EKSAMEN I FAG 45011
ALGORITMER OG DATASTRUKTURER

Tirsdag 11. januar 1994 kl. 0900-1300

Faglig kontakt under eksamen: Arne Halaas, tlf. 3442

Alle trykte og håndskrevne hjelpemidler tillatt.
Godkjent lommekalkulator tillatt.

Merk: Alle svar skal skrives i de anviste plasser i oppgaveteksten. En besvarelse kan totalt gis inntil 100 poeng. Hvert oppgave gir inntil 20 poeng.

Oppgave 1

Gi O-grenser for $T(n)$ i hvert av tilfellene nedenfor. Løsningen skal kort forklares ved referanse til enten "Masterteoremet", bruk av iterasjon eller substitusjon.

a	$T(n) = T(n-2) + 1$
b	$T(n) = 2T(n/2) + n\text{lg}^2 n$
c	$T(n) = 9T(n/4) + n^2$
d	$T(n) = 3T(n/2) + n$
e	$T(n) = T(n/2 + \sqrt{n}) + n$

Oppgave 2

En $\Theta(n)$ tid algoritme etterspørres for å sortere de positive heltallene A[1], A[2], ..., A[n]. Ingen tall overskrider verdien n^3.

Foreslå en mulig sorteringsmetode, eller argumenter for at en slik metode ikke finnes. Svaret må begrunnes, evt. også med henvisninger til kjente metoder/teori.

Oppgave 3

a)
Er det mulig å benytte Bellman-Fords algoritme til å finne lengste vei(er) i en graf? Hvis du mener svaret er ja, hvilke minstekrav vil du da sette på G = (V,E)? Begrunn også et nei-svar.

I en rettet graf G = (V,E) har hver linje (u,v) $\in$ E tilknyttet en (desimal) verdi r(u,v) $\in$ [0,1] som uttrykker linjens grad av pålitelighet. Påliteligheten til en sti fra s til t er definert ved produktet av alle (uavhengige) r-verdier for linjer i stien.

b)
Vis kort hvordan delalgoritmene til DIJKSTRA(G,w,s) kan modifiseres slik at DIJKSTRA(..) kan benyttes for å finne en sti med maksimal pålitelighet.

c)
Istedet for å endre algoritmen DIJKSTRA som under (b) er det mulig å bruke algoritmen som den er dersom linjevektene velges riktig. Hvordan må vektene w(u,v) velges for at DIJKSTRAs algoritme, uten endringer, skal kunne brukes til å finne en sti med maksimal pålitelighet?

Oppgave 4

Gitt en rettet graf G=(V,E), der hver kant (u,v) har tilordnet et flytintervall [l(u,v), h(u,v)]. Vi sier at G har en *gyldig sirkulasjon* {f(u,v)} dersom f(u,v) $\in$ [l(u,v), h(u,v)] for alle (u,v) $\in$ E. Her er l(u,v) og h(u,v) laveste h.h.v. høyeste tillatte verdi for flyten f(u,v), og $\sum_i f(i,v) = \sum_j f(v,j)$ for alle v$\in$V.

Her er {(i,v)} mengden av alle kanter som går inn til v, mens {(v,j)} er mengden av alle kanter som går ut fra v. (**flyt inn = flyt ut** for alle noder).

Anta at vi har adgang til en O(|V| + |E|) algoritme **A** som enten finner en *gyldig sirkulasjon* i G=(V,E) eller rapporterer at en *gyldig sirkulasjon* ikke finnes.

Vis hvordan vi kan bruke algoritmen **A** som kjerne til å løse et ordinært flytmaksimeringsproblem.
Hva blir kompleksiteten til denne flytmaksimerings-algoritmen?

Oppgave 5

Vi skal her arbeide med det vi har kalt "generell" datastruktur for rettede nettverk (grafer), der vi har basert oss på følgende type-erklæringer:

```
TYPE Kantref   = ^Kant ;
     Kant      = RECORD
                     InfoKant       : .... ;
                     Bnode, Enode   : NodeRef;
                     Nik, Nuk       : KantRef;
                 END;

     NodeRef   = ^Node;
     Node      = RECORD
                     InfoNode       : .... ;
                     Fik, Fuk       : KantRef:
                     Merket         : Boolean;
                 END;
```

Her er *Bnode* og *Enode* referanser til de noder der en kant begynner hhv. ender, *Fik* og *Fuk* er referanser til første ut-kant hhv. første inn-kant i kantlistene knyttet til en node, og *Nik* og *Nuk* er de tilhørende videre referanser (fra kant til kant) i disse listene. Merk at hver kant på denne måten blir medlem av to lister, idet den går ut fra en node og inn til en annen node.

Datastrukturen ovenfor skal benyttes i funksjonen SYKELFRI som skal returnere TRUE hvis og bare hvis et nettverk ikke inneholder sykler. SYKELFRI skal inneholde den rekursive prosedyren NODEVISITT. Fyll ut programskjelettet nedenfor til et fullstendig PASCAL-program. (Dersom du ikke kan PASCAL kan C benyttes). Anta at alle noders verdi av "merket" er TRUE før SYKELFRI(START) kalles. Anta også at alle noder i nettverket direkte eller indirekte kan nås fra START-noden.

Svar:

```
Function SYKELFRI (startnode: NodeRef) : boolean;

    Procedure NODEVISITT (nestenode:NodeRef);

    begin









    end;

begin



   SYKELFRI:=
end;
```