UNIVERSITETET I BERGEN

KANDIDAT

330

PRØVE

INF102 0 Algoritmer, datastrukturer og programmering

Emnekode	INF102
Vurderingsform	Skriftlig eksamen
Starttid	03.12.2021 08:00
Sluttid	03.12.2021 11:00
Sensurfrist	--
PDF opprettet	02.10.2022 11:23

Informasjon

Oppgave	Tittel	Oppgavetype
ℹ	Declaration	Informasjon eller ressurser
ℹ	Kontaktinfo under eksamen, INF102 H21	Informasjon eller ressurser
ℹ	Generell info om denne eksamen, INF102 H21	Informasjon eller ressurser

Avkryssning

Oppgave	Tittel	Oppgavetype
1	Kjøretid	Flervalg
2	Kjøretid products	Flervalg
3	Loop de loop	Flervalg
4	Chopping firewood	Flervalg
5	Santa aid	Flervalg
6	Shortest path	Paring

Text svar og programmering

Oppgave	Tittel	Oppgavetype
ℹ	Info til seksjon 3	Informasjon eller ressurser
7	Save Rudolph	Tekstfelt
8	Count products	Programmering

Poeng fra semesteroppgavene

Oppgave	Tittel	Oppgavetype
9	Poeng fra Semesteroppgavene H21	Tekstfelt

1 Kjøretid

Se på denne java koden og analyser kjøretiden som funksjon av **n**.
Du skal velge det svaret som best beskriver kjøretiden.

```
public static int numPairs(int n) {
    int count = 0;
    for(int i=0; i<n; i++) {
        for(int j=i+1; j<n; j++) {
            count++;
        }
    }
    return count;
}
```

Velg ett alternativ:

- $O(\sqrt{n})$
- $O(n^3)$
- $O(n\sqrt{n})$
- O(n log n)
- $O(n^2)$
- O(log n)
- O(n)
- O(1)

2 Kjøretid products

Se på denne java koden og analyser kjøretiden som funksjon av **n**.
Du skal velge det svaret som best beskriver kjøretiden.

```
public static void printProducts(int n) {
    int a=0;
    int b=n;

    while(a<=b) {
        if(a*b==n) {
            System.out.println(a+"*"+b+"="+a*b);
            a++;
            b--;
        }
        else if(a*b>n)
            b--;
        else if(a*b<n)
            a++;
    }
}
```

Velg ett alternativ:

- O(1)
- $O(n\sqrt{n})$
- $O(\sqrt{n})$
- O(log n)
- $O(n^3)$
- O(n)
- O(n log n)
- $O(n^2)$

Etter at du har studert koden en stund ser du at koden kan forbedres litt, under ser du den forbedrede koden. Hva er kjøretiden til den forbedrede koden under?

```
public static void printProducts2(int n) {
    int a=1;
    int b=n;

    while(a<=b) {
        if(a*b==n) {
            System.out.println(a+"*"+b+"="+a*b);
            a++;
            b--;
        }
        else {
            if(a*b>n)
                b=n/a;
            else if(a*b<n)
                a++;
        }
    }
}
```

Velg ett alternativ

- O(log n)
- O(1)
- O(n log n)
- $O(n\sqrt{n})$
- $O(\sqrt{n})$
- O(n)
- $O(n^2)$
- $O(n^3)$

3 Loop de loop

Sjå på denne java koden og analyser køyretida som funksjon av **n**.
Du skal velja det svaret som best beskriv køyretida.

```
public static String loopDeLoop(int n) {
    String lyrics = "";
    for(int i=0; i<n; i++) {
        lyrics += "Loop de loop, flip flop.\n";
        lyrics += "Flying in an aeroplane.\n";
    }
    return lyrics;
}
```

Velg ett alternativ:

- O(n log n)
- O(log n)
- O(n)
- O(1)
- $O(n^3)$
- $O(n^2)$
- $O(\sqrt{n})$
- $O(n\sqrt{n})$

4 Chopping firewood

Strømprisen er høy og det er kald ute, som student merker du dette ekstra godt og har bestemt deg for å hugge ved.
Du har fått en hel haug med kubber som må kløyves og har bestemt deg for følgende strategi:
Finn største kubbe, del den i to og legg de to delene tilbake i haugen, igjen finn største kubbe og del den i to osv. helt til du er ferdig.
Får å gjøre dette effektivt skal du organisere vedkubbene i en datastruktur.
Hvilken datastruktur passer best?

Din datastruktur må kunne utføre følgende metoder effektivt:

- Legg til en vedkubbe til samlingen
- Hent ut og fjern den største vedkubben fra samlingen

Anta at de to operasjonene alle skal utføres O(n) ganger hver, velg den datastrukturen som totalt sett vil gi best kjøretid.
Velg ett alternativ:

- ArrayList
- Union-Find
- TreeSet (binært søketre)
- HashSet (hash tabell)
- Uvektet graf
- LinkedList (lenket liste)
- Heap

5 Santa aid

Du skal dele ut julegaver til alle professorene ved UiB og trenger et program som kan hjelpe deg.
Gavene samles inn etter hvert og professorene kommer innom og henter gavene en etter en.
Du tenker at hvis du gir de beste gavene til de professorene som du har hatt dette semesteret så får du bedre karakter, mens de professorene du ikke har hatt kan få de dårligste gavene.

Din datastruktur må kunne utføre følgende metoder effektivt:

- Legg til en ny gave til samlingen.
- Hent ut og fjern den beste gaven fra samlingen.
- Hent ut og fjern den dårligste gaven fra samlingen.

Anta at de tre operasjonene alle skal utføres O(n) ganger, velg den datastrukturen som totalt sett vil gi best kjøretid.
Velg ett alternativ:

- Union-Find
- HashSet (hash tabell)
- Uvektet graf
- TreeSet (binært søketre)
- ArrayList
- LinkedList (lenket liste)
- Heap

6 Shortest path

Velg den algoritmen du ville brukt for å finne korteste sti i de følgende situasjoner:

Finn de som passer sammen:

	BellMan Ford	Brute force med eksponentiell kjøretid	Typological order+dynamic programming	DFS	BFS	Dijkstra
Directed weighted graph with positive weights	○	○	○	○	○	◉
Graph	○	○	○	○	◉	○
Weighted graph with possibly negative cycles	○	◉	○	○	○	○
Directed Acyclic graph	○	○	◉	○	○	○
Weighted Graph with negative weights but no negative cycle	◉	○	○	○	○	○
Weighted graph with positive weights	○	○	○	○	○	◉

7 Save Rudolph

En gruppe samer i nordlige russland trenger å lede reinsdyrene sine trygt over isen. På grunn av klimaendringer er isen ekstra tynn i år.
De har modelert isen som en graf, og for hver kant har de målt tykkelsen på isen.

Du skal beskrive en algoritme som finner en sti fra en start node **a** til en sluttnode **b** slik at reinsdyrene er tryggest mulig. (På en sti er det er kanten med den tynneste isen som bestemmer hvor trygg stien er.)

Skriv ditt svar her

For å finne den tryggeste stien mulig så hadde jeg valgt å bruke prims algoritme til og finne en omvent minimum spanning tree. Måten jeg hadde gjort dette på er at når jeg bruker prim så sorterer jeg priorityQueuen omvent slik at vi får kanten med størt vekt først i stede for å få den minste kanten først. Da vil vi finne veien som er tryggest mulig og gå over.

8 Count products

Skriv et program som gitt en liste **a** av lengde **n** finner alle tripler **i**,**j**,**k** med **i** <= **j** slik at:
a[**i**] * **a**[**j**] = **a**[**k**]

Metodesignaturen kan f.eks. se slik ut:

```
public static int countProducts(int[] a) {
    int total = 0;

    return total;
}
```

Skriv ditt svar her

```
1   //O(n * log n * n) = O(n^2 log n)
2   public static int countProducts(int[] a) {
3       int total = 0;
4       int n = a.length;
5       Arrays.sort(a); //O(n log n)
6       for (int i = 0; i < n; i++) { //n iterations
7           total += countTotal(a, i); //O(n log n)
8       }
9
10      return total;
11  }
12
13  //O(n log n)
14  private static int countTotal(int[] a, int i) {
15      int total = 0;
16      //Make sure that the i<=j criteria is met
17      for (int j = i; j < a.length; j++) { //log n iterations
18          for (int k = 0; k < a.length; k++) { //n iterations
19              if (a[i] * a[j] == a[k]) //O(1)
20                  total++;
21          }
22      }
23      return total;
24  }
```

9 Poeng fra Semesteroppgavene H21

Denne oppgaven skal du ikke gjøre noe på. Her får du poeng du for det du har gjort på semesteroppgavene.

Du er nå ferdig med eksamen :-)

God Jul
hilsen Martin

Skriv en hyggelig melding til foreleser ;-)

en hyggelig melding til foreleser ;-)