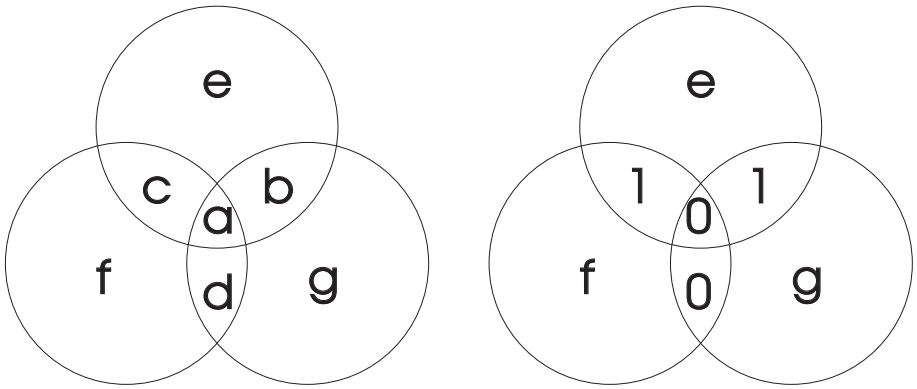
Fejlkorligerende køder
$\longmapsto$
Fejlkorrigerende koder

Olav Geil

Skal man sende en fødselsdagsgave til fætter Børge, så pakker man den godt ind i håb om, at kun indpakningen er beskadiget ved modtagelsen. Noget tilsvarende gør man ofte, når elektronisk data skal sendes via en såkaldt informationskanal. Det værktøj, man bruger, kaldes fejlkorrigerende koder. Uden fejlkorrigerende koder ville såvel en CD-afspiller som en DVD-afspiller være ubrugelig, og billeder sendt til jorden fra Mars eller månen ville være af en meget dårlig kvalitet.

Et eksempel på en fejlkorrigerende kode er den såkaldte binære Hammingkode. Her indpakkes beskeder af 4 binære symboler til ialt 7 symboler. Som vi skal se senere, beskytter denne indpakning de 4 symboler. Først et eksempel på, hvordan vi pakker ind.

Eksempel 1:
Beskeden $(0, 1, 1, 0)$ indkodes ved hjælp af følgende figur Vi tilføjer tre ekstra symboler,



Figur 1:

kaldet efg, ud fra følgende regler:

Summen af elementerne i hver af de tre cirkler skal være lige. Eller sagt i et mere passende sprogbrug, summen i hver af de tre cirkler skal være 0 modulo 2.

Øverste cirkel giver anledning til:

$$\begin{aligned} & 0 + 1 + 1 + e \equiv 0 \pmod 2 \\ \Downarrow \\ & e \equiv 0 \pmod 2 \\ \Downarrow \\ & e = 0. \end{aligned}$$

Venstre cirkel giver anledning til:

$$\begin{aligned} & 0 + 1 + 0 + f \equiv 0 \pmod 2 \\ \Downarrow \\ & 1 + f \equiv 0 \pmod 2 \\ \Downarrow \\ & f = 1. \end{aligned}$$

Højre cirkel giver anledning til:

$$
\begin{aligned}
& 0 + 1 + 0 + g \equiv 0 \pmod 2 \\
\Downarrow \quad & \\
& 1 + g \equiv 0 \pmod 2 \\
\Downarrow \quad & \\
& g = 1.
\end{aligned}
$$

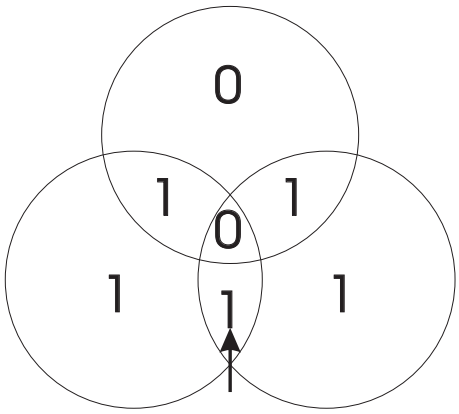
Så beskeden $(0, 1, 1, 0)$ indkodes altså til kodeordet $(0, 1, 1, 0, 0, 1, 1)$.

Opgave 2:
Indkod beskeden $(1, 1, 0, 1)$ til et kodeord i den binære Hammingkode.

Opgave 3:
Indkod beskeden $(1, 1, 1, 1)$ til et kodeord i den binære Hammingkode.

I det følgende ser vi på, hvordan de tre ekstra symboler beskytter de 4 informationssymboler.

Eksempel 4:
Lad os sige, at du befinder dig i den modtagende ende af informationskanalen og at du modtager ordet $(0, 1, 1, 1, 0, 1, 1)$. Et første gæt ville være, at beskeden er $(0, 1, 1, 1)$. Men summen er ikke nul i alle de tre cirkler i nedenstående figur. Så der er altså ikke tale om



Figur 2:

et rigtigt kodeord. Der er sket fejl. Det er naturligt at starte med at antage, at der kun er sket en enkelt fejl. Som det fremgår af nedenstående figur, er dette en mulighed. Nemlig hvis symbolet 1 i positionen d i virkeligheden skulle have været 0. Vi konkluderer, at

det afsendte kodeord i virkeligheden var $(0,1,1,0,0,1,1)$ og at beskeden derfor er $(0,1,1,0)$.

Opgave 5:
Du modtager ordet $(0,0,1,1,0,0,1)$. Er dette et kodeord? Hvis vi nu antager, at der højst er sket en fejl, hvad er så det rigtige kodeord? Hvad er beskeden?

Opgave 6:
Du modtager ordet $(0,1,1,0,1,0,0)$. Hvis vi nu antager, at der højst er sket en fejl, hvad er så det rigtige kodeord? Hvad er beskeden?

Den binære Hammingkode er et eksempel på en såkaldt lineær kode. Det betyder, at hvis $\vec{c}_1 = (a,b,c,d,e,f,g)$ og $\vec{c}_2 = (A,B,C,D,E,F,G)$ er to kodeord, så er summen $\vec{c}_1 + \vec{c}_2 = (a+A, b+B, c+C, d+D, e+E, f+F, g+G)$ også et kodeord. At dette rent faktisk er tilfældet ses på følgende måde. Vi konstaterer, at hvis $\vec{c}_1 = (a,b,c,d,e,f,g)$ er et kodeord i Hammingkoden, da må der gælde $a+b+c+e$ er lige, $a+c+d+f$ er lige og $a+b+d+g$ er lige. Tilsvarende har vi, at hvis $\vec{c}_2 = (A,B,C,D,E,F,G)$ er et kodeord i Hammingkoden, da må der gælde $A+B+C+E$ er lige, $A+C+D+F$ er lige og $A+B+D+G$ er lige. Men så er $(a+A)+(b+B)+(c+C)+(e+E) = (a+b+c+e)+(A+B+C+E)$ lige (lige + lige er lige), $(a+A)+(c+C)+(d+D)+(f+F) = (a+c+d+f)+(A+C+D+F)$ er lige og $(a+A)+(b+B)+(d+D)+(g+G) = (a+b+d+g)+(A+B+D+G)$ er lige. Dermed er $(a+A, b+B, c+C, d+D, e+E, f+F, g+G)$ et kodeord i Hammingkoden.

Opgave 7:
Ordene $(0,1,1,0,0,1,1)$ og $(1,1,1,1,1,1,1)$ er begge kodeord i den binære Hammingkode. Find summen. Vis ved hjælp af de tre cirkler, at summen er et nyt kodeord.

Lineære koder er specielt nemme at beskrive.

Eksempel 8:
Hver af de fire rækker i nedenstående system (kaldet en matrix) svarer til et kodeord i den binære Hammingkode.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Vil du indkode for eksempel beskeden $(1,1,0,1)$, så kan du gøre det ved at lægge 1 gange række 1 sammen med 1 gange række 2, sammen med 0 gange række 3 og sammen med 1 gange række 4. Hvorfor virker denne metode? Jo for det første er Hammingkoden lineær, så række 1 plus række 2 er et nyt kodeord. Til dette lægges række 4. Igen fås et kodeord, da summen af to kodeord jo er et kodeord igen. Så række 1 plus række 2 plus række 4 er helt klart et kodeord. Hvad er de fire første tal i det herved udregnede kodeord? Jo ved

at betragte matricen ovenfor ser vi, at disse netop udgør vores besked $(1,1,0,1)$. Metoden virker selvfølgelig ikke bare for beskeden $(1,1,0,1)$ men for alle beskeder (a,b,c,d).

Opgave 9:
Benyt metoden fra eksempel 8 til at indkode beskeden $(1,0,1,1)$.

Lister vi alle lovlige kodeord i Hammingkoden, så vil vi opdage, at bortset fra kodeordet $(0,0,0,0,0,0,0)$ så er der ingen kodeord med mindre end 3 ikke-nul symboler. Det mindste antal ikke-nul symboler, der kan forekomme i et kodeord forskelligt fra $(0,0,\cdots,0)$ kaldes minimumsafstanden, og forkortes d. Der gælder følgende simple resultat, som vi dog ikke vil bevise.

Sætning 10:
En lineær kode med minimumsafstand d kan rette $\left\lfloor \frac{d-1}{2} \right\rfloor$ fejl (altså $(d-1)/2$ rundet ned).

Opgave 11:
Hvor mange fejl kan den binære Hammingkode rette? Hvis en kode har minimumsafstand $d=4$, hvor mange fejl kan den så rette? Hvad hvis den har minimumsafstand $d=17$?

Indtil videre har vi kun arbejdet med symbolerne 0 og 1. Det vil sige, vi har indtil videre kun arbejdet med det binære alfabet $\mathbb{F}_2 = \{0,1\}$. Man arbejder i praksis med forskellige alfabeter svarende til de såkaldte endelige legemer. Disse har alle størrelsen p^m, hvor p er et primtal. I det følgende ser vi kun på de tilfælde, hvor $m=1$. Altså på de tilfælde, hvor alfabetet indeholder p elementer.

Eksempel 12:
Alfabetet $\mathbb{F}_3 = \{0,1,2\}$ tilfredsstiller ligningen $3=0$. Det vil sige $1+1=2$, $2+1=3=0$ og $2+2=4=3+1=0+1=1$. Vi siger, at vi reducerer modulo 3. Tilsvarende gælder $2\cdot 2 = 4 = 3+1 = 0+1 = 1$. Vi får følgende additions- og multiplikationstabeller.

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

·	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Opgave 13:
Alfabetet $\mathbb{F}_5 = \{0,1,2,3,4\}$ tilfredsstiller ligningen $5=0$. Vi siger, at vi regner modulo 5. Udfyld følgende additions- og multiplikationstabeller.

+	0	1	2	3	4
0					
1					
2					
3					
4					

·	0	1	2	3	4
0					
1					
2					
3					
4					

Vi behandler afslutningsvis de såkaldte Reed-Solomonkoder. Disse er defineret for alle alfabeter.

Eksempel 14:
Lad os for eksempel betragte alfabetet $\mathbb{F}_5$. Lad os sige, at vi gerne vil have beskeder af længde 3 pakket ind i kodeord af længde 5. Vi vil opstille en matrix, som i eksempel 8. Da vi vil have beskeder af længde 3, skal denne matrix have 3 rækker. Vi vælger som første række $(1,1,1,1,1)$. Som anden række vælger vi $(0^1\,1^1\,2^1\,3^1\,4^1) = (0,1,2,3,4)$. Som tredie række vælger vi $(0^2\,1^2\,2^2\,3^2\,4^2) = (0,1,4,4,1)$. Så anden række er altså fremkommet ved at stoppe punkterne $0,1,2,3,4$ ind i polynomiet x, tredie række er fremkommet ved at stoppe punkterne $0,1,2,3,4$ ind i polynomiet x^2 og første række er fremkommet ved at stoppe punkterne $0,1,2,3,4$ ind i polynomiet 1.

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 4 & 4 & 1 \end{bmatrix}$$

Vil vi nu indkode beskeden $(1,2,4)$ lægger vi simpelthen 1 gange række 1 sammen med 2 gange række 2 og sammen med 4 gange række 3. Vi får $1\cdot(1,1,1,1,1)+2\cdot(0^1,1^1,2^1,3^1,4^1)+4\cdot(0^2,1^2,2^2,3^2,4^2) = (1+2\cdot 0^1+4\cdot 0^2, 1+2\cdot 1^1+4\cdot 1^2, 1+2\cdot 2^1+4\cdot 2^2, 1+2\cdot 2^1+4\cdot 2^2, 1+2\cdot 3^1+4\cdot 3^2, 1+2\cdot 4^1+4\cdot 4^2) = (p(0),p(1),p(2),p(3),p(4))$, hvor $p(x) = 1+2x+4x^2$. Generelt indkodes beskeden (a,b,c) til $(p(0),p(1),p(2),p(3),p(4))$, hvor $p(x) = a+bx+cx^2$.
Bemærk, at denne indkodning har en enkelt ulempe! Når først vi har indkodet vores besked, da kan vi ikke direkte se af kodeordet, hvad beskeden er. F.eks. indkodes $(1,2,4)$ ovenfor til $(1,2,1,3,3)$. Dette er ikke et problem i praksis.

Opgave 15:
Lad os igen betragte alfabetet $\mathbb{F}_5$. Lad os nu sige, at vi gerne vil have beskeder af længde 2 pakket ind i kodeord af længde 5. Opskriv den tilhørende matrix. Hvilket polynomium skal du bruge for at indkode beskeden $(2,2)$?

Opgave 16:
Samme som opgave 15 men med beskeder af længde 4. Hvilket polynomium skal du bruge for at indkode beskeden $(2,1,3,1)$?

En af fordelene ved Reed-Solomonkoderne er, at det er så let at finde deres minimumsafstand.

Eksempel 17:
Betragt koden fra eksempel 14. Et kodeord findes ved at benytte et polynomium af grad højst 2. Et ikke-nul polynomium af grad højst 2 har højst to nulpunkter. Derfor må der i ethvert ikke-nul kodeord forekomme højst 2 nuller. Men så er der mindst 3 ikke-nul symboler. Hermed er minimumsafstanden d lig 3. Vi kan rette $\lfloor \frac{3-1}{2} \rfloor = 1$ fejl.

Opgave 18:
Benyt metoden fra eksempel 17 til at finde minimumsafstanden af koden i opgave 16. Hvor mange fejl kan vi rette?

Opgave 19:
Benyt metoden fra eksempel 17 til at finde minimumsafstanden af koden i opgave 15.

Opgave 20:
Find selv på flere eksempler. Prøv for eksempel at lave Reed-Solomonkoder over alfabetet $\mathbb{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$.

I det følgende vil vi gerne arbejde os hen imod at vise, at et polynomium af grad n højst kan have n forskellige nulpunkter. Vi får brug for divisionsalgoritmen for polynomier. Vi nupper lige et eksempel med heltal først.

Eksempel 21:
Betragt de positive heltal $a = 7612$ og $b = 7$. Vi vil gerne finde ikke-negative heltal q og r således, at der gælder $a = q \cdot b + r$ og således, at $r < b$ holder. Divisionsalgoritmen er velkendt, men er måske her skrevet lidt anderledes op end I er vant til.

$$
\begin{array}{rll}
7612 & : 7 = & 1 \cdot 1000 \\
7000 & & \\
\hline
612 & & +0 \cdot 100 \\
000 & & \\
\hline
612 & & +8 \cdot 10 \\
560 & & \\
\hline
52 & & +7 \cdot 1 \\
49 & & \\
\hline
3 & &
\end{array}
$$

Vi konkluderer, at $7612 = 1087 \cdot 7 + 3$. Altså er $q = 1087$ og $r = 3$.

Opgave 22:
Lad $a = 9635$ og lad $b = 4$. Find q og r, så $a = q \cdot b + r$, hvor q og r er ikke-negative heltal, og hvor $r < 4$.

Vi konstaterer, at divisionsalgoritmen altid stopper (det vil sige, den bliver færdig i endelig mange trin). Det er også let at indse, at vi altid vil ende med et r som opfylder $0 \leq r < b$. At vores bud på q og r altid opfylder $a = qb + r$, ses nok lettest ved at betragte eksempel 21 en gang til.

Eksempel 23:
Vi betragter udregningerne i eksempel 21. Sidste linie i den ni-liniers udregning svarer til r. Ottende linie svarer til $7 \cdot b$ og syvende linie svarer til $7 \cdot b + r$. Sjette linie svarer til

$80 \cdot b$ og femte linie svarer til $80 \cdot b + 7 \cdot b + r$. Fjerde linie svarer til $0 \cdot b$ og tredie linie svarer til $0 \cdot b + 80 \cdot b + 7 \cdot b + r$. Anden linie svarer til $1000 \cdot b$ og første linie svarer til $1000 \cdot b 0 \cdot b + 80 \cdot b + 7 \cdot b + r$. Vi har $1000 \cdot b 0 \cdot b + 80 \cdot b + 7 \cdot b + r = (1000 + 0 + 80 + 1) \cdot b + r = q \cdot b + r$. Men første linie er jo også lig a, og vi har $a = q \cdot b + r$.

Vi går nu over til at se på divisionsalgoritmen for polynomier. Det virker selvfølgelig lidt mere abstrakt end ovenstående, men er dybest set det samme.
Eksempel 24:
Betragt polynomierne $a(x) = 8 \cdot x^3 + 6 \cdot x^2 + 1$ og $b(x) = 2 \cdot x + 1$. Vi vil finde polynomier $q(x)$ og $r(x)$ så $a(x) = q(x) \cdot b(x) + r(x)$, hvor graden af $r(x)$ er mindre end graden af $b(x)$.

$$
\begin{array}{rll}
8x^3 + 6x^2 + 0 \cdot x + 1 & : 2x + 2 = & 4x^2 \\
8x^3 + 8x^2 & & \\
\hline
-2x^2 + 0x + 1 & & -x \\
-2x^2 - 2x & & \\
\hline
2x + 1 & & +1 \\
2x + 2 & & \\
\hline
-1 & &
\end{array}
$$

Så $q(x) = 4x^2 - x + 1$ og $r(x) = -1$ opfylder $a(x) = q(x) \cdot b(x) + r(x)$ og ganske rigtigt er $r(x)$ af grad nul.

Opgave 25:
Lad $a(x) = 8x^4 + 4x^3 - 2x^2 + x + 10$ og $b(x) = x + 4$. Find ved hjælp af divisionsalgoritmen $q(x)$ og $r(x)$, så $a(x) = q(x)b(x) + r(x)$ og så graden af $r(x)$ er mindre end graden af $b(x)$.

Opgave 26:
Lad $a(x) = 8x^4 + 4x^3 - 2x^2 + x + 10$ og $b(x) = x^2 + x + 4$. Find ved hjælp af divisionsalgoritmen $q(x)$ og $r(x)$, så $a(x) = q(x)b(x) + r(x)$ og så graden af $r(x)$ er mindre end graden af $b(x)$.

Opgave 27:
I denne opgave benyttes metoden fra eksempel 23. Denne gang tager vi udgangspunkt i eksempel 24. Forklar, hvorfor vi kan være sikre på, at divisionsalgoritmen altid stopper. Forklar hvorfor vi altid ender med et $r(x)$ som har grad mindre end $b(x)$. Forklar hvorfor der altid gælder $a(x) = q(x)b(x) + r(x)$.

Eksempel 28:
Vi øger nu abstraktionsniveauet yderligere idet vi istedet for de rationale tal vil arbejde med legemet $\mathbb{F}_3$ fra eksempel 12. Blandt andet gælder der altså nu $2 \cdot 2 = 1$ og $1 - 2 = 2$. Vi betragter polynomierne $a(x) = 2x^3 + 2x^2 + x + 1$ og $b(x) = x + 2$. Vi søger som ovenfor $q(x)$ og $r(x)$, men denne gang med koefficienter i $\mathbb{F}_3$.

$$
\begin{array}{rl}
2x^3+2x^2+x+1 & : x+2 = \quad 2x^2 \\
2x^3+x^2 & \\
\hline
x^2+x+1 & \qquad\qquad +x \\
x^2+2x & \\
\hline
2x+1 & \qquad\qquad +2 \\
2x+1 & \\
\hline
0 &
\end{array}
$$

Så $q(x) = 2x^2 + x + 2$ og $r(x) = 0$.

Opgave 29:
Ligesom i ovenstående eksempel regner vi i legemet $\mathbb{F}_3$. Lad $a(x) = 3x^2 + x^2 + 2x + 1$ og $b(x) = 2x + 2$. Find ved hjælp af divisionsalgoritmen $q(x)$ og $r(x)$.

Opgave 30:
I dette eksempel regner vi i legemet $\mathbb{F}_3$ fra eksempel 13. Lad $a(x) = 2x^3 + 2x^2 + x + 1$ og $b(x) = x + 2$ og find ved hjælp af divisionsalgoritmen $q(x)$ og $r(x)$.

Eksempel 31:
Betragt polynomiet $f(x) = x^3 - 7x^2 + 14x - 8$ med koefficienter i de rationale tal. Ved indsættelse kan man overbevise sig om, at $f(x)$ har nulpunkterne $1, 2$ og 4. Vi vil nu bevise, at det ikke har flere nulpunkter end disse. Vi får kraftigt brug for divisionsalgoritmen. Først tages udgangspunkt i nulpunktet 1. Vi søger som sædvanlig $q(x)$ og $r(x)$ så

$$x^3 - 7x^2 + 14x - 8 = q(x) \cdot (x - 1) + r(x) \tag{1}$$

og så $r(x)$ har grad mindre end polynomiet $x - 1$. Men så er $r(x)$ altså en konstant. Vi ved, at $1^3 - 7 \cdot 1^2 + 14 \cdot 1 - 8 = 0$, og selvfølgelig gælder også $q(1) \cdot (1 - 1) = 0$. Vi konkluderer, at $r(x)$ derfor bliver nødt til at være lig 0.

$$
\begin{array}{rl}
x^3-7x^2+14x-8 & : x-1 = \quad x^2 \\
x^3-x^2 & \\
\hline
-6x^2+14x-8 & \qquad\qquad -6x \\
-6x^2+6x & \\
\hline
8x-8 & \qquad\qquad +8 \\
8x-1 & \\
\hline
0 &
\end{array}
$$

Vi har altså $f(x) = (x^2 - 6x + 8) \cdot (x - 1) + 0$. Bemærk, at vi som forventet fik rest r lig 0. Vi fortsætter nu med at kigge på nulpunktet 2. Da 2 ikke er nulpunkt i polynomiet $x - 1$,

må 2 i følge ovenstående være nulpunkt i $x^2 - 6x + 8$. Vi regner ligesom ovenfor.

$$\begin{array}{rl} x^2 - 6x + 8 & : x - 2 = \quad x \\ x^2 - 2x & \\ \hline -4x + 8 & \qquad\qquad -4 \\ -4x + 8 & \\ \hline 0 & \end{array}$$

Hermed haves $x^2 - 6x + 8 = (x-2)\cdot(x-4)$, og dermed altså $f(x) = (x-1)\cdot(x-2)\cdot(x-4)$. Ud fra denne opskrivning (faktorisering) er det klart, at der ikke kan være flere nulpunkter end de tre vi allerede kender.

Sætning 32:
Lad $f(x)$ være et polynomium af grad g over et givet legeme. Da kan $f(x)$ højst have g forskellige nulpunkter i legemet.

Opgave 33:
Skitser med baggrund i eksempel 31 et bevis for sætning 32 (hint: antag, at $f(x)$ har mindst $d + 1$ forskellige nulpunkter og nå frem til en modstrid).

Sætning 34:
Betragt Reed-Solomon koden over alfabetet $\mathbb{F}_p$ defineret ved hjælp af polynomier af grad op til g. Koden har minimumsafstand mindst lig $p - g$.

Opgave 35:
Bevis sætning 34 ved hjælp af sætning 32.

Opgave 35:
Lad polynomiet $f(x) = x^3 + 2x^2 - x - 2$ have koefficienter i $\mathbb{F}_3$. Find ved indsættelse rødderne i $\mathbb{F}_3$ for dette polynomium. Benyt metoden fra opgave 31 til at finde ud af, hvilken af disse rødder, der er dobbeltrod.

Opgave 36:
Vi betragter divisionsalgoritmen for positive heltal (behandlet i eksempel 21). Vi vil argumentere for, at det r vi finder ved hjælp af algoritmen altid er det mindste positive heltal, som opfylder, at $a = q \cdot b + r$. Lad q og q' være positive heltal og lad r og r' være positive heltal mindre end b således at $a = q \cdot b + r$ og $a = q' \cdot b + r'$. Vi vil vise, at så gælder $q = q'$ og $r = r'$. Betragt differensen $0 = a - a = (q \cdot b + r) - (q' \cdot b + r') = (q - q') \cdot b - (r - r')$. Overbevis dig om, at der er følgende to muligheder.
Mulighed 1: $q = q'$ og $r = r'$. Mulighed 2: Der gælder $\mid (q - q') \cdot b \mid \geq b$ og $\mid r - r' \mid < b$.
Argumenter for, at mulighed 2 aldrig kan forekomme.
Genereliser resultatet til divisionsalgoritmen for polynomier.

Opgave 37:
Legemet $\mathbb{F}_{2^2} = \mathbb{F}_4$ består af elementerne

$$\{0, 1, \alpha, 1 + \alpha\} \tag{2}$$

Der benyttes følgende regneregler

$$2 = 0 \qquad \text{og} \qquad \alpha^2 + \alpha + 1 = 0 \tag{3}$$

Første regel indebærer, at $-$ og $+$ er det samme. Kan du se hvorfor? Vis, at der derfor gælder $\alpha^2 = \alpha + 1$. Opskriv additionstabellen for $\mathbb{F}_4$. Hvilke elementer i (2) svarer til α^2 og α^3? Konkluder, at $\{\alpha, \alpha^2, \alpha^3\}$ svarer til ikke-nul elementerne i (2). Udnyt, at $\alpha^2\alpha^2 = \alpha^4 = \alpha^3\alpha$, til at finde det element i (2), som svarer til $\alpha^2\alpha^2$. Opskriv multiplikationstabellen for $\mathbb{F}_4$.