



KANDIDAT

495

PRØVE

# DAT201 G Algoritmer og datastrukturer

Emnekode	DAT201
Vurderingsform	Skriftlig eksamen
Starttid	25.11.2015 08:00
Sluttid	25.11.2015 12:00
Sensurfrist	16.12.2015 00:00
PDF opprettet	17.11.2019 12:59

Oppgave 1

Oppgave	Tittel	Oppgavetype
1	Forside DAT201Høst 2015	Flervalg
2	Oppgave 1a	Programmering
3	Oppgave 1b	Programmering

Oppgave 2

Oppgave	Tittel	Oppgavetype
4	Oppgave 2a	Langsvar
5	Oppgave 2b	Langsvar
6	Oppgave 2c	Langsvar
7	Oppgave 2d	Langsvar
8	Oppgave 2e	Langsvar

Oppgave 3

Oppgave	Tittel	Oppgavetype
9	Oppgave 3a	Langsvar
10	Oppgave 3b	Programmering
11	Oppgave 3c	Programmering
12	Oppgave 3d	Programmering

Oppgave 4

Oppgave	Tittel	Oppgavetype
13	Oppgave 4a	Langsvar
14	Oppgave 4b	Langsvar

Oppgave 5

Oppgave	Tittel	Oppgavetype

DAT201 G Algoritmer og datastrukturer			Candidate 495
15	Oppgave 5a	Langsvar	
16	Oppgave 5b	Langsvar	

1

## Forside DAT201Høst 2015

**Emnekode:** DAT201  
**Emnenavn:** Algoritmer og datastrukturer

**Dato:** 25.11.2015  
**Varighet:** 4 timer  
**Tillatte hjelpemidler:** Ingen

**Merknader:**  
Ved bedømmelsen av besvarelsen vil det bli lagt vekt på ryddighet, godt strukturerte algoritmer / pseudo-koder / program og god dokumentasjon.

Ta dine egne forutsetninger (husk kommentar) hvis du i noen av oppgavene finner uklarheter og/eller mangler.

Dersom du blir bedt om å skrive kode, så kan du enten skrive Java kode eller skrive i et annet datalignende språk. Dette i motsetning til det å skrive en algoritme som kan gjøres med norsk tekst.

**Håndskrevne tegninger og skisser leveres på ark merket kandidatnummer.**

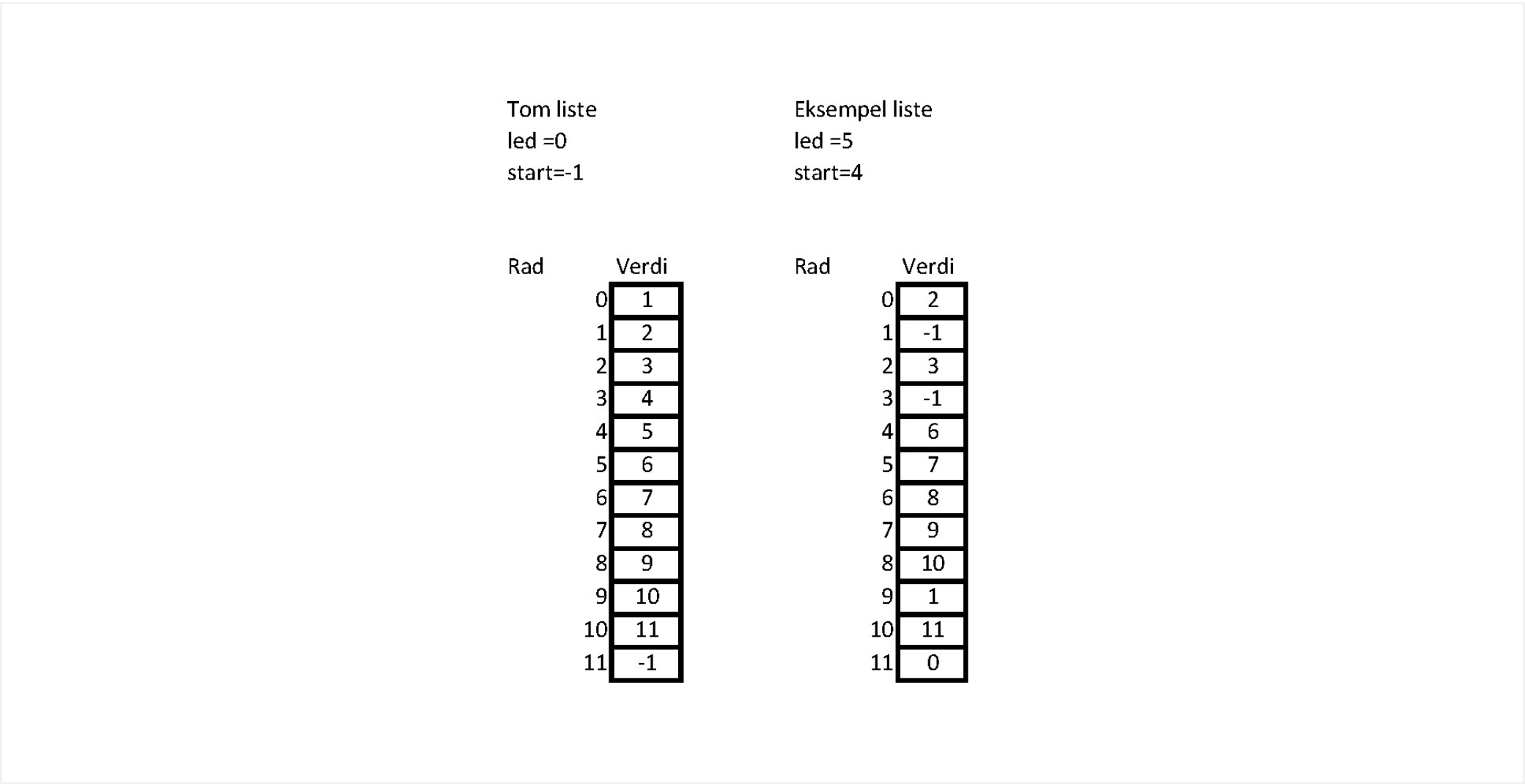
-----  
Det forekommer av og til spørsmål om bruk av eksamensbesvarelser til undervisnings- og læringsformål. Universitetet trenger kandidatens tillatelse til at besvarelsen kan benyttes til dette. Besvarelsen vil være anonym.  
Tillater du at din eksamensbesvarelse blir brukt til slikt formål?

☐ Nei

☒ Ja

2

## Oppgave 1a



Figuren viser en enkel lenka, sirkulær liste uten dummyelement. Variabelen start er indeks (radnummer) til

første element i lista. Dersom verdien er -1, så er lista tom. Variabelen led er indeksen(radnummeret til første ledige rad. Dersom led er -1, så er lista full. Ledig lista en enkel liste uten dummyelement (ikke sirkulær)

Du har en klasse som skal implementere denne datastrukturen

```
public class Stakk{
    private int led=0;
    private int start =-1;
    private int[] values = new int[12];
    public boolean push(int value){
        //din kode
    }
}
```

Skriv koden til metoden

```
public boolean push(int value)
```

som legger verdien value inn sist i lista.

For å få fullgost svar, må du håndtere alle feilsituasjoner og koden push sammen med koden pull i oppgave 1b) skal tilsammen implementere en fungerende stakk.

Metoden skal returnere false hvis innsetting ikke gikk, true ellers.

**Skriv ditt svar her...**

1	Du har en klasse som skal implementere denne datastrukturen	
2		
3	public class Stakk{	
4	private int led=0;	
5	private int start =-1;	
6	private int[] values = new int[12];	
7		
8	public boolean push(int value){	
9	for (int i; i>12; i++);	
10	if (i==12);	
11	try{	
12	stakk.add(i));	
13		
14	catch{nullpointer.exeption}	
15	expect return False;	
16	else {return True};	
17		
18	// Her saa skal koden teste aa legge til et nytt tall i en try catch loop, som tar i mot om den skulle faa en nullpointer exeption, og så returnere false, hvis annet så returnere den True	
	.	

3 Oppgave 1b

Du har en klasse som skal implementere denne datastrukturen

```
public class Stakk{
    private int led=0;
    private int start =-1;
    private int[] values = new int[12];
    public boolean push(int value){
        //din kode fra 1a)
    }
    public int pop(){
        //din kode
    }
}
```

Skriv koden til metoden

```
public int pop()
```

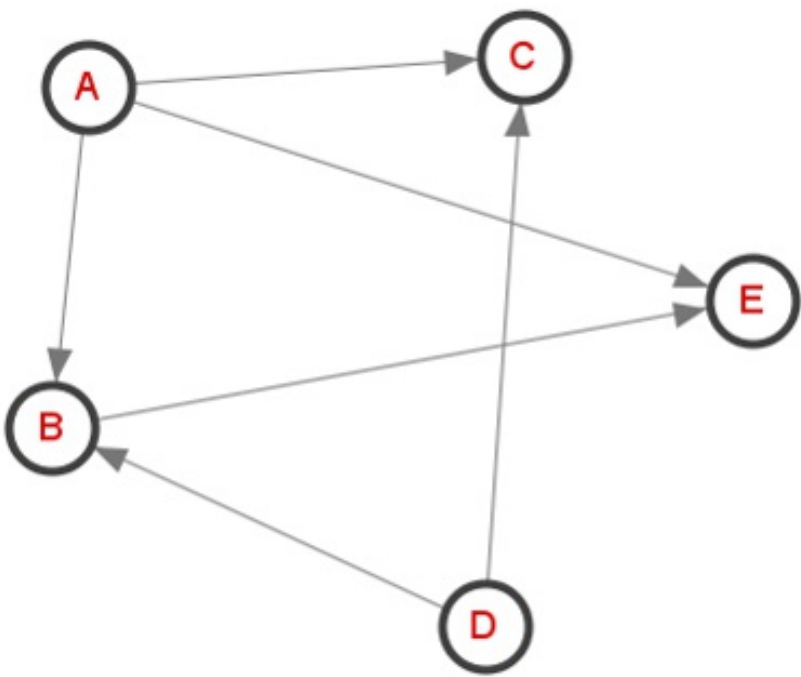
som henter og fjerner sist verdi fra lista.

For å få fullgodt svar, må du håndtere alle feilsituasjoner og koden push sammen med koden pull i oppgave 1b) skal tilsammen implementere en fungerende kø.

Skriv ditt svar her...

```
1 public class Stakk{
2     private int led=0;
3     private int start =-1;
4     private int[] values = new int[12];
5
6     public boolean push(int value){
7         for (int i; i>12; i++);
8             if (i==12);
9                 try{
10                     stakk.add(i));
11
12                     catch{nullpointer.exeption}
13                         expect return False;
14                     else {return True};
15
16             }
17
18     public int pop(){
19         for (int i; i>12;i++)
20             if (i>12)
21                 system.out.println(i);
22             if (i == 12)
23                 this.remove(i[12]);
24                 system.out.println("The Last element in the list has been Removed.")
25
26         }
27     }
28
29     // Jeg vil forutse at koden nok ikke kommer til å funke, med grunn lag at jeg ikke har en
30     //ordentlig debugger som kan sjekke for feil, men teoretisk sett så skal denne være rett.
31     //Men her så skal koden først sjekk om i er mindre enn 12, hvis den er det skal den
32     //incrementere med 1.
33     // Når i er lik 12, så skal den slette det elemente som har indeksen 12.
34     //
```

4 Oppgave 2a



Hvordan definerer du en graf ved hjelp av en nabomatrise.  
Figuren viser en rettet graf. Skriv nabomatrisen til denne grafen.

Skriv ditt svar her...

Nabomatrisen  
til Grafen

	A	B	C	D	E
A	0	1	1	0	1
B	0	0	0	0	1
C					
D	0	1	0	0	0
E					

På figuren vises det en u-vektet rettet graf, fra A - E

Nabomatrisen sier hvilken bokstav som peker til hvilken bokstav, og når bokstaven ikke peker til noe så er hele kolonnen tom, forutsatt at det er bare nuller her.

Ved hjelp av noabo matrisen kan vi også finne inn og utgrad til de forskjellige Nodene, Kolonner er inngraden og Rader er ut gradene. Så her kan vi da se at A, har 3utgrader og ingen inngrader.

## 5 Oppgave 2b

Er nabomatriksen til en urettet graf symmetrisk? Begrunn svaret.

Kan nabomatriksen til rettet graf være symmetrisk? Gi et eksempel.

**Skriv ditt svar her...**

### **Er nabomatriksen til en urettet graf symmetrisk? Begrunn svaret**

Nei, for om en graf skulle være urettet så vil det ikke finnes en nabomatrise heller, Nabomatriksen kan skrives ved at grafen er rettet, siden grafen ikke er rettet så kan vi derfor ikke vite/ si noe om en node, om den peker til den neste noden eller den andre.

### **Kan nabomatriksen til rettet graf være symmetrisk? Gi et eksempel**

Nabomatriksen til en rettet graf være symmetrisk.. både ja og nei, det eneste som kan være symmetriskt er diagonalen, om den inneholder enere(1) eller nuller(0) fordi om det er enere eller nuller på diagonalen sier om grafen er Asyklisk eller Usyklisk. Om vi skulle for eksempel ha bare enere på diagonalen, så ville grafen være Syklisk og da vil jo Diagonalen mest sannsyelig være det eneste som er symmetriskt i grafen.. men ellers så spiller d

6

Oppgave 2c

Er grafen i starten av oppgave 2 en aktivitetsgraf?  
Hvis den er det, finn en topologisk sortering.

Skriv ditt svar her...

Grafen i starten av oppgave 2, er en aktivitetsgraf... grunnen til at den er en aktivitetsgraf er fordi den er Asyklisk, hadde den vært syklisk så kunne den ikke været en aktivitetsgraf.

Topologisk Sortering; ABCDE

7

Oppgave 2d

Hvor mange veier med 2 kanter er det mellom A og E? Skriv svaret med å angi nodene som definerer kantene.  
Finn svaret med regning.

Skriv ditt svar her...

Hvis vi ser på nabomatrisen som vi fant i Oppgave 2a

	A	B	C	D	E
A	0	1	1	0	1
B	0	0	0	0	1
C					
D	0	1	0	0	0
E					

Så kan vi finne alle veiene med 2 kanter mellom **A** og **E**.  
Her ser vi da at B-C er den eneste veien som har 2 kanter mellom seg.  
som ligger mellom nodene **A** og **E**.



## 8 Oppgave 2e

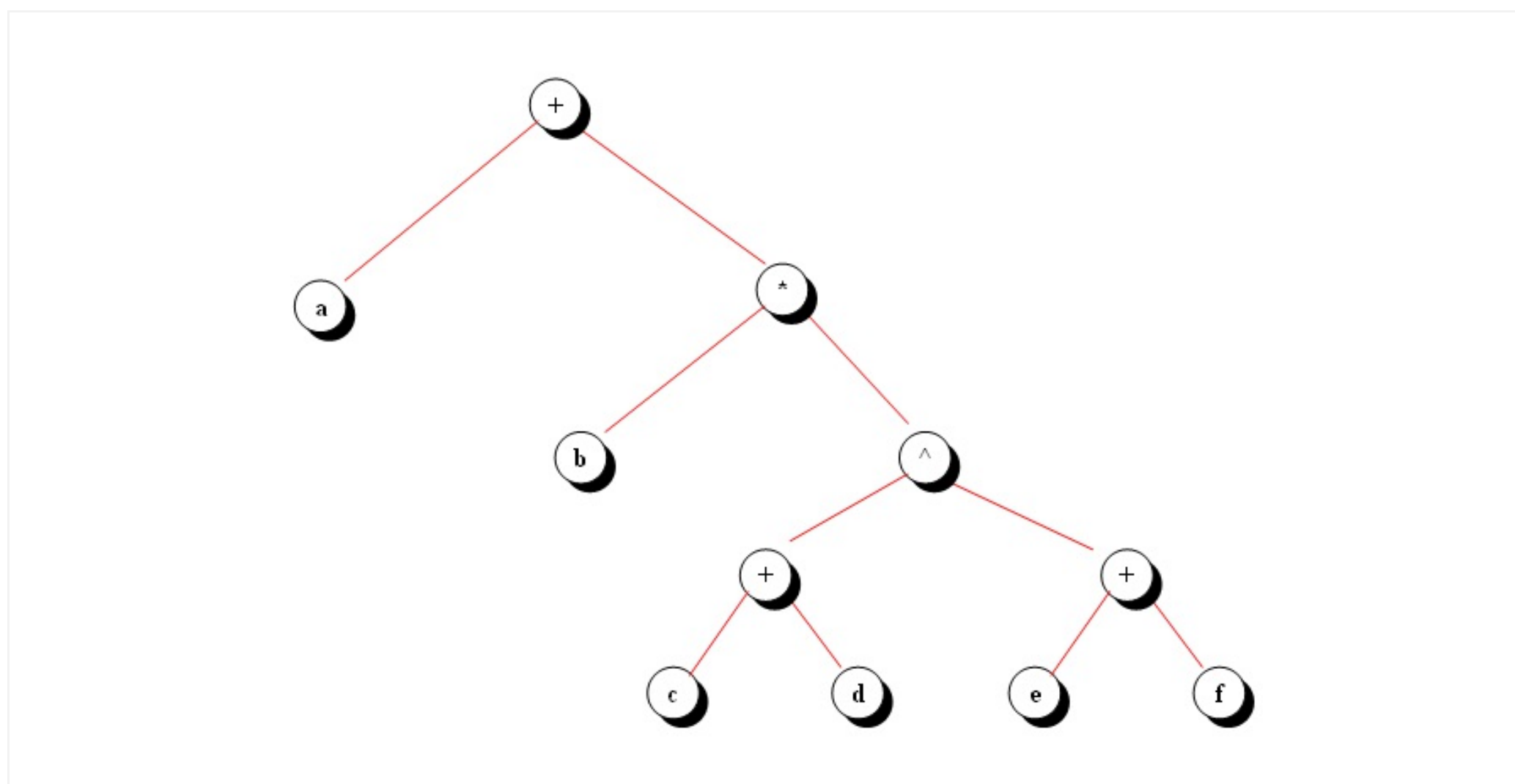
Grafen er asyklisk. Hvordan kan du vise det ved regning?

**Skriv ditt svar her...**

Normalt så kan vi kikke på grafen og se om den enten er syklisk eller asyklisk.. Skulle vi være usikker ovenfor dette og måtte finne det ut ved regning så kan man da gange sammen alle matrisene ( $M^n$ ) så her så måtte vi ha ganget sammen disse matrisene ( $M+M^2+M^3+M^4+M^5$ ) også skulle vi kunnet sett på diagonalen i nabomatrisen om den var syklisk eller asyklisk. Hvis Diagonalen hadde inneholdt bare 1(enere) så var grafen Syklisk, men hadde Diagonalen inneholdt bare 0(nullere) så ville den vært Asyklisk om så bare noe av diagonalen hadde vært fylt med enener og resten nuller, så vil den fortsatt være Asyklisk.

Så om man skulle se på grafen å være usikker på om den er Asyklisk eller Syklisk så kan man bruke regning til å finne dette ut.

## 9 Oppgave 3a



Det er tre måter å traverser et binært tre med metoden dybde først – preorden, inorden og postorden. Traverser (gå gjennom) treet i figuren med metoden postorden. Skriv opp verdiene av nodene i den rekkefølgen du da vil traversere nodene.

**Skriv ditt svar her...**

Siden det er Postorder vi skal bruke til å traversere nodene med så vil dette si at vi "printer" verdien på noden når vi er på høyre side av noden.

Se tegning som er lagt med (**Ark1**).

Dette er rekkefølgen til nodene med verdiene når dem blir traversert med Postorder.

**a, b, c, d,+,e,f,+,^,\*,+**

10 **Oppgave 3b**

Skriv pseudokode/kode for hvordan du kan skrive en rekursiv metode  
public String postOrden(Node n)  
i for eksempel Java for å traversere et tre med metoden postorden. Metoden skal returnere en tekst med  
nodens verdi(navn) i postorden rekkefølge.

Du kan anta du har en klasse Node

```
public class Node{
    Node left=null,right=null;
    String value=null;
    // og de tilhørende get/set metodene
}
```

**Skriv ditt svar her...**

1	public class Node{	
2	Node left=null,right=null;	
3	String value=null;	
4		
5	this.getLeft();	
6	return getLeft();	
7		
8	this.setLeft(String l);	
9		
10	this.get Rigth();	
11	return getRigth();	
12		
13	this.setRigth(String r);	
14	}	
15		
16	if (focus.getLeft().setLeft(left))	
17	(focus.getRigth().setRigth(value));	
18	system.out.println(value)	
19		
20		
21		
22	}	

11 Oppgave 3c

Skriv pseudokode/kode for hvordan du kan skrive en ikke-rekursiv metode  
public String postOrden(Node n)

i for eksempel Java for å traversere et tre med metoden postorden.

Du kan anta du har en klasse Node

```
public class Node{
    Node left=null,right=null;
    String value=null;
    // og de tilhørende get/set metodene
}
```

Skriv ditt svar her...

```
1 public String postOrden(Node n)
2
3     for (int i; i++)
4         if n.getLeft().setLeft(l)
5             if n.getRigth().setRigth(r)
6
7
8
9
10    public class Node{
11        Node left=null,right=null;
12
13        String value=null;
14
15        this.getLeft();
16        return getLeft();
17
18        this.setLeft(String l);
19        (left = l);
20
21        this.get Rigth();
22        return getRigth();
23
24        this.setRigth(String r);
25        (rigth = r);
26    }
27 }
28
29 //Algoritmen for PostOrder gå jo som følger
30 // 1. Man skal først finne ut hvor mange noder det er
31 // 2. Så skal man Begynne på Venstre side.
32 // 3. Når man passerer en node fra Høyre siden, så skal noden sin Verdi bli printet ut.
33 // 4. Så sjekker den om den kan gå lengre.
34 // 5. Begynner den fra Steg 2.
35 og holder på slik til den er til bake til rota.
```

12   **Oppgave 3d**

En klasse har Arithm skal ha en metode evaluate. Den har to parametre. Den ene er roten til et binært tre som representerer et matematisk uttrykk slik som beskrevet tidligere i oppgaven Den andre er en HashMap hm som inneholder verdiene til variablene i treet, f.eks. a,b,...

Skriv koden/pseudokoden til metoden evaluate som beregner verdien til det matematiske uttrykket.

Du kan forutsette at de eneste regneoperasjonen er + (addisjon), \* (multiplikasjon) og ^(eksponent - Math.pow(a,n))

Hint: Når du leser utskriften fra oppgave 3a fra venstre til høyre som en liste: hver gang du finner en regneoperasjon, så skal det være to tall til venstre (evt varable du kan finne verdien til i hm)  
F.eks. hvis det står c d +, så skal de tre elementene fjernes fra lista og erstattes med verdien av uttrykket.

**Skriv ditt svar her...**

1	public void evaluate (Node n){	
2		
3	hashmap =	
4	binarytree =	
5		
6	}	

13

Oppgave 4a

I en tekst med 100 bokstaver er frekvensen av de enkelte bokstavene gitt av tabellen under.

Bokstav	A	E	F	L	N	T	Å
Frekvens	24	27	9	12	12	14	2

Huffmans algoritme kan brukes til å lage en bitkode for hver bokstav i tabellen. Dette kan gjøres ved å bygge et binærtre *-bt*.

Skriv koden for en klasse Node som du kan bruke for å lage binærtreet *bt*.

Bruk denne klassen til å beskrive hvordan Huffmans algoritme brukes for å konstruere *bt*.

Skriv ditt svar her...

<b>Bokstav</b>	<b>E</b>	<b>A</b>	<b>T</b>	<b>L</b>	<b>N</b>	<b>F</b>	<b>Å</b>
<b>Bitkode</b>	000	001	010	011	100	110	111

Tegning av Huffman Binærtre se Tegning (**ARK 2**)

Skriv koden for en klasse Node som du kan bruke for å lage binærtreet *bt*.

```
public class void Node(Node n){

}
```

Bruk denne klassen til å beskrive hvordan Huffmans algoritme brukes for å konstruere *bt*.

For å kunne konstruere et binær Tree for å kunne finne ut hva bit kodene er til de forskjellige bokstavene, så kan man generelt bare lage et normalt binært tre med noder og verdi for nodene, men eneste forskjellen er at på høyre siden av leddene så skal det stå 1, og på venstre siden skal det stå 0

**14 Oppgave 4b**

Anta du har et binærtre som er konstruert med Huffmans algoritme. Forklar hvordan du kan bruke dette til å lage en bitkode for hver bokstav.

Du kan enten forklare det med ord, pseudo-kode eller kode.

**Skriv ditt svar her...**

For å lage en bitkode for hver av bokstavene i huffmans algoritme, så tegner du opp et binær tre som jeg har tegnet på Ark 2, Hvor jeg da laget et "normalt" binær tre med noder med verdier. Bare her har vi satt entne en 1 eller en 0 på et av leddene så vi kan lese av fra toppen og ned hvilken bitkode, f.eks F har, da går man fra toppen og følger leddene ned igjen om så ser man at F... har bitkoden 110.

**15 Oppgave 5a**

I) Forklar hva en *hashcode* er.

II) Forklar hvilke egenskaper du mener en god metode for lage hashcode skal ha.

III) Beskriv en god måte å lage en *hashcode* fra en tekst (dvs en java String)

IV) Forklar hva en *hashfunksjon* er.

**Skriv ditt svar her...**

**I) Forklar hva en hashcode er.**

En hashcode, er generelt sett et nummer eller en string som har blitt "scramblet" om til en hashcode... så en hashcode er en string som inneholder både bokstaver og tall. Det er for å kunne slippe å representere et passord i klartekst så hasher dem det, med hjelp av en matematisk ligning som gjør det umulig for oss mennesker å løse hashen ved mindre vi har en maskin som kan cracke hashen for oss og finne ut hva som står bak hash'en

**II) Forklar hvilke egenskaper du mener en god metode for lage hashcode skal ha.** å ta oversette stringen til char's og så char'en og ta modulus 5 av den.

**III) Beskriv en god måte å lage en hashcode fra en tekst (dvs en java String)**

**IV) Forklar hva en hashfunksjon er.**

er en funksjon man bruker for å kunne hash'e ting ved bruk av en samme matematisk formel omigjen ... osv...



**16 Oppgave 5b**

En tabell der objekter plasseres ved hjelp av hashing, kalles en *hashtabell*. Posisjonen (eller indeksen)  $k$  i tabellen der et objekt blir plassert, kalles objektets adresse.

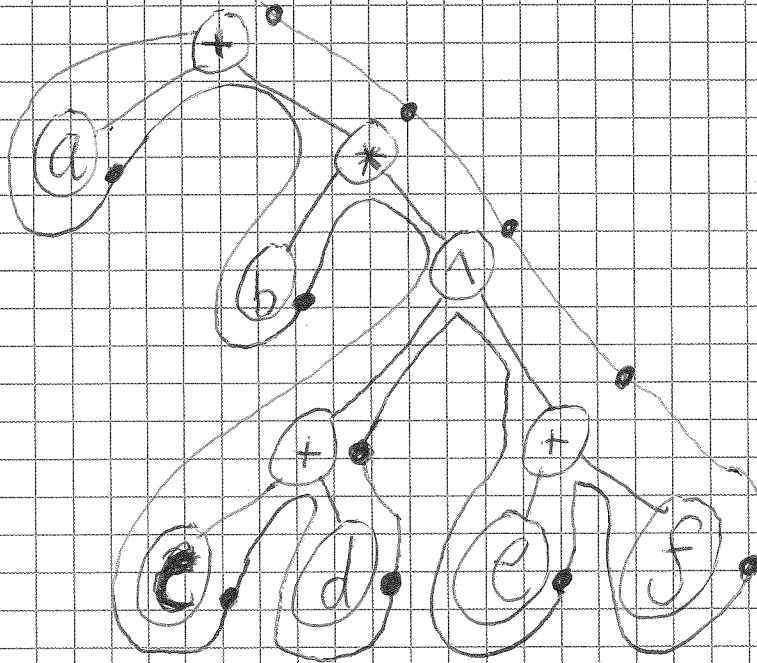
Forklar hva som i denne sammenheng menes med lukket adressering.

**Skriv ditt svar her...**

Lukket adressering menes med det at man da tar og hash'er index og objektets adresse, ..



oppgave  
3a)



Ved hver + punkt på "banen" så printes verdien i noden

Rækkefølgen ved brug av postorden  
a, b, c, d, +, e, f, +, ^, \*, +

