



KANDIDAT

381

PRØVE

DATS2300 1 Algoritmer og datastrukturer

Emnekode	DATS2300
Vurderingsform	Skriftlig eksamen under tilsyn
Starttid	22.11.2023 14:00
Sluttid	22.11.2023 16:00

Sensurfrist**11.12.2023 22:59**

PDF opprettet**12.09.2024 09:38**

Informasjon

Oppgave	Tittel	Oppgavetype
i	Eksamensinformasjon	Informasjon eller ressurser

Flervalgs

Oppgave	Tittel	Oppgavetype
1	Kompleksitet	Flervalg
2	Binærtre	Flervalg
3	Tid	Flervalg
4	Huffman-koding	Flervalg
5	Hashfunksjon	Flervalg
6	2-3-4-tre	Flervalg
7	Dijkstra	Flervalg
8	LZW-komprimering	Flervalg

Langsvar

Oppgave	Tittel	Oppgavetype
9	Minimumhaug	Langsvar
10	Filtrer LenketListe	Programmering
11	Innsetting/uttagning av minhaug	Muntlig

1 Komplexitet

Hva er laveste kompleksitet vi kan få for å fjerne et element fra en kø?

Velg ett alternativ:

- ☐ $\mathcal{O}(\log(n))$
- ☐ $\mathcal{O}(n)$
- ☒ $\mathcal{O}(1)$
- ☐ $\mathcal{O}(n^2)$
- ☐ $\mathcal{O}(n \log(n))$

Maks poeng: 4

Knytte håndtegninger til denne oppgaven?

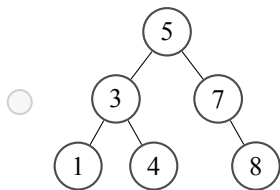
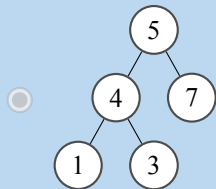
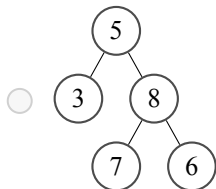
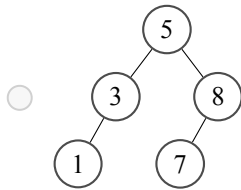
Bruk følgende kode:

4 9 2 3 0 6 0

2 Binærtre

Hvilket av følgende binærtrær er et komplett binærtre?

Velg ett alternativ:



Maks poeng: 8

Knytte håndtegninger til denne oppgaven?

Bruk følgende kode:

2 1 2 5 7 3 4

3 Tid

Et program bruker bubblesortering til å sortere en liste, og kan sortere en liste med 1000 elementer på 10 sekunder.

Vi kjører programmet på nytt på en liste med 2000 elementer, og måler hvor lang tid programmet bruker. Hvor lang tid vil programmet bruke?

Velg ett alternativ:

☐ Omtrent 15 sekunder

☒ Omtrent 40 sekunder

☐ Omtrent 25 sekunder

☐ Omtrent 20 sekunder

☐ Omtrent 70 sekunder

Maks poeng: 8

Knytte håndtegninger til denne oppgaven?

Bruk følgende kode:

8 9 3 9 2 7 8

4 Huffman-koding

Gitt følgende frekvenstabell, lag det kanoniske venstreorienterte Huffmantreet, og komprimer følgende melding: "KJØTTKAKE". Hva blir den kodede meldingen?

Frekvenstabell

A	90
E	152
J	57
K	350
T	179
Ø	34

Velg ett alternativ:

- ☐ 0111 1111 0100 1000 1100 101
- ☐ 1001 1001 0011 0111 0001 010
- ☐ 1010 0010 1000 0001 0111 001
- ☒ 1000 0000 1011 0111 0011 010

Maks poeng: 15

Knytte håndtegninger til denne oppgaven?
Bruk følgende kode:

1 7 0 6 4 9 8

5 Hashfunksjon

Hva er hovedpoenget med en hashtabell?

Velg ett alternativ:

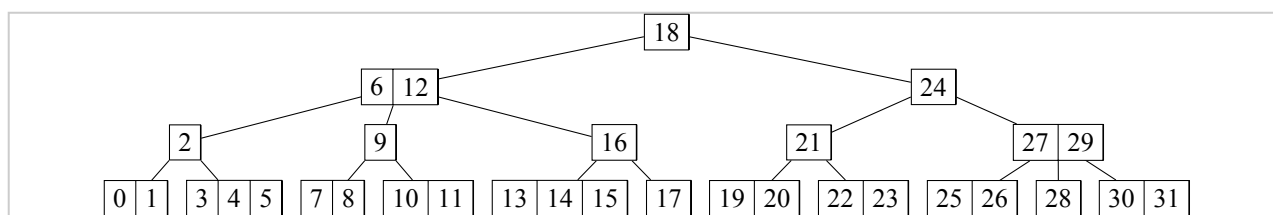
- ☐ Lavere minnebruk
- ☐ Kryptere informasjon
- ☒ Rask innsetting og henting av verdier
- ☐ Sortere data raskere

Maks poeng: 6

Knytte håndtegninger til denne oppgaven?

Bruk følgende kode:

8 4 7 8 7 3 6

6 2-3-4-tre

Over er det tegnet opp et 2-3-4-tre. Om det gjøres om til et rødsort-tre, hvor mange røde noder vil det være i treet?

Velg ett alternativ:

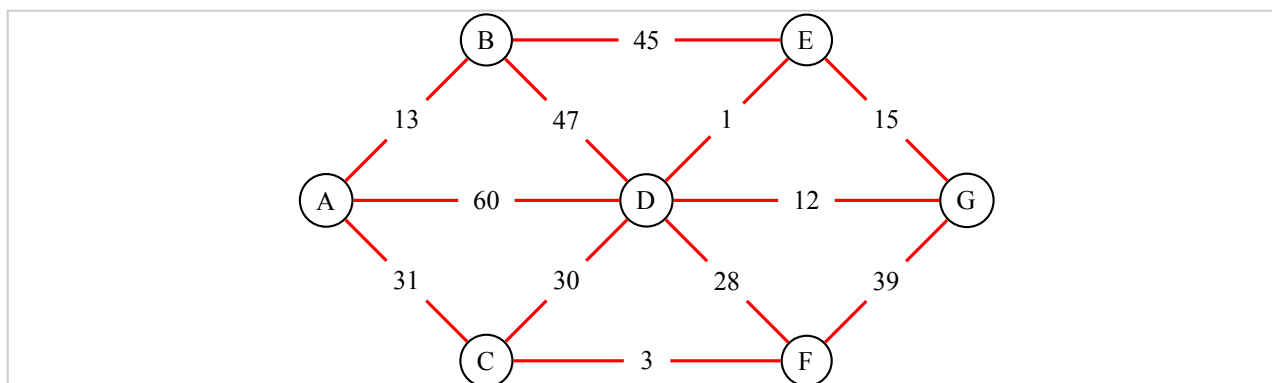
- ☐ 4
- ☐ 21
- ☐ 7
- ☒ 13

Maks poeng: 15

Knytte håndtegninger til denne oppgaven?
Bruk følgende kode:

2084237

7 Dijkstra



I Dijkstras algoritme besøkes nodene i en bestemt rekkefølge for å finne korteste vei mellom A og G. Hvilken rekkefølge besøker algoritmen nodene i?

Velg ett alternativ:

☒ ABCFEDG

☐ ACBDFEG

☐ ABDCEFG

☐ ABCEFDG

Maks poeng: 15

Knytte håndtegninger til denne oppgaven?

Bruk følgende kode:

4 5 2 4 9 5 7

8 LZW-komprimering

Vi skal LZW-komprimere ordet BARBARER, og starter med ordboka med følgende verdier:

A	0
B	1
E	2
R	3

Etter komprimering, hva er den høyeste verdien vi har i ordboka (verdien trenger ikke være brukt i komprimeringa)?

Velg ett alternativ:

☒ 9

☐ 5

☐ 11

☐ 4

Maks poeng: 15

Knytte håndtegninger til denne oppgaven?

Bruk følgende kode:

9 8 3 4 1 0 9

9 Minimumhaug

Forklar kort hvorfor det i gjennomsnitt tar konstant tid å legge inn et element i en minimumsheap. Gi en enkel forklaring som viser din forståelse av hvorfor denne operasjonen har konstant tidskompleksitet, uten å gå inn i tekniske detaljer. En til to setninger er nok.

Skriv ditt svar her

De nederste 2-3 nivåene vil ha ca. 50-80% av alle elementene. Et nytt element hører sannsynligvis hjemme nær bunnen av minimumsheapen og i snitt klarer vi oss med 2-3 sammenligninger.

Ord: 30

Maks poeng: 10

Knytte håndtegninger til denne oppgaven?

Bruk følgende kode:

0 4 0 9 8 1 3

10 Filtrer LenketListe

Denne oppgaven tar utgangspunkt i den minimale LenketListe-koden som er vedlagt.

Du skal lage metoden

```
public LenketListe<T> filtrer(Test<T> p)
```

som gir ut en ny lenket liste som inneholder alle verdiene i den originale lenkede listen som tilfredsstiller `p.test(T verdi)`, i samme rekkefølge som den originale lenkede listen. Den originale lenkede listen skal ikke endres på.

Eksempelvis, om din lenkede liste "lenketListe" består av alle tall fra 1 til 100 i stigende rekkefølge, og testen `p` tester om et tall er mindre enn 50, så skal koden `LenketListe<Integer> nyLenketListe = lenketListe.filtrer(p)` gi deg en ny lenket liste med tallene fra 1 til 49 i stigende rekkefølge. Den originale lenkede listen skal fremdeles da bestå av tall fra 1 til 100, i stigende rekkefølge.

Du kan ikke endre eksisterende kode, men kan lage nye hjelpemetoder om du ønsker.

Prøv å gjør koden så effektiv som mulig, både i tidsbruk og minnebruk. Skriv kommentarer i koden som forklarer hvordan den fungerer, og begrunn eventuelle hjelpestrukturer og hjelpemetoder du har laget.

```
1 public LenketListe<T> filtrer(Test<T> p) {
2     /* Dette er koden dere skal skrive */
3     /* Oppretter en subliste */
4     public List<T> subliste (int fra, int til) {
5         fratilkontroll(fra, til);
6
7         int antallelementer = til - fra;
8         if (antallelementer == 0) {
9             return new LenketListe<Integer>();
10            /* returnerer en tom liste for et tomt intervall */
11        }
12        LenketListe<Integer>() nyLenketListe = lenketListe.filtrer(p);
13        Node<T> current = finnNode(fra);
14        for (int i=0; i<antallelementer; ++i) {
15            subliste.leggin(v)
16        }
17    }
18
19
20 }
```

Maks poeng: 30

Knytte håndtegninger til denne oppgaven?

Bruk følgende kode:

5 5 7 8 2 3 3

11 Innsetting/uttagning av minhaug

Vi skal lage en minimumsheap ved å sette inn heltall sortert ved naturlig ordning.

Tegn hvordan treet ser ut etter å ha satt inn tallene 7, 23, og 19, i denne rekkefølgen.

Tegn hvordan det ser ut etter at du har lagt inn først 17 og så 2 i det samme treet.

Tegn hvordan treet ser ut etter at du har lagt inn først 3 og så 13 i det samme treet.

Forklar hvordan du legger inn tallet 5 i det samme treet, og tegn hvordan treet ser ut etter denne innleggingen.

Fjern minste element fra treet, og tegn treet etter denne fjerninga. Forklar hvordan du fjernet elementet.

Maks poeng: 30

Knytte håndtegninger til denne oppgaven?

Bruk følgende kode:

5 3 9 1 9 2 8

OSLOMET

i Fill out Question Code and Test Information on every sheet. Fyll inn oppgavekode og emneinformasjon på alle skisseark.

Total number of pages

Totalt antall ark

(Only on the first sheet/Kun på første side)

Question Code

Oppgavekode

Date

Dato

Subject code

Emnekode

Candidate ID

KandidatID

Question nr

Oppgavenr

Page number

Sidetall

5	3	9	1	9	2	8
0	0	0	0	0	0	0
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3	3	3
4	4	4	4	4	4	4
5	5	5	5	5	5	5
6	6	6	6	6	6	6
7	7	7	7	7	7	7
8	8	8	8	8	8	8
9	9	9	9	9	9	9

22.11.23

DATS2300

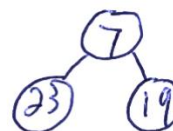
381

11

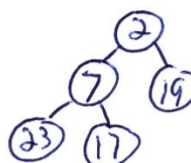
1

Drawing area Tegneområde

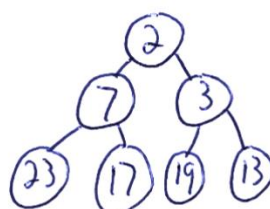
+ 7, 23 og 19 =



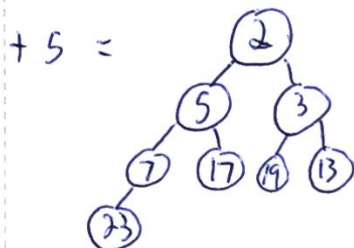
+ 17 og 2 =



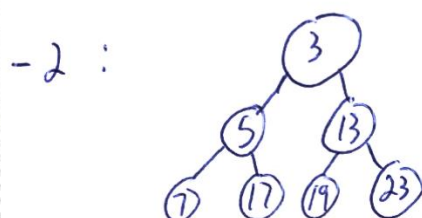
+ 3 og 13 =



5 blir last nederst til venstre og får 23 som forelder node. Deretter bytter den plass med forelder helt til forelder er mindre enn 5.



Fjerner 2 og setter 23 i rot-noden. 23 bytter plass med sitt minste barn helt til barnet er større.



Use only black or blue pen / Benytt kun svart eller blå penn