Student nr:

KONTINUASJONSEKSAMEN I FAG

45011 ALGORITMER OG DATASTRUKTURER

Torsdag 14. august 1997, kl 0900-1300

Faglig kontakt under eksamen: Knut Magne Risvik, tlf 73 594489
Alle kalkulatortyper tillatt. Alle trykte og håndskrevne hjelpemidler tillatt.
Alle svar avgis i angitte svarruter i oppgaveteksten. Vedlegg evt. kladdeark / utregninger som du mener er viktige for å vurdere et svar. Henvis ved "Se kladd" i margen ved svarruten.
Fyll inn rubrikken "Student nr." på alle ark.

Oppgave 1 (15%)

a) Vis at $(n+1)^2 = O(n^2)$ ved å bestemme konstantene n_0 og c i definisjonen av O-notasjonen. Konstantene skal være heltallige og minimale.

Svar: (5%) Må ha $(n+1)^2 \leq c \cdot n^2$ for alle $n \geq n_0$.
$(n+1)^2 \leq 2n^2$ hvis $2n + 1 \leq n^2$, altså hvis $n \geq 3 = n_0$.

$n_0 = 3$ $c = 2$

b) Gitt at $T(n) = T(n-2) + \lg n$. Vis, eller motbevis, at $T(n) = \Omega(n \lg n)$

Svar: (10%)

$T(n) = T(n-2) + \lg n$
$= T(n-2\cdot2) + \lg(n-1\cdot2) + \lg n = \ldots = T(n/2-2) + \underbrace{\lg(n-n/4\cdot2) + \ldots + \lg(n-2\cdot2) + \lg(n-1\cdot2) + \lg n}_{n/4 \text{ ledd, det minste er } \lg(n-n/4\cdot2) = \lg(n/2)}$

$\geq n/4 \lg (n/2)$,
Altså er $T(n) = \Omega(n \lg n)$

Oppgave 2 (35%)

Formelen $A(b) = A(b-1) + A(b-2) + A(b-5)$ kan, for $b > 5$, benyttes til å beregne antall forskjellige måter et beløp b kan puttes på en pengeautomat der kun myntstørrelsene 1, 2 og 5 er tillatt brukt.

Vi har at $A(1) = 1$, $A(2) = 2$. Vi finner videre at $A(3) = 3$ fordi beløpet 3 kan oppnås ved å putte på mynter på en av de 3 måtene (1,1,1), (1,2) eller (2,1). Merk at vi skiller mellom (1,2) og (2,1).

a) Finn A(4), A(5), A(6) og A(7)

Svar: (5%)
A(4) = 5 , A(5) = 9 , A(6) = 15 , A(7) = 26

(b) Skriv en rekursiv funksjon som beregner A(b). (Bruk Pascal, C, C++ eller pseudokode)

Svar: (5%)

```
function A(b:integer) : integer;
if b  = 1 then return 1 else if … else if b = 5 then return 9;
else return A(b-1) + A(b-2) + A(b-5);
```

Student nr:

(c) Finn tidskompleksiteten til funksjonen (b) angitt ved Ω(…)- notasjonen..

Svar: (10%) $T(n) > 3T(n\text{-}5) > 3^2T(n\text{-}2\cdot 5) > \ldots > 3^kT(n\text{-}k\cdot 5)$; $n\text{-}k\cdot 5 = 5 \Rightarrow k = \frac{n}{5} - 1$

$\Omega\ (\ 1.25^n\ )$ idet $T(n) > 3^{(n/5-1)}$, $c = \frac{1}{3}\sqrt[5]{3^n} \approx \frac{1}{3}1.25^n$

(d) Skriv et programavsnitt der dynamisk programmering benyttes til å beregne A(b).

Svar: (10%)

```
A[1...5] ← 1,2,3,5,9;
for i := 6 to n do
        A[I] ← A[i-1] + A[i-2] + A[i-5];
```

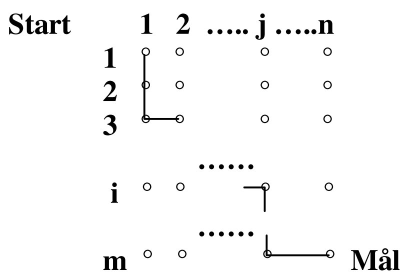
(e) Finn tidskompleksiteten til programavsnittet (d) angitt ved Θ(….)-notasjonen.

Svar: (5%)

Θ (n)

Oppgave 3 (20%)

Gitt (m × n) punkter i et rektangulært rutenett:



Punktet [1,1] kaller vi "Start" og punktet [m,n] "Mål".
En ***lovlig*** vei fra Start til Mål defineres ved at et ***skritt*** fra punkt [i,j] på veien skal gå enten til punktet [i +1,j] eller til punktet [i,j+1]. To veier er ***forskjellige*** dersom de ikke er identisk like, skritt for skritt.
Vårt problem er å beregne antallet forskjellige veier (**v[n,m]**) fra Start til Mål.
Vi har eksempelvis at **v[3,2] = 3, v[2,2] = 2,** mens **v[3,3] = 6.**

a) Vis hvordan **v[n,m]** kan beregnes ved dynamisk programmering**:**

Svar: (15%)
Initialisering: v[1,j] ← 1 for j ∈ [1..n]; v[j,1] ← 1 for i ∈ [1..m];

(Fyll inn startverdier)

v[i,j] := v[i-1,j] + v[i,j-1] for i∈ [2,m], j∈ [2,n] (Utarbeid formel)

b) Angi kompleksiteten til et program for beregning av **v[n,m]** ved Dynamisk Programmering.
Bruk Θ(….)-notasjonen.

Svar: (5 %)
Θ(n m) (Fyll inn i parantesen)

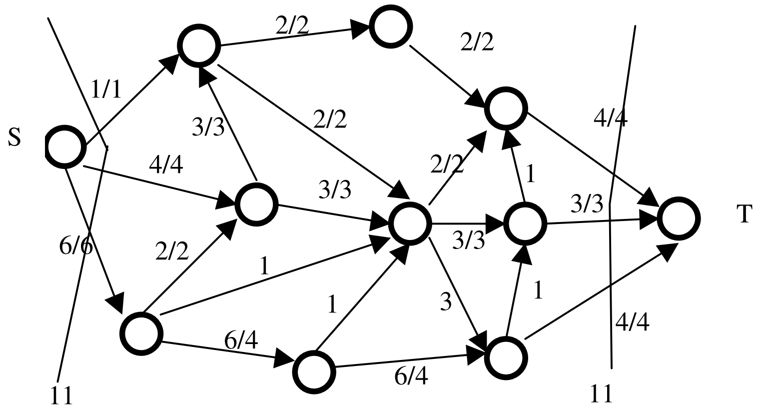
Oppgave 4 (15%)

Skisser en Θ(n)-algoritme for sortering av n heltall X[1..n], der
en er garantert at X[i] ∈ {1,4,7,28} for i∈[1..n].

Svar: (15%)
Fase 1: Tell sekvensielt alle sorter tall; Θ(n)
Fase 2: "Rull ut" tallene etter stigende verdier; Θ(n)

Oppgave 5 (15%)

Gitt følgende flytnettverk med linjekapasiteter påført:



Finn en maksimal flyt fra node S til node T i nettverket.

Svar: (15%) Flyten for alle linjene skal skrives på figuren og et minimalt snitt skal tegnes inn.