Universitetet i Bergen
Det matematisk-naturvitenskapelige fakultetet
Eksamen i INF102 - Algoritmer, datastrukturer og programmering
Fredag 10. desember, 2004, 09:00 - 14:00
Norsk tekst

Ingen hjelpemidler er tillatt.

Enhver algoritme eller pseudo-kode som du skriver må forklares og forsvares enten med en uformel begrunnelse eller ett formelt bevis.

Hvis du bes om å oppgi kjøretiden til en algoritme, eller å lage en algoritme med en gitt kjøretid, må du begrunne hvorfor algoritmen har denne kjøretiden.

1 Tidskompleksitet og rekursjon

1. Angi tidskompleksiteten (dvs kjøretiden) til hver av de følgende algoritmene med "stor-O" notasjon (dvs $\mathcal{O}(\cdot)$).

```
1) public static void F1(int n)
{
     int sum=0;
     for (int i=0; i<n; i++)
       for (int j=0; j<i; j++)
         sum++;
}

2) public static int F2(int n)
{
     if (n<=1) return 1;
     else return 3*F2(n/3);
}

3) public static int F3(int n)
{
     if (n<=1) return 1;
     else return F3(n-2)+F3(n-2);
}

4) public static int F4(int n)
{
     if (n<=1) return 1;
     else return F4(n-1)+F4(n-2);
}

5) public static void F5(int n)
{
     int sum=0;
```

```
     for (int i=0; i<2*n; i++)
       for (int j=0; j<2*i; j++)
        for (int k=0; k<2*j; k++)
         sum=sum+k;
}
```

```
6)public static int F6(int n)
{
     if (n<=1) return 0;
     else return 2*F6(n/2)+n;
}
```

2. Hvilken output skrives til skjermen av følgende algoritme?

```
public static void main(String[] args)
{
     int i=5;
     System.out.println( F3(F2(i))+F6(i/2) );
}
```

2 Sortering

1. Skriv pseudokode som beskriver en rekursiv inserttingssortering (insertion-sort). Hva er kjøretiden i beste og verste tilfelle? Forklar svaret ditt.

2. Sorter tabellen med heltall 5 6 1 9 8 12 0 2 10 i stigende rekkefølge ved å bruke insertion-sort. Skriv ut innholdet i tabellen hver gang sorteringsalgoritmen endrer den.

3. Sorter tabellen med heltall 5 6 1 9 8 12 0 2 10 i stigende rekkefølge ved å bruke Shell-sort med "indeks-skille" (i læreboken "index separations") 4, 2 og 1. Skriv ut innholdet i tabellen hver gang sorteringsalgoritmen endrer den.

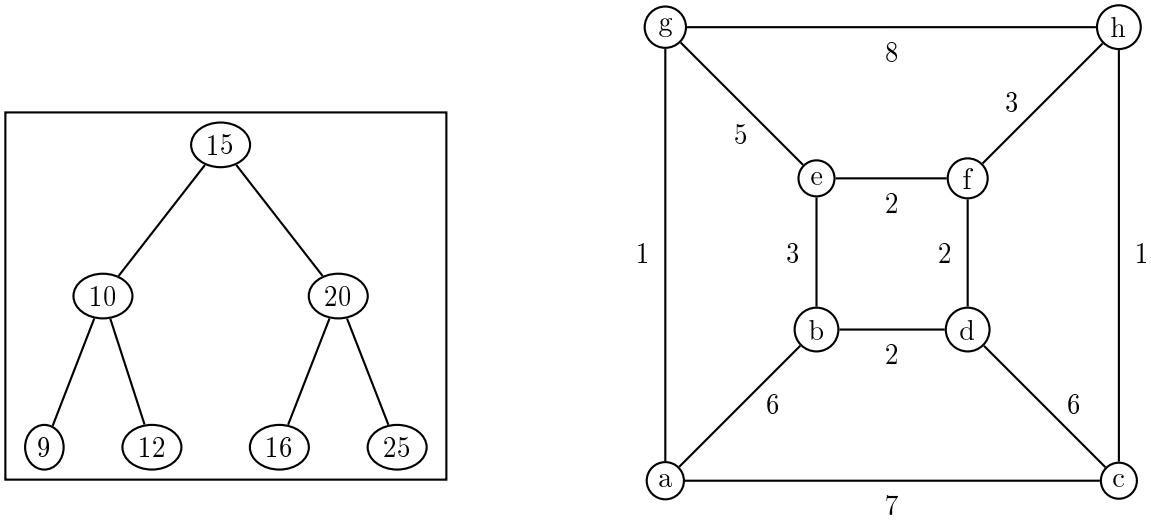
4. Sorter tabellen med heltall 5 6 1 9 8 12 0 2 10 i stigende rekkefølge ved å bruke quick-sort. Bruk "medianen av tre tall" som pivot-element. Skriv ut innholdet i tabellen hver gang sorteringsalgoritmen endrer den.

3 Trær

1. Vi betrakter nå det binære søketreet som er avbildet i venstre halvdel av figur 1. Se nå for deg at du treverserer dette treet og lagrer dets data i en fil. Så leser du denne filen inn igjen og legger søke-nøklene inn i ett nytt (tomt til å begynne med) søketre. Vis resultatet av disse operasjonene hvis det første treet ble traversert på følgende måter **a) preoder b) inorder c) level order d) postorder**.

2. Vis stegene heapsort utfører på følgende tabell 5 12 9 81 4 2 10 11 13 1.

3. Tegn treet som lages hvis du legger til verdiene 9, 20, 69, 48, 48, 60, 70, 75, 1 og 100 til ett **a) AVL tre b) Rød-svart tre**.



Figur 1: Binære søketrær og vektede grafer

4 Grafer

1. Skriv pseudokode for en algoritme som gitt en rettet graf G på n noder og m kanter avgjør i tid $\mathcal{O}(n+m)$ om G har en sykel. Ikke glem å begrunne hvorfor kjøretiden i din algoritme er $\mathcal{O}(n+m)$.

2. Betrakt den vektede grafen avbildet i høyre halvdel av figure 1. Vis stegene som de følgende algoritmene utfører på denne grafen.

   a) Med utgangspunkt i bredde-først søk, finn den korteste stien fra a til h.

   b) Prim's algoritme for å beregne minste utspennende tre.

   c) Kruskal's algoritme for å beregne minste utspennende tre.

5 Design av algoritmer

I sjakk kan dronningen bevege seg så langt som hun måtte ønske enten i horisontal, vertikal eller diagonal retning. Ett sjakkbord har 8 rader og 8 kolonner. I "Dronning-problemet" spør man hvordan 8 dronninger skal plasseres på ett sjakkbord slik at ingen av dronningene kan ta noen av de andre dronningene. Skriv pseudokode til ett program som løser dette problemet.

Fedor Fomin Pinar Heggernes