Universitetet i Bergen
Det matematisk-naturvitenskapelige fakultet
Eksamen i emnet INF 102 - Algoritmer, datastrukturer og programmering
Torsdag 2 desember 2010, kl 09:00 - 12:00
Bokmål

Ingen hjelpemidler er tillatt bortsett fra kalkulator.

Du trenger ikke å skrive java-kode for noen av oppgavene. Det holder med pesudokode eller en beskrivelse av algoritmen. **Alle kjøretider skal gis med O-notasjon og alle svar må begrunnes.** Det er totalt 3 oppgaver.

Oppgave 1 (25%)

I denne oppgaven skal vi se på kjøretid.

Anta at n er et gitt positivt heltall. Hva er kjøretiden i O-notasjon av følgende kodefragmenter? Du skal gi så god kjøretid som mulig. Husk å begrunne dine svar.

a)

```
i = 1;
j = n;
while(i < j) {
   i = i + 1;
   j = j - 1;
}
```

b)

```
i = 1;
while(i < n) {
   j = 1;
   while(j < n) {
      j = j + 1;
   }
   i = i + 1;
}
```

c)

```
i = 1;
while(i < n) {
   j = 1;
   while(j < n) {
      j = j + 1;
   }
   i = i * 2;
}
```

d)

```
i = 1;
while(i < n) {
   j = 1;
   while(j < i) {
      j = j + 1;
   }
   k = 1;
   while(k < i) {
      k = k + 1;
   }
   i = i + 1;
}
```

e)

```
i = 1;
while(i < n) {
   j = 1;
   while(j < i) {
      k = 1;
      while(k < i) {
         k = k + 1;
      }
      j = j + 1;
   }
   i = i + 1;
}
```

Skriv O-notasjonsuttrykk for funksjonene nedenfor. Dine O-notasjonsfunksjoner skal være så forenklet og så saktevoksende som mulig. Husk å begrunne dine svar.

f) $20n^2 + 123n + 1000 \log_2 n$

g) $12n^3 + 5n^2 \log_2 n$

h) $n^4 + 2^n$

i) $n + n \log_2 n + \log_2 n$

j) $2^{10} + n$

k) $2^{10}n$

Vi har kommet frem til følgende rekursive funksjoner som beskriver kjøretiden til to rekursive algoritmer. Hva er kjøretiden i O-notasjon for hver av dem når $T(1) = 1$ i begge tilfeller? Husk å gi utledningen.

l) $T(n) = T(\frac{n}{2}) + 1$

m) $T(n) = T(n-1) + n$

Oppgave 2 (35%)

I denne oppgaven skal vi se på urettede grafer uten vekter på kantene (dvs. hver kant har vekt 1). Anta at vi er gitt en slik graf med n noder og m kanter.

a) Beskriv en raskest mulig algoritme for å finne ut om grafen er sammenhengende. Hvis grafen ikke er sammenhengende skal algoritmen returnere antall sammenhengende komponenter av grafen. Hva er kjøretiden?

b) Anta at grafen er sammenhengende. Beskriv en raskest mulig algoritme for å finne et utspennende tre av grafen. Hva er kjøretiden?

c) En *bro* i en sammenhengende graf er en kant slik at hvis vi fjerner kanten blir grafen ikke-sammenhengende. Forklar hvorfor følgende er sant: hvis grafen har en kant som er bro så må ethvert utspennende tre inneholde denne kanten.

d) Anta at grafen er sammenhengende. Beskriv en raskest mulig algoritme for å finne ut om grafen inneholder en bro. Hva er kjøretiden?

e) Er påstanden i c) også sant når grafen er vektet (med positive heltallige vekter på kantene)? Begrunn ditt svar.

Oppgave 3 (40%)

Vi er gitt en usortert tabell (array) A med n positive heltall der samme tall kan forekomme flere ganger og det ikke er noen begrensninger på størrelsen av tallene.

a) Beskriv en raskest mulig algoritme som finner og returnerer de 5 største elementene i A. Hva er kjøretiden?

b) Anta at vi er gitt et positivt heltall $k \leq \frac{n}{2}$ som en del av inndata i tillegg til A. Beskriv en algoritme med kjøretid $O(nk)$ som finner og returnerer de k største elementene i A.

c) Beskriv en algoritme med kjøretid $O(n \log k)$ for problemet i b). Hvilken datastruktur må du bruke til å lagre de k største elementene? (**Bonus:** Kan du beskrive en enda raskere algoritme?)

d) Beskriv hvordan vi kan bruke partisjoneringsmetoden fra Quicksort til å lage en algoritme for å løse problemet i b). Hva blir kjøretiden dersom pivot-elementet alltid deler A i midten? Hva blir verste kjøretid?

Lykke til!

Pinar Heggernes   Martin Vatshelle