

Institutt for datateknologi og informatikk

## Eksamensoppgave i Algoritmer og datastrukturer, IDATT2101

**Faglig kontakt under eksamen: Helge Hafting**

**Tlf.: 924 386 56**

**Eksamensdato: 20. desember 2021**

**Eksamenstid (fra-til): 09:00–14:00**

**Hjelpemiddelkode/Tillatte hjelpemidler: A / Alle hjelpemidler tillatt.**

**Annen informasjon:**

**Målform/språk: bokmål**

**Antall sider (uten forside): 4**

**Antall sider vedlegg: 0**

### Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig	<input type="checkbox"/>	2-sidig	<input checked="" type="checkbox"/>
sort/hvit	<input type="checkbox"/>	farger	<input checked="" type="checkbox"/>
Flervalgskjema?			<input type="checkbox"/>

**Kontrollert av**

.....  
Dato Sign

---

Merk! Studentene finner sensur i Studentweb. Har du spørsmål om sensuren må du kontakte instituttet ditt. Eksamenskontoret vil ikke kunne svare på slike spørsmål.

I noen oppgaver må du bruke *siste siffer i kandidatnummeret ditt* i oppgaven. (Rett siffer burde være lett å finne. F.eks. er «7» siste siffer i «100247».) Hensikten er at ulike studenter skal få litt ulike oppgaver. Hvis *feil* siffer likevel brukes, underkjennes deloppgaven, og det gir mistanke om juks/plagiat.

## Oppgave 1

**24%**

Analyser disse programmene. Bruk  $\Theta$  om mulig. Om ikke, bruk  $O$  og  $\Omega$ . Alle parametre er positive.

(Om det er vanskelig å skrive « $\Theta$ » på ditt tastatur, kan du skrive «Theta» i stedet.)

```
int prog_a(int a, int b, int c) {
    int sum = 0;
    for (int i=0; i<a ; ++i) {
        for (int j=0; j<c; ++j) {
            sum += i*j;
            if (sum > b) return sum - b;
        }
    }
    return sum;
}
```

---

```
int prog_b(int q, int p, int r) {
    int sum = 0;
    for (int i=0; i<q ; ++i) {
        for (int j=0; j<p; ++j) {
            sum += i*j;
            if (sum > r) sum = sum % r;
        }
    }
    for (int k=r; k>0; --k) sum += k*k;
    return sum;
}
```

```

int prog_c(int q, int n, int [] tab) {
    int sum = 0;
    if (q > 0) {
        sum += prog_c(q/2, n/4, tab);
        for (int i=0; i<q; ++i) sum += tab[i*n+q];
        sum += prog_c(q/2, n/4, tab);
    }
    return sum;
}

```

---

```

double prog_d(int n, float x) {
    if (n == 0) return 1.0;
    else return x * prog_d(n-1, x);
}

```

---

```

int prog_e(int a, int b, int c) {
    int sum = 0;
    if (a >= b) return -1;
    for (int i = a; i < b; i += c) sum += i*i;
    return sum;
}

```

---

```

int prog_f(int n) {
    if (n > 0) {
        int produkt = 1;
        for (int i = 1; i <= 6; ++i) {
            produkt *= prog_f(n/3);
        }
        return produkt;
    } else return 1;
}

```

## Oppgave 2

6%

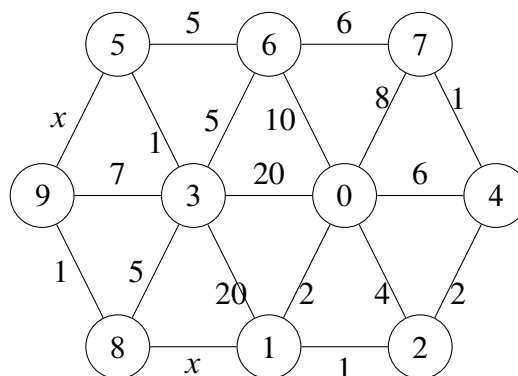
La  $x$  være siste siffer i kandidatnummeret ditt.

- Sett disse tallene (9, 1,  $x$ , 8, 2, 5, 7, 6) inn i en min-heap. Tegn opp hvordan heapen ser ut etter hvert tall.
- Sett de samme tallene inn i ett binært søketre. Tegn opp hvordan treet ser ut etter hvert tall.

### Oppgave 3

25%

La  $x$  være siste siffer i kandidatnummeret ditt. To kanter i grafen har denne vekta.



- Finn korteste vei fra node nr.  $x$  til alle andre noder. Tegn korteste-vei treet, og skriv opp avstanden til hver node.
- Finn og tegn et minimalt spenntre for grafen. Skriv opp den totale vekta for spenntreet.
- Finn og tegn et annet minimalt spenntre for grafen, eller forklar hvorfor det ikke er mulig.
- Dijkstras algoritme og Bellman-Ford algoritmen løser samme problem, men med ulik kompleksitet. Grei ut om når vi bruker den ene algoritmen, og når vi bruker den andre.
- I analysen av Prims algoritme, står det at  $O((N + K) \log N)$  kan forenkles til  $O(K \log N)$ . Forklar hvorfor dette stemmer.

### Oppgave 4

25%

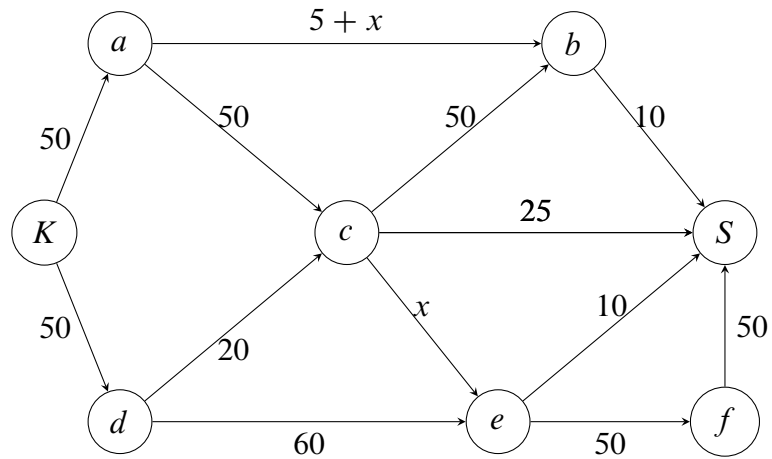
- Fortell om sterke og svake sider ved quicksort og innsettingssortering. Gi eksempel på når du vil bruke den ene sorteringa, og når du vil bruke den andre.
- Når vi implementerer quicksort, er det lett å gjøre feil som gir unødvendig dårlig ytelse. Fortell om slike problemer, og hvordan vi unngår dem.
- Kan vi bruke dynamisk programmering for å lage en enkel og effektiv sorteringsalgoritme?  
Hvis ja, foreslå en slik algoritme. Hvis nei, forklar hvorfor dette er vanskelig.
- Hva vil det si, at et problem er i kompleksitetsklassen **NP**? Gi også eksempel på et problem som er i **NP**, og fortell hva problemet går ut på.
- Pristilsynet vil ha oversikt over 10 000 dagligvarer. De vil ha varane i en hashtabell, så de kan slå opp via pris. De vil for eksempel finne «alle varer som koster 200 kr» for å sjekke om butikkene følger regelverket. Tilsvarende for alle andre priser. Mange varer har priser som ender i enten 0 eller 9.

Foreslå tabellstørrelse, hashfunksjon og kollisjonshåndtering så dette blir effektivt. (Du trenger ikke finne konkrete primtall/toerpotenser. Om du f.eks. trenger et primtall over 1000, skriv «la  $p$  være neste primtal høyere enn 1000.» Så kan du bruke  $p$  i svaret ditt.)

## Oppgave 5

20%

La  $x$  være siste siffer i kandidatnummeret ditt. Legg merke til at  $x$  brukes i noen av kantvektene i grafen:



- Finn maksimal flyt fra  $K$  til  $S$ . Nytt flytøkende veier, og skriv opp hver vei og hvor mye flyt du legger til langs veien.
- Forklar hva fenomenet «flytkansellering» går ut på, når vi jobber med maksimum flyt.
- Tegn et binærtre med fire noder. Det skal være slik at enten vi skriver ut nodene i postordenrekkefølge eller in-orden, så får vi samme rekkefølge.