EKSAMEN I FAG 78010 / 45011
ALGORITMER OG DATASTRUKTURER

Lørdag 13. desember 1997, kl. 0900 - 1300

Faglig kontakt under eksamen: Arne Halaas, tlf 73 59 34 42
Alle kalkulatortyper tillatt. Alle trykte og håndskrevne hjelpemidler tillatt.

Merk: Det anbefales å ikke skrive mer enn totalt 1 side pr oppgave.

Oppgave 1 (15 %)

To lag A og B skal spille inntil 2n-1 kamper, der vinneren er det lag som først oppnår n seire. Ingen kamper ender uavgjort. For hver kamp er det samme sannsynlighet p for at lag A vinner, q = 1-p for at lag B vinner.

(a) (10%) Beskriv presist hva følgende rekursive funksjon beregner:

```
function P(i,j)
    if i = 0 then return 1
    else if j = 0 then return 0
    else return (p*P(i-1,j) + q*P(i,j-1))
```

(b) (5%) La T(k) være tiden i verste tilfelle for beregning av P(i,j), der k = i+j.
Beregn tidskompleksiteten for T(k) ved O( )- notasjonen

Oppgave 2 (30 %).

Gitt en n x n nabomatrise (adjacency matrix) K(1..n,1..n) som uttrykker personlige relasjoner mellom n personer. K(i,j) = 1 dersom person #i kjenner person #j, ellers er K(i,j) = 0.

Vi skal løse **Kjendis-problemet**: Finnes en person #p som ikke kjenner noen andre enn seg selv, men som alle de øvrige personene kjenner?

Problemet kan løses trivielt med en $\theta(n^2)$-algoritme der vi kan risikere å måtte undersøke alle elementene i K. Dette er ikke effektivt.

(a) (15%) Gi en ide til en $\theta(n)$-algoritme som løser Kjendis-problemet. Legg vekt på kun å få fram algoritmeideen, kort og konsist.

(b) (10%) Gi en kompakt beskrivelse (kvasikode) som viser hvordan du vil realisere ideen fra (a) i form av et program. Trivielle deler kan forklares med enkel tekst.

(c) (5 %) Begrunn kort hvorfor din algoritme har kompleksitet $\theta(n)$.

Oppgave 3 (30 %)

Nedover langs en stri elv ligger n utleiesteder for kano på rekke og rad. En kano kan leies på ethvert slikt sted og returneres på hvilket som helst annet utleiested lengre ned i elva. Det er ikke mulig å padle motstrøms. For hvert mulig avgangssted #i og ankomststed #j er det gitt en leiepris P(i,j). Prisene er slik at P(i,j) kan være høyere enn ved å leie flere forskjellige kanoer på strekningen mellom sted #i og sted #j. Det koster ikke noe ekstra å bytte kano.

(a) (20%) Finn en effektiv algoritme for å bestemme laveste pris for leie av kano(er) for å komme fra ethvert mulig avgangsted, #1,2,3, ... ,n-1, til ethvert mulig ankomststed #2,3, ... ,n lenger nede i elva. Din algoritme skal ta utgangspunkt i en kjent algoritme og modifisere denne til å løse problemet mest mulig effektivt.
Skriv opp den kjente algoritmen og anmerk (med korte begrunnelser) de endringer du vil gjennomføre.

(b) (10%) Beregn tidskompleksiteten til din algoritme under (a), og drøft denne i forhold til "den kjente algoritmen."

Oppgave 4 (25 %)

Alle verdiene i denne oppgaven er heltall.
Et lukket polygon P: (X1,Y1),(X2,Y2),...,(Xq,Yq), der (X1,Y1)=(Xq,Yq), består av q-1 punkter i planet. P avgrenser et **enkeltsammenhengende område**, d.v.s. at det finnes en eller annen sti S mellom ethvert punktpar innenfor P slik at S i sin helhet er innenfor P. Dersom dette gjelder også når vi krever at S er en rett linje, sier vi at P avgrenser et **konvekst** område. Se figurene nederst på siden.

Anta nå at vi skal løse følgende problem A:
Gitt n punkter (U1,V1),(U2,V2),...,(Un,Vn) og polygonet P som avgrenser et enkeltsammenhengende område. Finn hvilke av de n punktene som er innenfor P.
For enkelhets skyld skal du anta at ingen av de n punktene befinner seg på P's rand og at alle punktene ligger innenfor et rektangel som omskriver polygonet.

Gi en kort (2-10 linjers) besvarelse til hver deloppgave nedenfor. Du skal her ta utgangspunkt i en kjent innenfor/utenfor-algoritme, uten å skrive denne av.

(a) (5 %) Anta at P avgrenser et konvekst område. Skisser en algoritme som løser problemet A i $\theta(q\ n)$ tid. Gi kort forklaring.

(b) (10 %) Anta at P avgrenser et konvekst område. Skisser en algoritme som løser problemet A i $\theta((q+n) \log q)$ tid. Gi kort forklaring.

(c) (10 %) Anta at P avgrenser et enkeltsammenhengende område som ikke nødvendigvis er konvekst. Anta også at innenfor/utenfor-algoritmen finner høyst $\sqrt{q}$ skjæringer med randen, uansett hvilket punkt som testes. Hva er nå kompleksiteten til den mest effektive algoritmen som løser problemet A? Gi kort forklaring.

Eksempler: Et enkeltsammenhengende område og et konvekst område med sine omskrevne rektangler: