt tallsekvens $\{v_i\}$ er en opp&ned sekvens.

A2 forutsetter at en gitt tallsekvens $\{v_i\}$ er en opp&ned sekvens og har som oppgave å finne topp-punktet.

Gi en kort skisse av de uavhengige algoritmene A1 og A2, og fyll inn skjemaet for analyse av algoritmenes kjøretider.

Oppgave 3 (15 %)

Stipendiat Odd Hopp har tatt mål av seg til å finne på noe lurere enn binær søking i sorterte tabeller A[1...n].
Odd vil innføre ”3-steg hopping” som alternativ. Hans ide er å hoppe med først steglengde x inntil søkt verdi v grovt er lokalisert, deretter hoppe med steglengde y, for så å avslutte med sekvensiell (nabo)søking. Ideen kan illustreres slik: (n>x>y)



a) Hva er asymptotisk kjøretid i verste tilfelle for Hopps metode ?
b) Er Hopp inne på gode ideer? Diskuter dette spørsmålet.

Oppgave 4 (15%)

Et sett av *n* aktiviteter er gitt ved deres start- og slutt-tidspunkter $start_i$ og $slutt_i$ for $i \in \{1, 2, \ldots, n\}$. Ingen aktiviteter begynner samtidig.
En aktivitet *A* sies å være dekket hvis det finnes en aktivitet *B* som tilfredsstiller $start_B \leq start_A$ og $slutt_A \leq slutt_B$. En effektiv algoritme for å finne ut hvor mange aktiviteter som er dekket har følgende enkle form:

Du skal her fylle inn (på Svararket) riktig kode for boksene A, B og C.

```
sorter aktivitetene etter stigende start-tidspunkt;
teller = 0
maks = - ∞
for i = 1 to n {
        if (?????? A ??????)

                ????? B ?????
        else
                ????? C ?????
}           // På svararket bes du også om å angi algoritmens kjøretid.//
```

Oppgave 5 (15 %)

Organisasjonen ISFIT-03 skal arrangere en festbankett for 200 jenter og 200 gutter. Hver person har fylt ut et skjema som viser personens ”interesseprofil”. Her er det krysset av ja/nei for ialt 25 interesseområder. Arrangørene vil prøve å lage en parvis bordsetting gutt/jente der parene får en mest mulig lik interesseprofil. For hvert par (g,j) gjelder det at differansen i interesseprofil, $d_{g,j} \geq 0$, er så liten som mulig. Det er ikke viktig for oss her hvordan d-verdiene beregnes.

For festbanketten vil vi oppnå å minimalisere summen

$\sum d_{g,j}$, der summen går over alle de 200 parene.

Oppgaven går ut på å vise hvordan du kan finne en optimal parsammensetting ved bruk av en gitt algoritme (S) for ”optimal sirkulasjon”. (Du skal altså vise hvordan du kan se på ISFIT’s parsammensettings-problem som et problem som dreier seg om å finne en ”optimal sirkulasjon”.)

Vi har en **”optimal sirkulasjon**” i en rettet graf (et nettverk) G = (V,E), med kantkostnader c(E), dersom vi finner et sett av (ukjente) heltallige flytverdier $\{f_{i,j}\}$ som tilfredsstiller følgende 3 krav: (Nb.: Alle verdiene er heltall.)

1. $l_{i,j} \leq f_{i,j} \leq h_{i,j}$ (d.v.s. hver kant har eget krav m.h.t. min./ maks. kantflyt.)

2. $\sum_{\text{Innkantene } \{i,j\} \text{ til node } j} f_{i,j} = \sum_{\text{Utkantene } \{j,k\} \text{ fra node } j} f_{j,k}$ (flyt inn = flyt ut $\forall$ noder.)

3. $\sum_{\text{Alle kanter } \{i,j\} \in E} (c_{i,j} * f_{i,j})$ er minimal ($c_{i,j}$ er kost pr. flytenhet for kant (i,j).)

Algoritme S, som finner ”optimal sirkulasjon”, skal ha følgende inn- og ut verdier:

Inn: [$L = \{l_{i,j}\}$, $H = \{h_{i,j}\}$, $C = \{c_{i,j}\}$] → **Algoritme S** → Ut: [$F = \{f_{i,j}\}$]

Inndata er kant-parametrene L, H og C, mens utdata er flytverdiene F.

Vis med en figur hvordan algoritmen S kan anvendes for å finne den optimale parsammensettingen ved banketten. Figuren skal bestå av en graf der de nødvendige parametre er påskrevet.

Merk: Her er det ikke mye som skal gjøres. Ikke gå løs på å tegne grafen i sin fulle størrelse !