MAT1030 – Diskret Matematikk

Plenumsregning 2: Ukeoppgaver

Mathias Barra

Matematisk institutt, Universitetet i Oslo

23. januar 2009

(Sist oppdatert: 2009-02-02 14:23)

Plenumsregning 2

Oppgave 1.1

Modifiser algoritmen fra 1.2.1 slik at den også returnerer posisjonen i listen hvor det minste tallet forekommer.

Løsning

1. Input $x_1, \ldots, x_n$ [listen med tall]
2. Input n [antall tall i listen]
3. $\mathrm{min} \leftarrow x_1$
4. $\underline{\mathrm{posisjon} \leftarrow 1}$
5. **For** $i = 2$ **to** n **do**
   - 5.1. **If** $x_i < \mathrm{min}$ **then**
     - 5.1.1. $\mathrm{min} \leftarrow x_i$
     - 5.1.2. $\underline{\mathrm{posisjon} \leftarrow i}$
6. Output $\mathrm{min}, \mathrm{posisjon}$

- Merk at oppgaven ikke er mulig å løse helt som i boken.
- Listen $2, 3, 4, 2, 5$ har ikke noe minste element.
- Det vi får som Output er de(t) minimale elementen(e) og indeksen til ett av dem.

Oppgave 1.3

Skriv en algoritme som tar som input et tall n og som regner ut

$$1^2 + 2^2 + 3^2 + \ldots + n^2.$$

Løsning

1. Input n
2. $sum \leftarrow 0$
3. **For** $i = 1$ **to** n **do**
   3.1. $sum \leftarrow sum + i^2$
4. Output sum

Oppgave 1.6

Skriv en algoritme som tar en liste av tall $[x_1, \ldots, x_n]$ som input og som sjekker om listen er stigende. (Dvs. at $x_i \leqslant x_{i+1}$ for alle $0 \leqslant i < n$).
Tilleggskrav: Algoritmen skal designes slik at sjekkingen stopper med en gang svaret er gitt.

Løsning

1. Input $x_1, \ldots, x_n$ [$n \geqslant 1$]
2. $i \leftarrow 1$
3. $stigende \leftarrow$ true
4. **While** $i < n$ **and** $stigende$ **do** [**While** tar hånd om tilleggskravet]
    4.1. **If** $x_i > x_{i+1}$ **then**
        4.1.1. $stigende \leftarrow$ false
    4.2. $i \leftarrow i + 1$
5. **If** $stigende$ **then**
    5.1. Output 'Tallene er i stigende rekkefølge'
    else
    5.2. Output 'Tallene er ikke i stigende rekkefølge'

Oppgave 1.9

1. Input n $[n \geqslant 0]$
2. $i \leftarrow 0$
3. **While** n er partall **do**
   3.1. $n \leftarrow n/2$
   3.2. $i \leftarrow i + 1$
4. Output i

(a) Hva returnerer algoritmen når 12 er input?

(b) Hva returnerer algoritmen når input er et oddetall?

(c) Hva skjer når 0 er input?

(d) Er denne sekvensen av steg en algoritme?

Oppgave 1.9

1. Input n $[n \geqslant 0]$
2. $i \leftarrow 0$
3. **While** n er partall **do**
   3.1. $n \leftarrow n/2$
   3.2. $i \leftarrow i + 1$
4. Output i

(a) Hva returnerer algoritmen når 12 er input?

Løsning (a)

Steg	n	i
1	12	–
2	12	0
3	12	0
3.1	6	0
3.2	6	1
3	6	1
3.1	3	1
3.2	3	2
3	3	2
4	3	2

Svar (a): 2

Oppgave 1.9

1. Input n $[n \geqslant 0]$
2. $i \leftarrow 0$
3. **While** n er partall **do**
   3.1. $n \leftarrow n/2$
   3.2. $i \leftarrow i + 1$
4. Output i

(b) Hva returnerer algoritmen når input er et oddetall?

Løsning (b)

Steg	n	i
1	oddetall	–
2	oddetall	0
3	oddetall	0
4	oddetall	0

Svar (b): 0

Oppgave 1.9

1. Input n $[n \geqslant 0]$
2. $i \leftarrow 0$
3. **While** n er partall **do**
   3.1. $n \leftarrow n/2$
   3.2. $i \leftarrow i + 1$
4. Output i

(c) Hva skjer når 0 er input?

(d) Er denne sekvensen av steg en algoritme?

Løsning (c og d)

Steg	n	i
1	0	–
2	0	0
3	0	0
3.1	0	0
3.2	0	1
3	0	1
3.1	0	1
3.2	0	2
3	0	2
3.1	0	2
3.2	0	3
	$\vdots$	

Svar (c): Den terminerer ikke.
Svar (d): Nei.

Oppgave 1.10

1. Input n [$n \geqslant 0$]
2. $answer \leftarrow n$
3. **While** $n > 1$ **do**
   3.1. $n \leftarrow n - 1$
   3.2. $answer \leftarrow answer \times n$
4. Output $answer$

(a) Lag en kjøringstabell som viser hva som skjer når 4 er input.

(b) Er denne sekvensen av steg en algoritme? Begrunn svaret.

Løsning (a)

Steg	n	$answer$
1	4	–
2	4	4
3	4	4
3.1	3	4
3.2	3	12
3	3	12
3.1	2	12
3.2	2	24
3	2	24
3.1	1	24
3.2	1	24
4	1	24

Oppgave 1.10

1. Input n $[n \geqslant 0]$
2. $answer \leftarrow n$
3. **While** $n > 1$ **do**
   - 3.1. $n \leftarrow n - 1$
   - 3.2. $answer \leftarrow answer \times n$
4. Output $answer$

(a) Lag en kjøringstabell som viser hva som skjer når 4 er input.

(b) Er denne sekvensen av steg en algoritme? Begrunn svaret.

Løsning (b)

Ja, dette er en algoritme. **Stegene** er utvetydig definert og sekvensen av steg vil alltid terminere. **Terminasjon** følger av at steg 3 utføres (høyst) $n - 1$ ganger.

Oppgave 1.12

1. Input n $[n \geqslant 0]$
2. **For** $i = 1$ **to** n **do**
   2.1. $a_i \leftarrow 0$
3. **For** $i = 1$ **to** n **do**
   3.1. **For** $j = 1$ **to** n **do**
      3.1.1. **If** j kan deles på i **then**
         3.1.1.1. $a_j \leftarrow 1 - a_j$
         [a_j er alltid enten 0 eller 1]
4. **For** $i = 1$ **to** n **do**
   4.1. Output a_i

Løsning (a)

1. gang	1111111111
2. gang	1010101010
3. gang	1000111000
4. gang	1001111100
5. gang	1001011101
6. gang	1001001101
7. gang	1001000101
8. gang	1001000001
9. gang	1001000011
10. gang	1001000010

Oppgave 1.12

1. Input n $[n \geqslant 0]$
2. **For** $i = 1$ **to** n **do**
   2.1. $a_i \leftarrow 0$
3. **For** $i = 1$ **to** n **do**
   3.1. **For** $j = 1$ **to** n **do**
      3.1.1. **If** j kan deles på i **then**
         3.1.1.1. $a_j \leftarrow 1 - a_j$
         [a_j er alltid enten 0 eller 1]
4. **For** $i = 1$ **to** n **do**
   4.1. Output a_i

Løsning (b)

$a_j = 1$ nøyaktig når j er et kvadrattall. Grunnen er at det er kun kvadrattall som har et odde antall divisorer.

Løsning (b) (Kort)

a_j er 0 hvis og bare hvis j har et jevnt antall divisorer der vi teller med både 1 og j selv. Dette følger av at hver gang vi finner en divisor til j 'snur' vi a_j, og dersom vi kommer tilbake til 0 må vi ha 'snudd' a_j et jevnt antall ganger. Siden algoritmen forsøker med alle tall fra 1 til n, og siden $n \geqslant j$ har vi dessuten forsøkt alle mulig kandidater til en divisor.

Løsning (b) (Lang)

Gitt den korte løsningen ser vi at dersom $a_j = 1$, har j et ujevnt antall divisorer. La m være et vilkårlig tall, og la $\prod_k p_k^{e_k}$ være primtalls-dekomposisjonen til m (Husk at denne er unik!). enhver divisor d av m må da ha primtalls-dekomposisjon $\prod_k p_k^{d_k}$ der $0 \leqslant d_k \leqslant e_k$. Samtidig ser vi av dette at enhver kombinasjon av eksponenter som oppfyller dette kravet resulterer i en divisor. Dermed har vi $\prod_k (e_k + 1)$ antall divisorer til m. Det er klart at $\prod_k (e_k + 1)$ er et oddetall hviss hver $e_k + 1$ er et oddetall hviss hver e_k er et partall: dvs på formen $e'_k \cdot 2$. Men så er $\prod_k p_k^{e_k} = \left(\prod_k p_k^{e'_k}\right)^2$, altså et kvadrattall.

Oppgave 2.2

Overfør følgende binære tall til desimaltall ved å først skrive dem på ekspandert form.

(a) 1100101_2

(b) $1010111,1011_2$

Løsning

(a) $1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 =$
$64 + 32 + 4 + 1 = 101$

(b) $1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 +$
$1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} =$
$64 + 16 + 4 + 2 + 1 + \frac{1}{2} + \frac{1}{8} + \frac{1}{16} = 87,6875$

Algoritmen fra forelesningen gir for (a):
$0 \xrightarrow{\cdot 2+1} 1 \xrightarrow{\cdot 2+1} 3 \xrightarrow{\cdot 2+0} 6 \xrightarrow{\cdot 2+0} 12 \xrightarrow{\cdot 2+1} 25 \xrightarrow{\cdot 2+0} 50 \xrightarrow{\cdot 2+1} 101$

Oppgave 2.3

Overfør følgende tall fra desimal til binær form.

(a) 826_{10}

(b) $0,34375_{10}$

(c) $1604,175_{10}$

(d) $-471,25_{10}$

Repetisjon fra boka

- n div 2 er kvotienten når n deles med 2
- n mod 2 er resten når n er deles med 2

Eksempel

$12 \text{ div } 2 = 6$	$13 \text{ div } 2 = 6$
$12 \text{ mod } 2 = 0$	$13 \text{ mod } 2 = 1$

Algoritme for å overføre fra desimal til binær form (baklengs)

1. Input n [n et naturlig tall]
2. **Repeat**
   2.1. Output n mod 2
   2.2. $n \leftarrow n \text{ div } 2$

   until $n = 0$

Eksempel

$$\begin{array}{ccc} 2 & 8 & \\ & 4 & 0 \\ & 2 & 0 \\ & 1 & 0 \\ & 0 & 1 \\ \hline \end{array}$$

$$8_{10} = 1000_2$$

$$\begin{array}{ccc} 2 & 7 & \\ & 3 & 1 \\ & 1 & 1 \\ & 0 & 1 \\ \hline \end{array}$$

$$7_{10} = 111_2$$

Løsning (a) (826_{10})

$$
\begin{array}{cll}
2 & 826 & \\
 & 413 & 0 \\
 & 206 & 1 \\
 & 103 & 0 \\
 & 51 & 1 \\
 & 25 & 1 \\
 & 12 & 1 \\
 & 6 & 0 \\
 & 3 & 0 \\
 & 1 & 1 \\
 & 0 & 1 \\
\hline
\end{array}
$$

$$826_{10} = 1100111010_2$$

Mer repetisjon fra boka

- Anta at vi vil finne den binære formen til $0,375_{10}$.
- Siden $0,375$ er mindre enn $0,5$ vil første bit etter komma være 0.
- Siden $2 \cdot 0,375$ er mindre enn 1 vil første bit etter komma være 0.

- $\lfloor n \rfloor$ er heltallsdelen av n
- $\mathrm{frac}(n)$ er n minus heltallsdelen av n
- Altså er $n = \lfloor n \rfloor + \mathrm{frac}(n)$

Første bit i den binære formen til n er heltallsdelen av $2n$.

Eksempel

$\lfloor 2,7 \rfloor = 2$
$\mathrm{frac}(2,7) = 0,7$

$0,5_{10} = 0,1_2$
$0,25_{10} = 0,01_2$
$0,75_{10} = 0,11_2$

Utkast til algoritme

1. Input n $[0 \leqslant n \leqslant 1]$
2. **Repeat**
   2.1. $m \leftarrow 2n$
   2.2. Output $\lfloor m \rfloor$
   2.3. $n \leftarrow \text{frac}(m)$

   until $n = 0$

Hva hvis $n = 0$ aldri inntreffer?

Eksempel

$$\begin{array}{ll} & ,25 \quad 2 \\ 0 & ,5 \\ 1 & ,0 \\ \hline \end{array}$$

$$0,25_{10} = 0,01_2$$

$$\begin{array}{ll} & ,375 \quad 2 \\ 0 & ,75 \\ 1 & ,5 \\ 1 & ,0 \\ \hline \end{array}$$

$$0,375_{10} = 0,011_2$$

- Vi kan velge hvor nøyaktig vi vil ha svaret.

Algoritme

1. Input $n, sifre$ $[0 \leqslant n \leqslant 1,\ sifre \geqslant 1,\ sifre$ heltall$]$
2. $i \leftarrow 0$
3. **Repeat**
   - 3.1. $i \leftarrow i + 1$
   - 3.2. $m \leftarrow 2n$
   - 3.3. Output $\lfloor m \rfloor$
   - 3.4. $n \leftarrow \mathrm{frac}(m)$

   until $n = 0$ **or** $i = sifre$

Løsning (b) $(0,34375_{10})$

$$\begin{array}{ll} & ,34375 \quad 2 \\ 0 & ,6875 \\ 1 & ,375 \\ 0 & ,75 \\ 1 & ,5 \\ 1 & ,0 \\ \hline \end{array}$$

$$0,34375_{10} = 0,01011_2$$

Løsning (c) $(1604,175_{10})$

2	1604	
	802	0
	401	0
	200	1
	100	0
	50	0
	25	0
	12	1
	6	0
	3	0
	1	1
	0	1

$$1604_{10} = 11001000100_2$$

	,1875	2
0	,375	
0	,75	
1	,5	
1	,0	

$$0,1875_{10} = 0,0011_2$$

Svar: $11001000100,0011_2$

Løsning (d) $(-471,25_{10})$

$$
\begin{array}{rrr}
2 & 471 & \\
 & 235 & 1 \\
 & 117 & 1 \\
 & 58 & 1 \\
 & 29 & 0 \\
 & 14 & 1 \\
 & 7 & 0 \\
 & 3 & 1 \\
 & 1 & 1 \\
 & 0 & 1 \\
\hline
\end{array}
$$

$$471_{10} = 111010111_2$$

$$
\begin{array}{rll}
 & ,25 & 2 \\
0 & ,5 & \\
1 & ,0 & \\
\hline
\end{array}
$$

$$0,25_{10} = 0,01_2$$

Svar: $-111010111,01_2$

Oppgave 2.4

Overfør følgende tall til binær form, med 5 siffer etter komma.

(a) $0,2_{10}$

(b) $13,47_{10}$

Løsning (a)

$$
\begin{array}{ll}
 & ,2 \quad 2 \\
0 & ,4 \\
0 & ,8 \\
1 & ,6 \\
1 & ,2 \\
0 & ,4 \\
\hline
\end{array}
$$

$$0,2_{10} = 0,00110_2$$

Oppgave 2.4

Overfør følgende tall fra desimal til binær form, med 5 siffer etter komma.

(a) $0,2_{10}$

(b) $13,47_{10}$

Løsning (b)

$$
\begin{array}{ccc}
2 & 13 & \\
 & 6 & 1 \\
 & 3 & 0 \\
 & 1 & 1 \\
 & 0 & 1 \\
\hline
 & \multicolumn{2}{r}{13_{10} = 1101_2}
\end{array}
$$

$$
\begin{array}{ccc}
 & ,47 & 2 \\
0 & ,94 & \\
1 & ,88 & \\
1 & ,76 & \\
1 & ,52 & \\
1 & ,04 & \\
\hline
\multicolumn{3}{r}{0,2_{10} = 0,01111_2}
\end{array}
$$

Svar: $1101,01111_2$

Oppgave 2.5

Hva er resultatet på verdien til et naturlig tall hvis

(a) 0 legges til dets binære representasjon?

(b) 1 legges til dets binære representasjon?

Løsning

(a) Tallet blir fordoblet.
Eks: $1001_2 = 9_{10}$ og $10010_2 = 18_{10} = 2_{10} \times 9_{10}$.

(b) Tallet blir fordoblet og økt med 1.
Eks: $1001_2 = 9_{10}$ og $10011_2 = 19_{10} = 2_{10} \times 9_{10} + 1_{10}$.

Oppgave 2.8

Skriv ned Horners metode som pseudokode. Anta at basen blir gitt som første argument og deretter at tallet som skal overføres til desimal form blir gitt som en liste med sifre.

Horners metode baserer seg på følgende:

- $34_7 = 3 \cdot 7 + 4$
- $345_7 = 3 \cdot 7^2 + 4 \cdot 7 + 5 = (3 \cdot 7 + 4) \cdot 7 + 5$
- $3456_7 = 3 \cdot 7^3 + 4 \cdot 7^2 + 5 \cdot 7 + 6 = (((3 \cdot 7 + 4) \cdot 7) + 5) \cdot 7 + 6$

Løsning

1. Input base
2. Input $x_1, \ldots, x_n$
3. $\mathrm{resultat} \leftarrow x_1$
4. **For** $i = 2$ **to** n **do**
   4.1. $\mathrm{resultat} \leftarrow (\mathrm{resultat} \cdot \mathrm{base}) + x_i$
5. Output $\mathrm{resultat}$