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

Faglig kontakt under eksamen: Arne Halaas, tlf. 41661982.
Hjelpemiddel: Alle kalkulatortypar tillete. Alle trykte og handskrivne hjelpemiddel tillete.
VIKTIG: Vi anbefaler å skrive alle svar på det vedlagte svararket. Ein kan ta i bruk tilleggsark dersom dette er nødvendig. Før på "Student.nr." på kvart svarark.

Oppgåve 1 (15 %)

Du skal her vise korleis Floyd-Warshall si algoritme arbeidar ved å fylle inn verdiar i dei 5 matrisene som følgjer etter startmatrisa:

FRÅ \ TIL	1	2	3	4	5
1	0	2	9	∞	-3
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	-2	∞	-5	0	∞
5	∞	∞	∞	5	0

Oppgåve 2 (40 %)

Her skal du foreslå så effektive løysingar som mogleg på 4 forskjellige algoritmeproblem:

a) Foreslå ei algoritme for å finne eit svar på følgjande ja/nei spørsmål:

Gitt dei n heiltala $H_1, H_2, \ldots, H_n$. Er det mogleg å finne eit utval av eksakt k (< n) H-verdiar slik at summen av desse verdiane er mindre enn ein skranke T ?

Analyser køyretida for den beste algoritmen du finn.

b) Beskriv ein liten modifikasjon av Quicksort som gjer at algoritmen får køyretid O(n log n) også i verste tilfelle (worst case).

c) I ein uretta graf G=(V,E) uttrykker nodane køyreretningar (passeringsmåtar) i eit komplisert vegkryss, medan kantane uttrykker at køyreretningar er i konflikt med kvarandre, dvs at køyreretningane ikkje kan bli brukt samtidig fordi kollisjonar kan oppstå. Det er ikkje kantar mellom noder som ikkje er i konflikt med kvarandre. Anta at du vil finne ut om ein kan dele køyreretningane i 2 grupper, dei som skal ha raudt lys medan dei øvrige har grønt, og omvendt.
Skisser kort ei algoritme som avgjer om nodane i ein slik konfliktgraf G kan delast inn i 2 grupper, der køyreretningar (noder) i same gruppe kan køyrast samtidig. Algoritmen skal berre svare ja eller nei.

Analyser køyretida.

d) Ein talsekvens $\{v_i\}$ som er slik at $v_1 < v_2 < \;...\; < \; v_t \; > v_{t+1} > v_{t+2} > \;...\; > v_n$ kallar vi her ein opp&ned-sekvens. Den har topp-punktet v_t i posisjon t.
Vi skal her finne 2 effektive algoritmar, A1 og A2, som skal gjere følgjande:

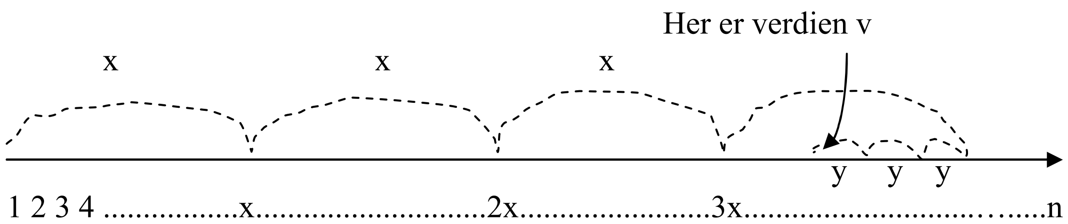
A1 skal avgjøre (ja/nei) om ein gitt talsekvens $\{v_i\}$ er ein opp&ned-sekvens.

A2 forutsetter at ein gitt talsekvens $\{v_i\}$ er ein opp&ned-sekvens og har som oppgåve å finne topp-punktet.

Gje ei kort skisse av de uavhengige algoritmane A1 og A2, og fyll inn skjemaet for analyse av køyretidene til algoritmane.

Oppgåve 3 (15 %)

Stipendiat Odd Hopp har tatt mål av seg til å finne på noko lurare enn binær søking i sorterte tabellar A[1...n].
Odd vil innføre ”3-steg hopping” som alternativ. Hans ide er å først hoppe med steglengde x inntil den søkte verdien v er grovt lokalisert, og så hoppe med steglengde y, for så å avslutte med sekvensiell (nabo)søking. Ideen kan illustrerast slik: (n>x>y)



a) Kva er asymptotisk køyretid i verste tilfelle for Hopps metode ?
b) Er Hopp inne på gode idear? Diskuter dette spørsmålet.

Oppgåve 4 (15%)

Eit sett av *n* aktivitetar er gitt ved deira start- og slutt-tidspunkt $start_i$ og $slutt_i$ for $i \in \{1, 2, \ldots, n\}$. Ingen aktivitetar byrjer samtidig.
Ein aktivitet *A* seiest å vere dekka viss det finnest ein aktivitet *B* som tilfredsstiller $start_B \leq start_A$ og $slutt_A \leq slutt_B$. Ein effektiv algoritme for å finne ut kor mange aktivitetar som er dekka har følgjande enkle form:

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

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

                ????? B ?????
        else
                ????? C ?????
}          // På svararket blir du også beden om å gi køyretida til algoritmen.//
```

Oppgåve 5 (15 %)

Organisasjonen ISFIT-03 skal arrangere ein festbankett for 200 jenter og 200 gutar. Kvar person har fylt ut eit skjema som viser ”interesseprofilen” til personen. Her er det kryssa av ja/nei for i alt 25 interesseområde. Arrangørane vil prøve å lage ein parvis bordsetting gut/jente der para får ein mest mogleg lik interesseprofil. For kvart par (g,j) gjeld det at differansen i interesseprofil, $d_{g,j} \geq 0$, er så liten som mogleg. Det er ikkje viktig for oss her korleis d-verdiane blir rekna ut.

For festbanketten vil vi oppnå å minimalisere summen

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

Oppgåva går ut på å vise korleis du kan finne ein optimal parsamansetting ved bruk av ei gitt algoritme (S) for ”optimal sirkulasjon”. (Du skal altså vise korleis du kan sjå på ISFIT sitt parsamansettingsproblem som eit problem som dreier seg om å finne ein ”optimal sirkulasjon”.)

Vi har ein **”optimal sirkulasjon**” i ein retta graf (et nettverk) G = (V,E), med kantkostnadar c(E), dersom vi finn eit sett av (ukjente) heiltalige flytverdiar $\{f_{i,j}\}$ som tilfredsstiller følgjande 3 krav: (Nb.: Alle verdiane er heiltal.)

1. $l_{i,j} \leq f_{i,j} \leq h_{i,j}$ (d.v.s. kvar kant har eige 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 følgjande inn- og ut verdiar:

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

Inndata er kant-parametrane L, H og C, medan utdata er flytverdiane F.

Vis med ein figur korleis ein kan anvende algoritmen S for å finne den optimale parsamansettinga ved banketten. Figuren skal bestå av ein graf der dei nødvendige parametrane er påskrivne.
Merk: Her er det ikkje mykje som skal gjerast. Ikkje gå laus på å teikne grafen i sin fulle storleik !