Opgave 1

Regning med rest

Den positive rest, man får, når et helt tal a divideres med et naturligt tal n, betegnes

$$\text{rest}(a,n)$$

Hvis $r = \text{rest}(a,n)$ kan a skrives $a = k \cdot n + r$ hvor $0 \leq r < n$ og k er et helt tal. Der gælder at

$$\text{rest}(a + b,n) = \text{rest}(\text{rest}(a,n) + \text{rest}(b,n),n) \quad (2)$$

$$\text{rest}(a \cdot b,n) = \text{rest}(\text{rest}(a,n) \cdot \text{rest}(b,n),n) \quad (3)$$

1. Giv eksempel på, at der ikke gælder

$$\text{rest}(a + b,n) = \text{rest}(a,n) + \text{rest}(b,n)$$

2. Udregn i hånden ved brug af (2) og (3):

$$\text{rest}(66 \cdot 84, 20) \text{ og } \text{rest}(3^{10}, 20)$$

3. Gør rede for, at

$$\text{rest}(a,n) = \text{rest}(b,n) \Leftrightarrow n \text{ går op i } a - b \quad (4)$$

Til sidens top

Opgave 2

Fermats lille sætning

Lad p være et primtal, der ikke går op i a. Da gælder

$$\text{rest}(a^{p-1}, p) = 1 \quad (5)$$

De følgende to opgaver giver beviset for sætningen.

Vi ser på tallene a , $2a$, … , $(p-1)a$ og deres rest b_1 , b_2 , … , b_{p-1} ved division med p.

1. Giv et indirekte bevis for at b'erne alle er forskellige og at ingen af dem er 0. Der gælder altså at b'erne er tallene 1 , 2 ,…, p -1 blot i en anden rækkefølge

2. Gør rede for at

$$\text{rest}(a \cdot 2a \cdot \ldots \cdot (p-1)a, p) = \text{rest}(1 \cdot 2 \cdot \ldots \cdot (p-1), p)$$

og at dette medfører Fermats lille sætning.

Sæt p =17. For x tilhørende mængden af hele tal fra 1 til 16 ser vi på funktionerne:

$$f(x) = \text{rest}(x^3, 17) \text{ og } g(x) = \text{rest}(x^{11}, 17)$$

3. Vis at $g(f(x)) = x$

Man kan altså kryptere med f og dekryptere med g.

Til sidens top

Opgave 3

Eratosthenes si

Der findes ikke nogen formel til at finde for eksempel primtal nummer 1000 uden at bestemme de foregående, men der er heller ikke nogen enkel metode til bare at afsløre om et tal er et primtal. Vi vil her se på den mest nærliggende metode, nemlig at forsøge at finde divisorer fra neden af.

Opgaverne er formuleret under forudsætning af, at der er en TI-83 lommeregner til rådighed; men opgaven kan nemt tilpasses en CAS lommeregner eller et computerprogram.

Lommeregneren har indbygget funktionen int(*x*), der beregner den hele del af *x*, altså det største hele tal mindre eller lig *x*. Ved hjælp af denne funktion kan rest(*n,k*) beregnes som

$$n - k \cdot \text{int}\left(\frac{n}{k}\right)$$

Dette udtryk indgår i følgende program PrgmPRIM, der finder ud af om et givet tal *N* er et primtal, ved at undersøge om nogen af tallene fra 2 til kvadratroden af *N* går op i *N*.

```
: Input "N = ",N
: For (K,2,Int(√(N)))
: If N-K*int(N/K)=0
: Then : Disp "NEJ"
: Goto A
: End : End
: Disp "JA"
: Lbl A
```

1. Undersøg hvor lang tid det tager lommeregneren at køre programmet for hver af følgende værdier af N:

1009 10007 100003 1000003 10000019

og opstil ud fra disse tal en sammenhæng mellem antal cifre og tid. Sammenlign den tid, det ville tage at undersøge et tal med 200 cifre, med universets alder, som anslås at være $14 \cdot 10^9$ år.

2. Udled teoretisk en sammenhæng mellem antal cifre og tid.

Metoden i programmet er i raffineret form *Eratosthenes si* (240 f.kr.). I talrækken:

2 , 3 , 4 , 5 , 6 , 7 , 8 , 9, ... , *n*

udtages det mindste tal 2 og alle egentlige multipla af 2 bortsies:

3 , 5 , 7 , 9 , 11, … , *n*

dernæst udtages 3 i listen og alle egentlige multipla af det 3 bortsies:

5 , 7 , 11 , … , *n*

Således fortsættes succesivt med at udtage det mindste tal i listen og bortsi egentlige multipla heraf. Når processen standser, har vi udtaget alle primtal mindre end eller lig *n*. Metoden kan afprøves i praksis for *n* = 400 på

www.faust.fr.bw.schule.de/mhb/eratosiv.htm

Til sidens top

Opgave 4

AKS-algoritmen

For at kunne forstå ideen i AKS-algoritmen får man brug for at

$$(x+1)^n = x^n + k_{n-1}x^{n-1} + \ldots + k_2 x^2 + k_1 x + 1 \quad (6)$$

hvor koefficienterne k_i er hele tal, der ifølge binomialformlen kan udregnes som

$$k_i = \frac{n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n-i+1)}{1 \cdot 2 \cdot 3 \cdot \ldots \cdot i} \quad (7)$$

1. Udregn $(x+1)^4$ og $(x+1)^5$ ved (6) og (7).

Der gælder følgende primtals kriterium:

Tallet *n* er et primtal, hvis og kun hvis *n* går op i samtlige koefficienter i polynomiet

$$(x+1)^n - x^n - 1 \quad (8)$$

2. Gør rede for at ovenstående påstand er rigtig.

Dette elegante primtals kriterium er imidlertid slet ikke til beregningsmæssig nytte. For store *n* er det umuligt at beregne $(x+1)^n$. Ideen er nu istedet at beregne resten *r* (x) af $(x+1)^n - x^n - 1$ ved polynomiers division med polynomiet $x^r - 1$ for et passende *r*. Denne rest kan beregnes ved brug af regnereglerne (2) og (3) uden at behøve beregne $(x+1)^n$. Samtidigt gælder (8) i en vis forstand næsten, når $(x+1)^n - x^n - 1$ erstattes med *r* (*x*), som det antydes af eksemplet i de følgende opgaver:

3. Når *r* =2 er det muligt at bestemme *r* (*x*) uden at lave divisionen. Sæt $p(x) = (x+1)^n - x^n - 1$. Gør rede for, at *r* (*x*) er et førstegradspolynomium, hvis graf går gennem (-1, *p*(-1)) og (1, *p*(1)) og at $r(x) = (2^{n-1} - 1)(x+1)$ når *n* er ulige.

4. Gør rede for at *n* går op i koefficienterne i *r* (*x*), hvis *n* er et primtal.

Der findes desværre *n*, der ikke er primtal, for hvilke *n* går op i samtlige koefficienter i polynomiet $\text{rest}((x+1)^n - x^n - 1, x^2 - 1)$. Det mindste er 341. Men metoden kan reddes ved så at se på rest ved division med $x^r - 1$ for større primtal *r* end 2. Vi stopper imidlertid her, hvor hovedideen i AKS-algoritmen er illustreret.

5. Udregn rest(2^{340},341) ved at bruge 2^{10} som mellemstation.

Det afgørende ved AKS-algoritmen er, at tidsforbruget ikke vokser eksponentielt som funktion af antallet af cifre, men som en potensfunktion (eller polynomium).

Til sidens top

Opgave 5

Euklids algoritme

Primtallene er byggeklodserne i systemet af hele tal. Ethvert helt tal kan nemlig faktoriseres i et produkt af primtal - endda på en entydig måde. Ved hjælp af primtalsfaktorisering kan man for eksempel i princippet bestemme den største fælles divisor for to tal *a* og *b*, som betegnes sfd(*a,b*).

1. Faktoriser 294 og 60, og angiv den største fælles divisor

I praksis er det ikke muligt at finde største fælles divisor for meget store tal på denne måde, da der er ikke nogen kendt effektiv måde at faktorisere på. Vi vil nu se på Euklids algoritme, der i et overkommeligt antal trin finder den største fælles divisor for to tal. Algoritmen bygger på følgende:

$$\text{sfd}(a,b) = \text{sfd}(b,\text{rest}(a,b)) \quad (9)$$

Eksempel:

Når $a = 294$ og $b = 60$ giver Euklids algoritme:

$$294 = 4 \cdot 60 + 54$$
$$60 = 1 \cdot 54 + 6$$
$$54 = 9 \cdot 6 + 0$$

Største fælles divisor for 6 og 54 er altså 6 (da 6 går op i 54), og 6 er derfor også største fælles divisor for 294 og 60.

Vi ser nu på to tal *a* og *b*, for hvilke den største fælles divisor er 1. Vi skal se, hvordan man kan bestemme hele tal *x* og *y*, således at

$$1 = ax+by$$

eller med andre ord: så $a \cdot x$ har resten 1 ved division med *b*. Metoden, der går ud på først at opskrive Euklids algoritme, og dernæst i en vis forstand regne baglæns, illustreres med følgende eksempel:

Når $a = 7$ og $b = 23$ giver Euklids algoritme:

$$23 = 3 \cdot 7 + 2$$
$$7 = 3 \cdot 2 + 1$$

I hver af ligningerne isoleres det sidste led:

$$2 = 23 - 3 \cdot 7$$

$$1 = 7 - 3 \cdot 2$$

I den nederste ligning erstattes 2 med højresiden i den ovenstående:

$$1 = 7 - 3 \cdot (23 - 3 \cdot 7) = - 3 \cdot 23 + 10 \cdot 7$$

Der gælder altså at $1 = x \cdot 7 + y \cdot 23$ for $x = 10$ og $y = -3$.

2. Bestem et tal *x*, således at $11 \cdot x$ har resten 1 ved division med 62.

3. Bestem hele tal *x* og *y*, således at $1000 = 7x + 11y$

Reference:

Johan P. Hansen og Henrik Spalk: *Algebra og talteori* , Gyldendal(2002)

Til sidens top

Opgave 6

Nøglekonstruktion i RSA-system

Første trin i opbygningen af et RSA-system er at vælge tallet *n*, der skal være et produkt af to primtal *p* og *q*.

I de følgende trin får vi brug for følgende begreb: to naturlige tal kaldes primiske, hvis deres største fælles divisor er 1. Antallet af naturlige tal, der er primiske med *n* og mindre end *n*, kaldes *f*.

1. Bestem *f*, når *n* = 15

2. Gør rede for at $f = (p - 1) \cdot (q - 1)$.
(Vink: Bestem antallet af tal mindre end *n*, der ikke er primiske med *n*).

Tallet *k*, der skal være en del af den offentlige nøgle, skal vælges som et tal, der er primisk med *f*. Man kan vise (ved blandt andet at kopiere ideen i beviset for Fermats lille sætning), at hvis *d* er en løsning til ligningen rest($k \cdot d$, *f*) = 1 så gælder at

$$\text{rest}\,(x^{kd}, n) = x \quad (10)$$

En sådan løsning *d* kan altså bruges som hemmeligt nøgletal.

3. Sæt (*k*,*n*) = (3,33). Bestem den hemmelige nøgle *d*.

Reference:

P. Landrock og K. Nissen: *Kryptologi*, Abacus (1997)

Til sidens top

Opgave 7

Elliptiske kurver

1. Betragt den elliptiske kurve med ligningen

$$y^2 = x^3 - x + 1$$

Beregn koordinaterne til *P* + *Q* og til 2*P*, når *P* =(1,1) og *Q* =(3,5).

2. Kurver med en ligning af formen

$$y^2 = x^3 + ax + b$$

kan for specielle værdier af *a* og *b* have punkter, hvor der ikke er en tangent. Det sker, hvis højresiden af ligningen har en dobbeltrod, det vil sige, hvis der findes tal *a* og *b* så

$$x^3 + ax + b = (x - \alpha)^2 (x - \beta) \quad (11)$$

Vis, at (11) medfører, at

$$\beta = -2\alpha \text{ og } 4a^3 - 27b^2 = 0$$

Tegn kurven med ligningen

$$y^2 = x^3 - 3x + 2$$

og undersøg specielt kurven omkring punktet (1,0).

Til sidens top

Opgave 8

Elliptiske kurver modulo *p*

At to tal *a* og *b* har samme rest ved division med *p* skrives

$$a = b \pmod{p}$$

1. Bestem samtlige løsninger (x,y) modulo 5 til ligningen

$$y^2 = x^3 - x + 1$$

Når en meddelelse skal repræsenteres som et punkt på kurven, omsættes meddelelsen til at tal, og der vælges et punkt, hvis x -koordinat er lig med eller tæt på dette tal. For at finde y -koordinaten indsættes i højresiden af kurveligningen og regnes modulo p, hvilket giver en ligning af typen

$$y^2 = c \pmod{p} \quad (12)$$

Resultatet i den følgende opgave kan bruges til at afgøre om (12) har en løsning:

2. Benyt Fermats lille sætning til at vise at

$$c^{\frac{p-1}{2}} = 1 \pmod{p} \quad (13)$$

hvis $c = k^2 \pmod{p}$.

Hvis (12) har løsninger kan en løsning bestemmes ved brug af resultatet i den følgende opgave, når p har rest 3 ved division med 4.

3. Benyt Fermats lille sætning til at vise at

$$c^{(p+1)/4} \quad (14)$$

er løsning til (12) hvis (13) er opfyldt. Bemærk at eksponenten i (14) kun er et helt tal når p har rest 3 ved division med 4.

4. Vis at ligningen

$$y^2 = 6 \pmod{23}$$

har en løsning og bestem en løsning.

Ud fra den geometriske beskrivelse af addition af punkter kan man udlede følgende koordinatudtryk. Lad $P(x_1, y_1)$ og $Q(x_2, y_2)$ og $P + Q = (x_3, y_3)$ så er

$$(x_3, y_3) = (s^2 - x_1 - x_2, -y_1 + s(x_1 - x_3)) \quad (15)$$

Her er s hældningskoefficienten til linjen ℓ, og formlen for s afhænger af om ℓ er sekantline ($P \bullet, Q$) eller tangentlinie ($P = Q$):

sekantlinie: $s = \frac{y_2 - y_1}{x_2 - x_1}$ tangentlinie: $s = \frac{3x_1^2 + a}{2y_1}$

Når der regnes modulo et tal p, skal p være et primtal for at division kommer til at fungere fornuftigt.

5. Bestem $\frac{1}{5} \pmod{17}$, som er løsningen til ligningen $5 \cdot x = 1$ modulo 17. Udregn dernæst $\frac{3}{5} \pmod{17}$.

Til sidens top