Introduksjon til DAT200 eksamen

EKSAMEN I: **DAT200 Algoritmer og Datastrukturer**

DATO: 7. mars 2019
TID FOR EKSAMEN: **4 timer**
TILLATTE HJELPEMIDDEL: Ingen trykte og håndskrevne hjelpemidler, standard enkel kalkulator

Merknad: Du må lese instruksjonene før du starter med å løse eksamensoppgavene!

For oppgavene som krever programmering: Det viktigste er at du viser at du forstår hvordan en løsning kan programmeres i Java. Enkle skrivefeil i navn på metoder vil ikke trekke ned. Å bruke et annet navn på en metode så lenge det går klart fram hva du prøver å gjøre vil heller ikke trekke ned.

Når du programmerer kan du bruke C++-lik syntaks for å indikere at en metode tilhører en klasse. Dette betyr at i stedet for å skrive metoden inni klassen så skriver du klassenavn::metodenavn.

Du trenger ikke å inkludere import-setninger i programmene dine med mindre oppgaven eksplisitt spør om det.

1 Kjøretid, enkel metode

Gitt følgende metode:

```
public static int finn(int[] tall, int verdi) {
        for (int i=0;i<tall.length;i++) {
                if (tall[i] == verdi) return i;
    }
        return -1;
  }
```

Hva er worst-case kjøretida til denne metoden? (2%)

Velg ett alternativ

- O(n*log(n))
- O(log(n))
- O(1)
- O(n)
- O(n^2)
- O(n^3)
- O(2^n)

Hva er best-case kjøretida til denne metoden? (2%)

Velg ett alternativ

- O(1)
- O(log(n))
- O(n)
- O(n*log(n))
- O(n^2)
- O(n^3)
- O(2^n)

Maks poeng: 4

2 Kjøretid, mer komplisert algoritme

Hva er kjøretida til metoden oppgitt under i O-notasjon, hvor n er "storrelse" verdien? (6%)

Velg ett alternativ

- O(1)
- O(log(n)
- O(n)
- O(n*log(n))
- O(n^2)
- O(n^3)
- O(2^n)

```
public static void tegnDiamant(int storrelse) {
      for (int j = 0; j <storrelse; j++) {
            for (int i = 1; i <= storrelse; i++) {
                  if (i == storrelse-j) {
      System.out.print("*");
                  } else {
      System.out.print(" ");
  }
}
            if (j>0) {
                  for (int i=0;i<j-1;i++) {
        System.out.print(" ");
      }
      System.out.print("*");
    }
    System.out.println();
}
        for (int j = storrelse-2; j >=0; j--) {
              for (int i = 1; i <= storrelse; i++) {
                    if (i == storrelse-j) {
        System.out.print("*");
                    } else {
        System.out.print(" ");
    }
}
              if (j>0) {
                    for (int i=0;i<j-1;i++) {
       System.out.print(" ");
    }
    System.out.print("*");
}
System.out.println();
```

```
    }

  }
```

Maks poeng: 6

3 Kjøretid, rekursiv

Hva er kjøretida til metoden "finnISortert" i O-notasjon? (6%)

```
    public static int finnISortert(int[] tall, int verdi) {
        return finnISortert(tall, verdi, 0, tall.length);
}


    private static int finnISortert(int[] tall, int verdi, int start, int slutt) {
        if (slutt-start <= 2) {
            if (tall[start] == verdi) return start;
            if (tall[slutt] == verdi) return slutt;
            return -1;
  } else {
            int midten = (start + slutt)/2;
            if (tall[midten] > verdi) {
                return finnISortert(tall, verdi, start, midten);
    } else {
                return finnISortert(tall, verdi, midten, slutt);
      }
    }
  }
```

Velg ett alternativ

- O(1)
- O(log(n))
- O(n)
- O(n*log(n))
- O(n^2)
- O(n^3)
- O(2^n)

Maks poeng: 6

4 Operasjoner for lister

Hvilke listeoperasjoner er raskest (i O-notasjon) på hvilken type liste? (1% pr. riktig svar)

Finn de som passer sammen

	LinkedList	Like rask på begge	ArrayList
Fjern på slutten	○	○	○
Hent ut neste element i en for each loop	○	○	○
Sett inn på starten	○	○	○
Gi elementet på opppgitt indeks en ny verdi	○	○	○
Sett inn på slutten	○	○	○
Fjern på starten	○	○	○
Hent ut element på oppgitt indeks	○	○	○

Maks poeng: 7

5 Innsetting i en ArrayList

Beskriv algoritmen for å sette inn et element på en valgfri indeks i en ArrayList, med pseudokode eller Java kode. (7%)

Skriv ditt svar her...

```
1
```

Maks poeng: 7

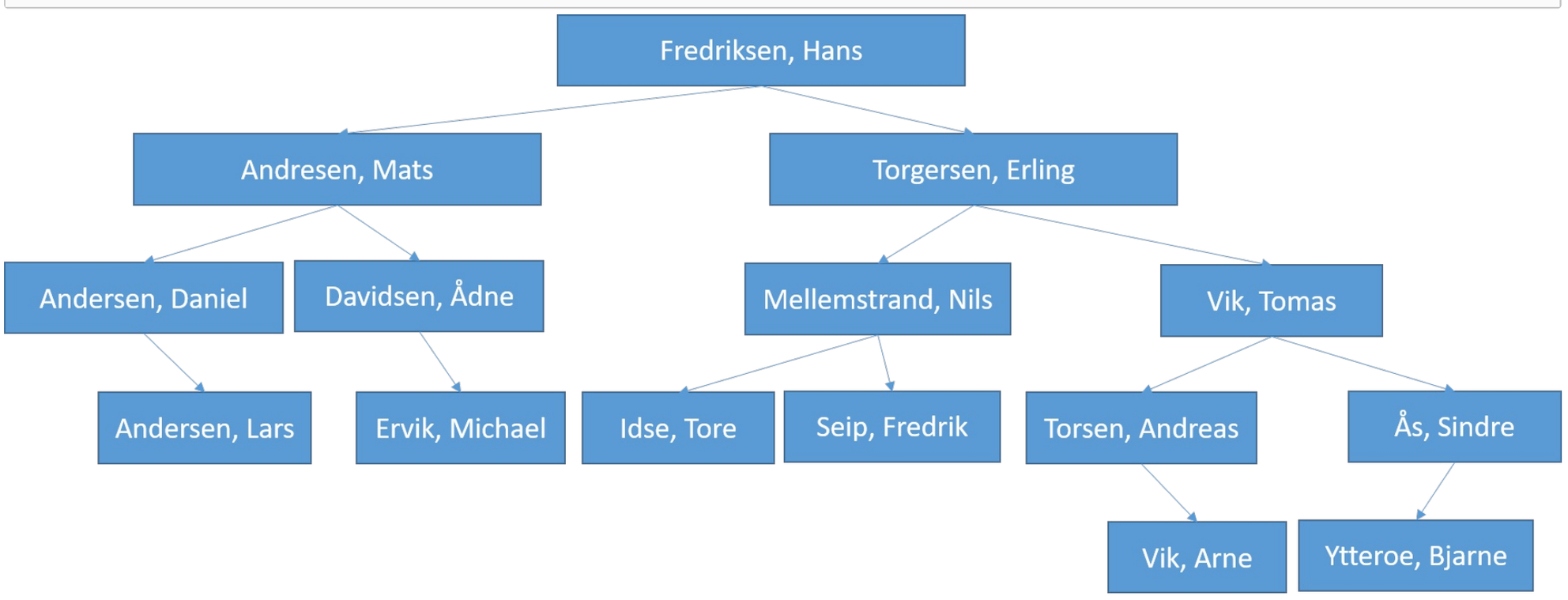
6 Identifiser datastrukturen

Hvilken datastruktur er avbildet på bildet under. Vær så spesifikk som mulig. Begrunn svaret ditt. Inkluder en kort beskrivelse av datastrukturen. (10%)

Skriv ditt svar her...

Format

Words: 0

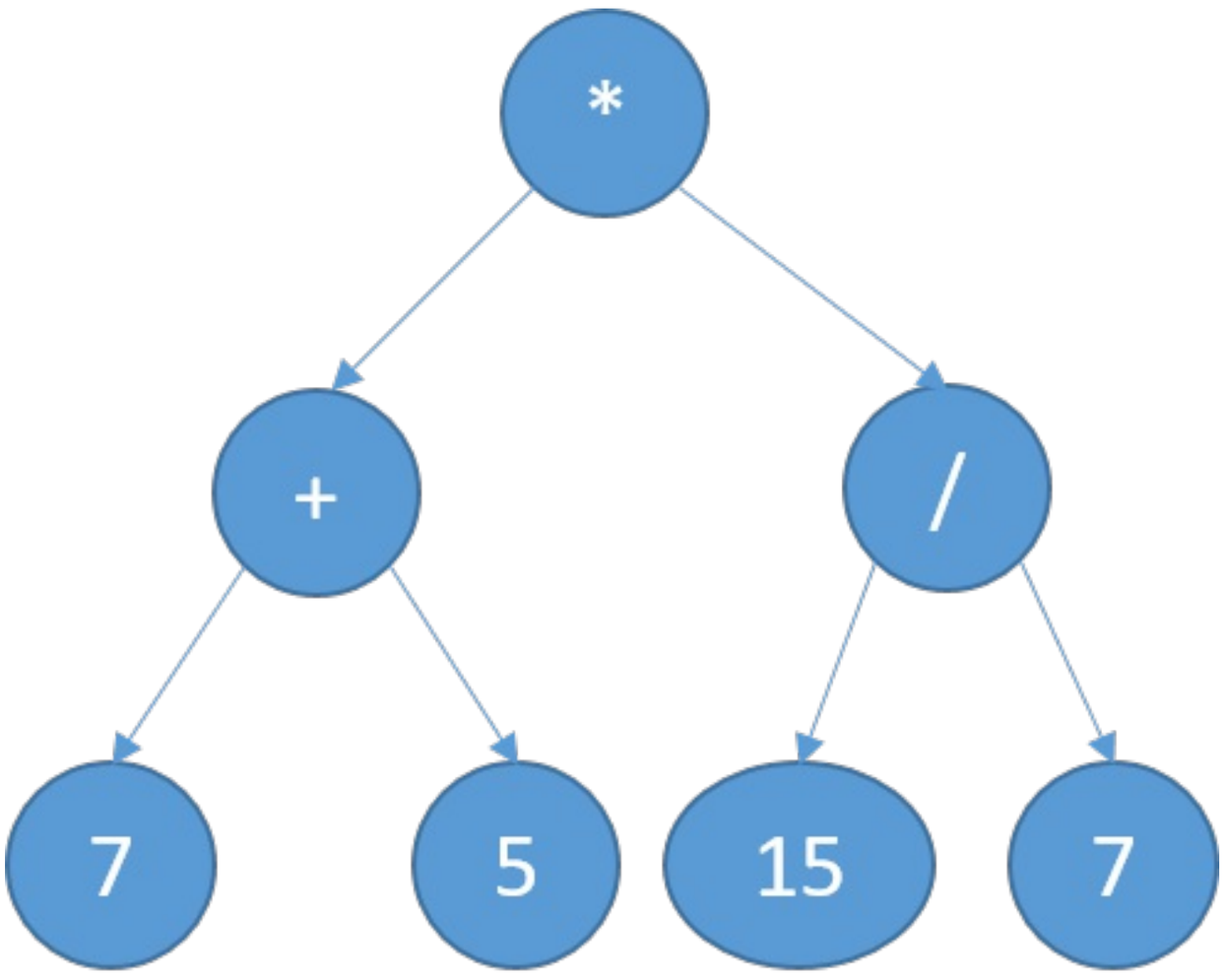


Maks poeng: 10

7 Tre-algoritme

Regnestykker kan representeres som trær. Et eksempel på et slikt tre er oppgitt under. Eksemplet under representerer regnestykket (7 + 5) * (15 / 7).

Skriv en algoritme som regner ut resultatet av et slike regnestykke-tre. Du kan skrive algoritmen i pseudokode eller Java kode. Algoritmen blir evaluert på kjøretid og hvor kort / elegant koden er. (10%)



Skriv ditt svar her...

```
1
```

Maks poeng: 10

8 Haug fra liste

En haug kan representeres som en array. Gitt arrayen under, hvor elementet "dummy" har indeks 0. Tegn opp haugen som denne arrayen representerer på papir. (5%)

Dummy	5	16	24	12	9	55	34	75	62

Maks poeng: 5

9 Lovlig haug

Er haugen representert med lista oppgitt under (og tegnet opp i forrige deloppgave) en lovlig haug? Hvis ikke, begrunn svaret. (5%)

Dummy	5	16	24	12	9	55	34	75	62

Skriv ditt svar her...

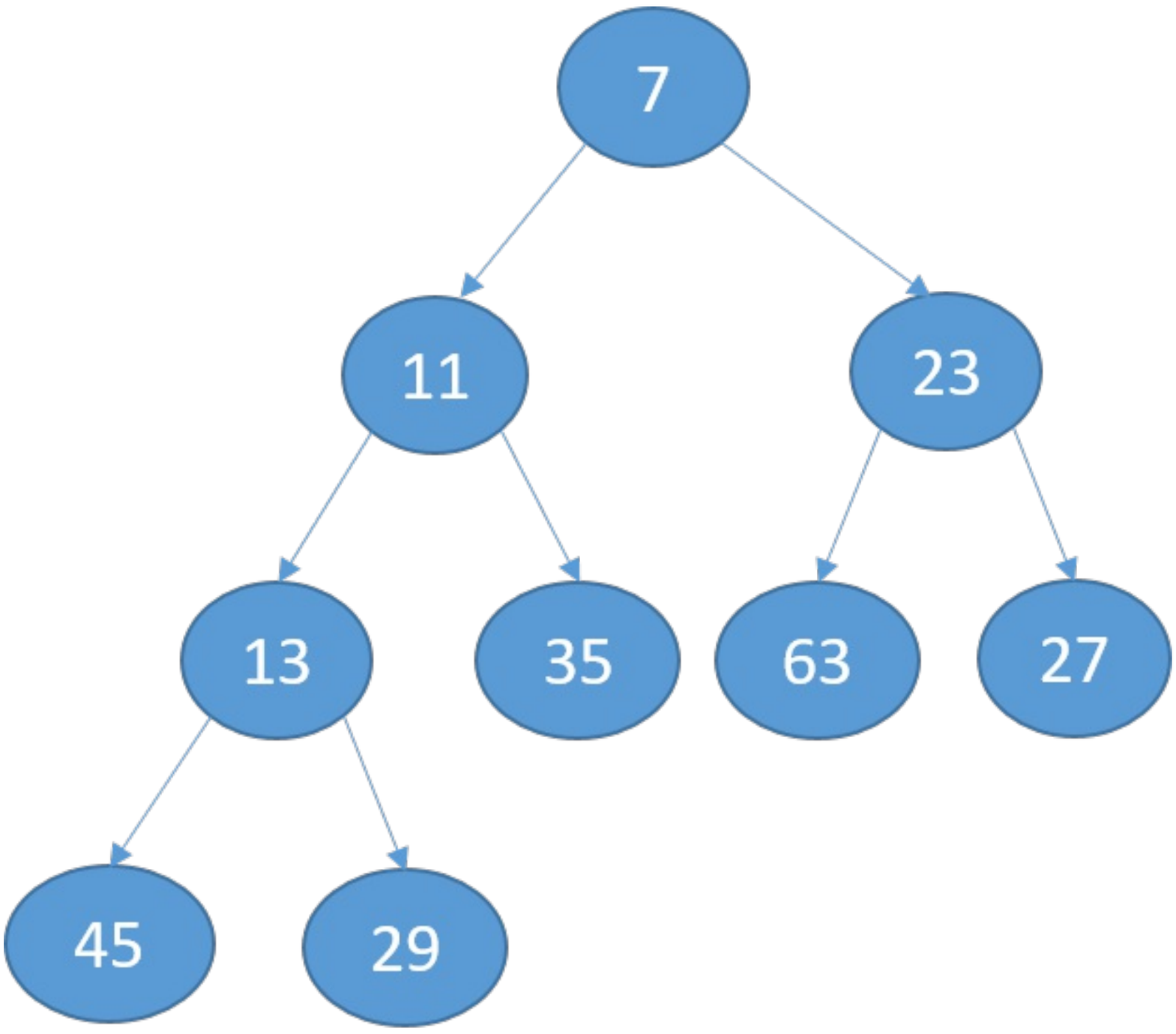
Format

Words: 0

Maks poeng: 5

10 Innsetting i hauger

Gitt haugen under. Vis hvordan algoritmen for innsetting i hauger virker gjennom å sette inn elementet "10" i denne haugen og vise stegene som algoritmen går gjennom på papir. (7%)



Maks poeng: 7

11 Huffman koder

Tegn opp et huffman kodetre for ordet "abrakadabra". (8%)

Maks poeng: 8

12 Ut-graden for en node

Ut-graden for en node i en graf er antall kanter som går ut fra noden. For en node med naboliste representasjon lik den som er oppgitt under, skriv en algoritme for å beregne utgraden til en oppgitt node. Du kan skrive algoritmen i pseudokode eller Java kode. Oppgi også kjøretida i O-notasjon. (5%)

Skriv ditt svar her...

```
1
```

```
public class Naboliste <E> implements KortesteVeiGraf<E> {

    private class Node {

   E element;

   HashMap<Integer, Integer> edges; // HashMap tilNodeIndex -> Vekt

        int kostnad;

        int forrige;


        public Node(E element) {

            this.element = element;

            edges = new HashMap<>();

        }

}


    private ArrayList<Node> nodes;


    public Naboliste() {

        nodes = new ArrayList<>();

  }

}
```

Maks poeng: 5

13 Inn-graden til en node

Inn-graden til en node er antall kanter i en rettet graf som kommer inn til noden. For en node med naboliste representasjon lik den oppgitt under, skriv en algoritme som beregner inn-graden til en oppgitt node. Du kan skrive algoritmen i pseudokode eller Java kode. Oppgi også kjøretida til algoritmen din i O-notasjon. (10%)

Skriv ditt svar her...

```
public class Naboliste <E> implements KortesteVeiGraf<E> {

    private class Node {
      E element;
      HashMap<Integer, Integer> edges; // HashMap tilNodeIndex -> Vekt
            int kostnad;
            int forrige;


            public Node(E element) {
                  this.element = element;
                  edges = new HashMap<>();
            }
}


    private ArrayList<Node> nodes;

        public Naboliste() {
              nodes = new ArrayList<>();
      }
}
```

Maks poeng: 10

14 Bellman Ford

Skriv Bellman Ford algoritmen, enten med Java kode eller pseudokode. (10%)

Skriv ditt svar her...

```
1
```

Maks poeng: 10