

Institutt for datateknologi og informatikk

Eksamensoppgave i Algoritmer og datastrukturer, IDATT2101

Faglig kontakt under eksamen: Helge Hafting

Tlf.: 924 386 56

Eksamensdato: 6. desember 2022

Eksamenstid (fra-til): 09:00–13:00

Tillatte hjelpemiddel: ett A4-ark med notater

Annen informasjon:

Målform/språk: bokmål

Antall sider (uten forside): 3

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

Oppgave 1

20%

Analysér disse programmene. Bruk Θ om mulig. Om ikke, bruk O og Ω . Alle parametre er positive.

```
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) {
            for (int k=b; k>0; --k) {
                sum += i*j-k;
            }
        }
    }
    return sum;
}
```

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

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

```
int prog_e(int a, int b, int c) {  
  
    int sum = 1;  
    for (int i = 1; i < a; ++i) sum *= i;  
    for (int j = 1; j < b; ++j) sum += j;  
    return sum;  
  
}
```

Oppgave 2

5%

- Sett disse tallene (5, 7, 1, 0, 8, 6, 2) inn i en max-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.

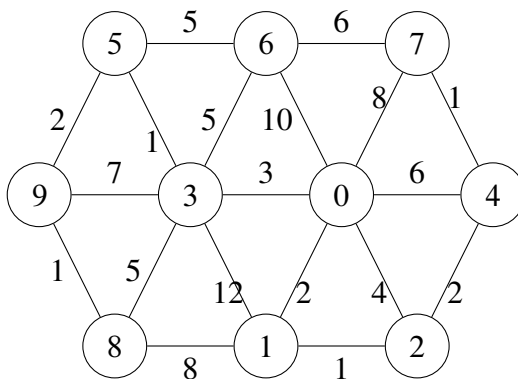
Oppgave 3

10%

- Fortell kort om Burrows-Wheeler transformasjonen. Hva er det den oppnår, som er nyttig for datakompresjon?
- Jeg har brukt Burrows-Wheeler transformasjonen på et ord, og fått strengen *OHHO. «*» er slutt-tegnet, som kommer etter alle andre tegn i sorteringsrekkefølgen. Finn ut hva det opprinnelige ordet var, og vis hvordan du gjorde det.

Oppgave 4

20%



- Finn korteste vei fra node nr. 9 til alle andre noder. Tegn korteste-vei treet, og skriv opp avstanden til hver node.
- Finn og tegn et minimalt spennetre for grafen. Skriv opp den totale vekta for spennetre.
- Finn og tegn et annet minimalt spennetre for grafen, eller forklar hvorfor det ikke er mulig.

Oppgave 5

25%

- a) Grafen G har 8 noder. Hvis vi sorterer den med topologisk sortering, får vi enten rekkefølgen 1, 3, 0, 7, 2, 5, 4, 6 eller rekkefølgen 1, 3, 0, 7, 5, 2, 4, 6.

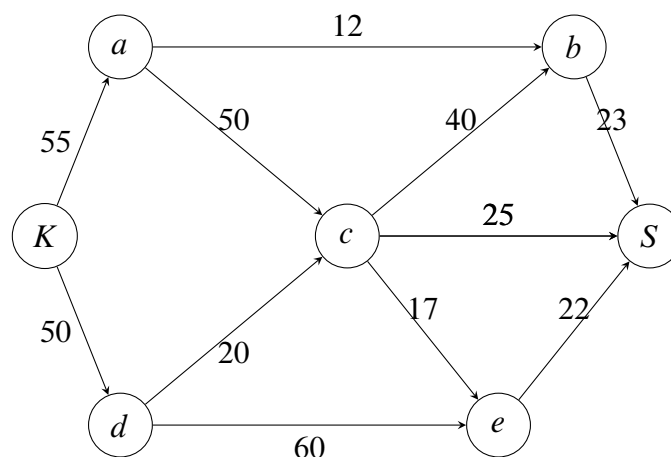
Hvor mange sterkt sammenhengende komponenter er det i grafen G ?

- b) Heapsort og innsettingssortering er ikke de mest populære sorteringsalgoritmene. Men når kan det være lurt å bruke disse?
- c) Lag frekvenstabell og huffmannstre med utgangspunkt i ordet «settekasse». Skriv binærkoden for «settekasse» med utgangspunkt i treet du lagde. Hvor mange bits trengte du?
- d) Hva vil det si, at et problem er NP-komplett? Gi også eksempel på et slikt problem, og fortell hva problemet går ut på.

Oppgave 6

20%

Bruk denne grafen i de to første deloppgavene.



- a) Finn maksimal flyt fra K til S . Nytt flytøkende veier, og skriv opp hver vei og hvor mye flyt du legger til langs veien.
- b) Sorter grafen topologisk, eller forklar hvorfor det ikke er mulig.
- c) Tegn et binærtre med tre noder. Det skal være slik at vi får ulike rekkefølger, om vi skriver nodene i post-orden, pre-orden eller in-orden.

Skriv nodene i pre-orden, post-orden, in-orden og nivå-orden.