Heltalsfaktorisering

Køreplan

01005 Matematik 1 — FORÅR 2017

1 Introduktion

Som I sikkert ved, er et primtal et heltal (større end 1), som kun 1 og tallet selv går op i; de første ti primtal er altså 2, 3, 5, 7, 11, 13, 17, 19, 23 og 29. Bemærk, at tallet 1 per definition *ikke* er et primtal. Ethvert heltal større end 1 kan skrives som et produkt af primtal, og bortset fra rækkefølgen kan dette kun gøres på én måde. Vi kalder dette at "opløse" et tal i dets primfaktorer, eller blot at "faktorisere" tallet.

Ønsker man at opløse et givet tal n i dets primfaktorer, kan man naturligvis undersøge primtallene én efter én i voksende rækkefølge og finde dem, der går op i n. Lad os kalde denne metode "divisionsmetoden". For visse tal n, er dette den nemmeste måde at gøre det på. Det gælder især for små heltal, samt for heltal som udelukkende har små primfaktorer. Det er klart, at primfaktorerne kan gå op i n mere end én gang; det gælder f.eks. tallet 12, som er lig $2 \cdot 2 \cdot 3$, dvs. 2 går op to gange.

Hvis n udelukkende har store primfaktorer, tager den nævnte metode meget lang tid. Tag f.eks. tallet $n = 2305843027467304993$. Dette tal består af kun to primfaktorer af cirka samme størrelse, og hvis vi skulle bruge divisionsmetoden, ville vi være nødt til at dividere de første cirka 54 millioner primtal op i n, før vi ville finde en primfaktor. Det tager tid at dividere store tal med hinanden; hvis vi kan foretage 20000 divisioner i sekundet, vil det tage knap en time at faktorisere n med divisionsmetoden.

Heltal med store primfaktorer kan faktoriseres langt hurtigere end med divisionsmetoden. I dette projekt skal man forstå og implementere en bestemt metode til heltalsfaktorisering, som vi kalder "Dixons tilfældige kvadraters metode" (udviklet af John D. Dixon omkring 1979).

Denne køreplan skal forstås som en vejledning, og det anbefales at der skrives en sammenhængende rapport, som ikke nødvendigvis behøver at indeholde svar på alle opgaver. Ligeledes er det naturligvis tilladt på egen hånd at udforske delemner nærmere, end der gøres i denne vejledning. Det sidste afsnit i køreplanen indeholder forøvrigt nogle forslag til "ekstraopgaver", man kan forsøge at løse hvis man har tid.

For at forstå Dixons metode må vi først introducere noget talteori.

2 Talteori

Matematik med heltal kaldes også *diskret matematik*. I den diskrete matematik betragter man ofte matematiske strukturer med et endeligt antal elementer. Det svarer til, at når man taler om et klokkeslet, benytter man kun timetal mellem 0 og 23. Hvis klokken er 15 nu, så er den om 12 timer ikke 27, men 3. Uden at tænke videre over det trækker vi altså 24 fra, så snart vi har et timetal større end 23: vi benytter såkaldt *modulo-regning*.

2.1 Modulo-regning og division

Lad a og n være to heltal, $n \neq 0$. Ved udtrykket "a modulo n", som også kaldes den *principale rest* af a ved division med n og skrives

$$a \bmod n,$$

forstår vi det mindste ikke-negative heltal r således at $r = a - qn$ for et eller andet heltal q. For positive tal a og n trækkes n altså fra a så mange gange det er muligt, uden at resultatet bliver et negativt tal – og dette resultat er så netop $a \bmod n$ (hvad sker der hvis a er negativ?). Vi kalder også r for "resten af a ved division med n", og der gælder at $0 \leq r < |n|$.

Der gælder følgende nyttige regneregler:
Sætning 1. *Lad a, b og n være vilkårlige heltal ($n \neq 0$).*

1. $a \bmod n = a \bmod (-n)$
2. $a+b \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$
3. $a \cdot b \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n$

Reglen (1) kan nemt forklares ved, at hvis $a \bmod n = r = a - qn$ for et eller andet q, da gælder $r = a - (-q) \cdot (-n)$, og derfor er r også lig $a \bmod (-n)$. Vi kan derfor i det følgende antage, at $n > 0$. De sidste to regler er ligeledes forholdsvis nemme at vise (se Opgave 15).

1. Følgende små opgaver løses i hånden, men løs dem også i Maple (man skriver `a mod n` for at beregne $a \bmod n$).

(a) Beregn $119 \bmod 13$, $-201 \bmod 12$, og $489 \bmod -61$

(b) Beregn $5^{717} \bmod 4$ (brug regnereglerne!)

I kender nok betydningen af udtrykket "a går op i b", men lad os præcisere det. At $a \neq 0$ går op i b, hvilket også skrives

$$a \,|\, b,$$

betyder, at der findes et heltal q således at $b = qa$. q behøver ikke være positiv. Bemærk, at $a \,|\, b \Leftrightarrow b \bmod a = 0$.

Der findes en række nyttige divisionsregler, som vi skal kigge på.
Sætning 2. *Lad a, b og c være vilkårlige heltal.*

1. *Hvis $a \,|\, b$ og $b \,|\, c$ så gælder $a \,|\, c$.*
2. *Hvis $a \,|\, b$ og $a \,|\, c$, så har vi $a \,|\, (sb + tc)$ for vilkårlige heltal s og t.*

3. *Hvis $ab \mid c$ så gælder både $a \mid c$ og $b \mid c$.*

Bevis. Vi viser (1). At $a \mid b$ betyder at der findes et heltal q_1 således at $b = q_1 a$. At $b \mid c$ betyder at der findes et heltal q_2 så $c = q_2 b$. Da $b = q_1 a$ har vi $c = q_2 q_1 a$, hvilket viser at $a \mid c$. □

2. Bevis på lignende måde de resterende to påstande i Sætning 2.

Når $a \bmod n = r$, går n op i $a - r$. Generelt siger vi, at a og b er *kongruente* modulo n hvis og kun hvis n går op i $a - b$. At a og b er kongruente modulo n skrives

$$a \equiv b \pmod{n}.$$

Når vi regner modulo n, er a og b så at sige repræsentanter for det samme tal. Der findes præcist én repræsentant i intervallet $[0, n-1]$, og det er $r = a \bmod n$.

Vi minder om, at hvis $a \equiv b \pmod{n}$, så findes der et heltal q_1 så $q_1 n = a - b$. Vi kan altså skrive $a = q_1 n + b$. Lad nu $a \bmod n = r$, dvs. der findes et heltal q_2 således at $a = q_2 n + r$, hvor $0 \leq r < n$. Idet vi nu har $a = q_1 n + b = q_2 n + r \Leftrightarrow b = (q_2 - q_1)n + r$ (hvor det stadig gælder at $0 \leq r < n$), har vi nu vist at $a \bmod n = b \bmod n$. Vi har altså vist '$\Rightarrow$'-delen af følgende sætning:
Sætning 3. *Lad a og b være vilkårlige heltal, og lad n være et positivt heltal. Da gælder*

$$a \equiv b \pmod{n} \Leftrightarrow a \bmod n = b \bmod n.$$

3. Vis '$\Leftarrow$'-delen af Sætning 3.

I e-noternes kapitel 1 nævnes begrebet *legeme*, som på engelsk kaldes *Field*, og i kapitel 7 defineres begrebet vektorrum. Senere i denne opgave, skal vi arbejde med mængden bestående af elementerne 0 og 1, og som er udstyret med kompositionerne addition og multiplikation modulo 2. Denne mængde med de anførte regneoperationer betegnes $\mathbb{F}_2$. Mængden af talsæt med n elementer fra $\mathbb{F}_2$, udstyret med regneoperationerne anført i e-noternes definition $1-2$, betegnes ${\mathbb{F}_2}^n$.

4. (a) Find ved at søge på internettet den matematiske definition på begrebet *legeme*, og bevis at $\mathbb{F}_2$ er et *legeme*.

(b) Bevis at ${\mathbb{F}_2}^n$ er et vektorrum.

I e-noterne oplyses, at man som legeme L kan vælge $\mathbb{R}$ eller $\mathbb{C}$. Foregående opgave viser, at man også kan vælge at sætte L lig med $\mathbb{F}_2$. Bemærk dog at et ligningssystem over $\mathbb{F}_2$ med endeligt mange variable altid har endeligt mange løsninger.

2.2 Største fælles divisor

Som nævnt i introduktionen kan ethvert heltal opløses i dets primfaktorer. Har vi to positive heltal a og b, kan vi betragte deres primfaktorer og se, hvilke de har til fælles. Produktet af disse kaldes *største fælles divisor* af a og b, og vi skriver $\text{sfd}(a,b)$. Tallet $\text{sfd}(a,b)$ er altså det største tal, som går op i både a og b. Hvis a og b ikke har nogen primfaktorer til fælles, er $\text{sfd}(a,b) = 1$. Man kan naturligvis finde $\text{sfd}(a,b)$ ved først at finde faktoriseringerne af a og b. Der findes dog en hurtigere metode, som kaldes *Euklids algoritme*. I har muligvis allerede stiftet bekendtskab med denne algoritme, men for en ordens skyld præsenterer vi den her som Algoritme 1. Vi skal

Input: Positive heltal a og b
Output: sfd(a,b)
1: $r_0 := a$ og $r_1 := b$
2: $i := 1$
3: **while** $r_i \neq 0$ **do**
4: $\quad r_{i+1} := r_{i-1} \bmod r_i$
5: $\quad i := i+1$
6: **end while**
7: **return** r_{i-1}

Algoritme 1: Euklids algoritme.

bruge Euklids algoritme senere. Vi får også senere brug for at kunne skrive *procedurer* i Maple, så derfor viser vi i følgende eksempel, hvordan Euklids algoritme kan implementeres som en Maple-procedure (kommandoen `proc`).

```
sfd := proc(a,b) local r0, r1, tmp;
 r0 := a;
 r1 := b;
 while r1 <> 0 do
  tmp := r0 mod r1;
  r0 := r1;
  r1 := tmp;
 end do:
 return r0;
end proc:
```

Proceduren gives navnet `sfd`, og den kan derfor kaldes med f.eks. `sfd(17,119)`. Variablene `a` og `b` er input til proceduren, og variablene `r0`, `r1` og `tmp` er lokale variable, som ikke skal gemmes efter proceduren er færdig. Da man ikke behøver at huske alle r_i'erne i Euklids algoritme, lader vi `r0` betegne den "næstsidste" værdi af r_i, og `r1` betegne den sidste værdi af r_i. I `while`-løkken sørges der for, at `r1` får den just beregnede værdi af r_i, og `r0` får den "gamle" værdi af r_i. Læs om `while`-løkker osv. i Maples hjælp.

Der gælder følgende nyttige sætning, som vi ikke vil bevise:
Sætning 4. *Lad a og b være positive heltal. Da findes heltal x og y således at*

$$\text{sfd}(a,b) = xa + yb.$$

Antag at et primtal p går op i produktet ab, og at p ikke går op i a. Det betyder at sfd$(p,a) = 1$, hvilket ifølge Sætning 4 betyder at der findes heltal x og y så

$$xp + ya = 1.$$

Ved at multiplicere med b fås

$$xpb + yab = b.$$

Da p går op i xpb og (ifølge antagelsen) i yab, og dermed i venstresiden ovenfor, går p også op i højresiden, som er b. Vi har nu vist følgende sætning.

Sætning 5 (Euklids lemma)**.** *Lad p være et primtal, og a og b heltal. Hvis $p \mid ab$, så gælder $p \mid a$ eller $p \mid b$.*

Bemærk, at hvis $1 < \text{sfd}(a,n) < n$, så er $\text{sfd}(a,n)$ en ikke-triviel faktor i n. De "trivielle" faktorer i n er 1 og n.

2.3 Kvadratiske rester

En *kvadratisk rest* modulo n er et heltal $a \not\equiv 0 \pmod{n}$, således at der findes et andet heltal x, så $x^2 \equiv a \pmod{n}$. De andre tal mellem 1 og $n-1$ kaldes kvadratiske ikke-rester. Tallet 0 er specielt, og er hverken en kvadratisk rest eller en kvadratisk ikke-rest. Et heltal x med egenskaben $x^2 \equiv a \pmod{n}$ kaldes en kvadratrod af a modulo n.

2.3.1 Kvadratiske rester modulo et primtal

Vi betragter først kvadratiske rester modulo et primtal. Husk at hvis $x^2 \equiv y^2 \pmod{p}$, så går p op i $x^2 - y^2$.

5. Antag nu at $x^2 \equiv y^2 \pmod{p}$.
Brug den velkendte regneregel for to tals sum gange de samme to tals differens, samt Sætning 5 til at vise, at hvis p er et primtal, så er $x \equiv \pm y \pmod{p}$.

Heraf ses, at enhver kvadratisk rest modulo et primtal p har præcist to kvadratrødder, x og $-x \pmod{p}$ – dog gælder specielt for primtallet $p = 2$ at $x \equiv -x \pmod{p}$.
Vi indfører nu det såkaldte *Legendre-symbol*.
Definition 1. Legendre-symbolet $\left(\frac{a}{p}\right)$, hvor p er et primtal større end 2, er defineret således:

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{hvis } a \equiv 0 \pmod{p} \\ 1 & \text{hvis } a \text{ er en kvadratisk rest modulo } p \\ -1 & \text{hvis } a \text{ er en kvadratisk ikke-rest modulo } p \end{cases}$$

Det viser sig at være relativt nemt at beregne Legendre-symbolet:
Sætning 6. *Legendre-symbolet $\left(\frac{a}{p}\right)$ kan beregnes som $a^{(p-1)/2} \bmod p$.*

Ved at udføre beregningen $a^{(p-1)/2} \bmod p$ finder man altså ud af, hvorvidt a er en kvadratisk rest modulo p.

6. Modulær eksponentiering, dvs. beregninger af formen $a^x \bmod n$, kan i Maple udføres effektivt ved hjælp af kommandoen `a&^x mod n`. Bemærk `&`-tegnet! Skriv et program som laver en liste indeholdende alle kvadratiske rester modulo primtallet 53.

2.3.2 Kvadratiske rester modulo et sammensat tal

Lad nu $n = pq$ være et produkt af de to forskellige primtal p og q, som vi først antager, at vi ikke kender. Bemærk at dette betyder at hvis $a \mid n$, så gælder $a \in \{1, p, q, n\}$ (vi ser her bort fra negative divisorer).

7. Lad x og y være givet således at $x^2 \equiv y^2 \pmod{n}$. Vis, at hvis $x \not\equiv \pm y \pmod{n}$, så er $\text{sfd}(x+y,n)$ en ikke-triviel faktor i n.

Opgave 7 giver en metode til at faktorisere et produkt n af to primtal: kender man to forskellige kvadratrødder x og y af samme tal a modulo n, og $x \not\equiv \pm y \pmod{n}$, så kan man finde en faktor i n ved at beregne $\text{sfd}(x+y,n)$.

Vi har ikke vist, at det kan lade sig gøre at to kvadratrødder x og y af en kvadratisk rest a kan have relationen $x \not\equiv \pm y \pmod{n}$. Det forholder sig imidlertid sådan, at hvis n er et produkt af to primtal, så har enhver kvadratisk rest a modulo n præcist fire forskellige kvadratrødder, forudsat at $\text{sfd}(a,n) = 1$. Bemærk, at hvis $\text{sfd}(a,n) > 1$ og $a \neq 0$, så har vi allerede faktoriseret n.

De fire kvadratrødder kan betegnes med $\pm r_1$ og $\pm r_2$. Lad os som eksempel kigge på kvadratrødderne af 9 modulo 55 (som jo er lig $5 \cdot 11$). Det er klart, at 3 og $-3 \equiv 52 \pmod{55}$ er to af kvadratrødderne. De andre to er 8 og $-8 \equiv 47 \pmod{55}$: vi har $(\pm 8)^2 = 64 \equiv 9 \pmod{55}$.

Dette viser, at det *kan* lade sig gøre for to kvadratrødder x og y af en kvadratisk rest a modulo n at have relationen $x \not\equiv \pm y \pmod{n}$. Ovenfor kunne vi f.eks. vælge $x = 3$ og $y = 8$. Vi kan nu som vist i Opgave 7 finde en ikke-triviel faktor i n ved at beregne $\text{sfd}(x+y,n)$. I eksemplet får vi $\text{sfd}(3+8,55) = 11$.

Det er dog i almindelighed svært at finde to kvadratrødder med de givne betingelser, også selvom vi selv kan vælge den kvadratiske rest. Vi kan vælge x, sætte $a = x^2$, og så søge efter en anden kvadratrod af a. Det nytter ikke at vælge $-x$ (som jo ellers også er en kvadratrod), da vi så ikke har to kvadratrødder med den påkrævede indbyrdes relation.

Hvis n (mere generelt) er et produkt af $t > 2$ forskellige primtal, eller hvis et primtal går op i n flere gange, er situationen lidt mere kompliceret. Vi vil i det følgende antage, at n har præcist to forskellige primfaktorer, da dette almindeligvis er det vanskeligste tilfælde.

3 Dixons tilfældige kvadraters faktoriseringsmetode

Vi er nu klar til at beskrive Dixons faktoriseringsmetode, der benytter teknikken beskrevet i det foregående afsnit. Opgaven er altså at finde en kvadratisk rest a og to kvadratrødder x og y af a modulo et produkt n af to primtal, således at $x \not\equiv \pm y \pmod{n}$. Vi bemærker, at hvis vi tilfældigt og uafhængigt er i stand til at vælge to forskellige kvadratrødder af en kvadratisk rest modulo n, så kan vi faktorisere n med sandsynlighed $1/2$ (overvej dette!).

3.1 Om at finde to kvadratrødder af en kvadratisk rest

Vi indfører begrebet *faktorbase*.
Definition 2. Faktorbasen $\mathcal{B}_T$ er mængden af primtal mindre end eller lig T, forenet med mængden $\{-1\}$. Vi lader i øvrigt t benævne antallet af elementer i $\mathcal{B}_T$. Bemærk, at når T er fastlagt, er t det også.

Vi har f.eks. $\mathcal{B}_{19} = \{-1,2,3,5,7,11,13,17,19\}$.

Definition 3. Et heltal a siges at være $\mathcal{B}_T$-smooth hvis alle a's primfaktorer ligger i $\mathcal{B}_T$. Tallet a må godt være negativt, og derfor er -1 inkluderet i $\mathcal{B}_T$, selvom -1 ikke er et primtal.

Tallet -2394 er f.eks. $\mathcal{B}_{19}$-smooth, da $-2394 = -1 \cdot 2 \cdot 3^2 \cdot 7 \cdot 19$. Tallet $2 \cdot 23 = 46$ er ikke $\mathcal{B}_{19}$-smooth.

En metode til at finde to kvadratrødder af en kvadratisk rest modulo n er som følger. Vælg flere forskellige værdier x_i, $1 \leq i \leq s$, og sæt

$$z_i = x_i^2 \bmod n.$$

Hvis alle tallene z_i kan faktoriseres over en lille faktorbase $\mathcal{B}_T$, så kan vi håbe på at vi kan finde en ikke-tom delmængde $S \subseteq \{1, \ldots, s\}$ således at produktet $\prod_{i \in S} z_i$ kan opløses i *lige* potenser af tallene i faktorbasen $\mathcal{B}_T$. Med andre ord: hvis p_i er det i'te element i $\mathcal{B}_T$, så håber vi på at vi kan finde en mængde S således at

$$\prod_{i \in S} z_i = \prod_{i=1}^{t} p_i^{e_i},$$

hvor alle e_i'erne er lige tal. Hvis dette lykkes, kan vi sætte

$$X = \prod_{i \in S} x_i \bmod n$$

og

$$Y = \prod_{i=1}^{t} p_i^{e_i/2} \bmod n.$$

Bemærk at alle eksponenterne er heltal når e_i'erne er lige. Herved får vi $X^2 \equiv Y^2 \pmod{n}$ (overvej dette!), og sandsynligheden for at $X \not\equiv \pm Y \pmod{n}$ er cirka $1/2$. Vi har vist brug for et eksempel.

3.1.1 Eksempel

Lad $n = 133897$ (et produkt af to forskellige primtal), og betragt følgende kvadratiske rester modulo n:

$$\begin{aligned}
358^2 &\equiv (-1)^1 \cdot 2^0 \cdot 3^2 \cdot 5^0 \cdot 7^2 \cdot 11^0 \cdot 13^1 \cdot 17^0 \cdot 19^0 \pmod{n} \\
359^2 &\equiv (-1)^1 \cdot 2^3 \cdot 3^1 \cdot 5^0 \cdot 7^0 \cdot 11^1 \cdot 13^0 \cdot 17^0 \cdot 19^1 \pmod{n} \\
363^2 &\equiv (-1)^1 \cdot 2^4 \cdot 3^0 \cdot 5^0 \cdot 7^1 \cdot 11^0 \cdot 13^0 \cdot 17^0 \cdot 19^1 \pmod{n} \\
365^2 &\equiv (-1)^1 \cdot 2^5 \cdot 3^1 \cdot 5^0 \cdot 7^1 \cdot 11^0 \cdot 13^0 \cdot 17^0 \cdot 19^0 \pmod{n} \\
367^2 &\equiv (-1)^0 \cdot 2^3 \cdot 3^2 \cdot 5^0 \cdot 7^0 \cdot 11^1 \cdot 13^0 \cdot 17^0 \cdot 19^0 \pmod{n} \\
370^2 &\equiv (-1)^0 \cdot 2^0 \cdot 3^1 \cdot 5^0 \cdot 7^1 \cdot 11^1 \cdot 13^1 \cdot 17^0 \cdot 19^0 \pmod{n} \\
381^2 &\equiv (-1)^0 \cdot 2^{10} \cdot 3^0 \cdot 5^0 \cdot 7^0 \cdot 11^1 \cdot 13^0 \cdot 17^0 \cdot 19^0 \pmod{n} \\
397^2 &\equiv (-1)^0 \cdot 2^5 \cdot 3^1 \cdot 5^0 \cdot 7^0 \cdot 11^0 \cdot 13^1 \cdot 17^0 \cdot 19^1 \pmod{n}
\end{aligned}$$

På højresiden har vi faktoriseret den kvadratiske rest, og vi ser at alle de valgte kvadratiske rester er $\mathcal{B}_{19}$-smooth. Vi har faktoriseret på en sådan måde, at hvis x^2 er tættere på n end på 0, så faktoriserer vi i stedet $x^2 - n$ (som jo er kongruent med x^2 modulo n), og vi har angivet

eksponenterne til alle elementerne i faktorbasen. Hvordan vi finder frem til disse kvadratiske rester ser vi på senere. Men lad os nu opskrive eksponenterne i en matrix M således:

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 4 & 5 & 3 & 0 & 10 & 5 \\ 2 & 1 & 0 & 1 & 2 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Hver række hører til et element i faktorbasen $\mathcal{B}_{19}$, og hver søjle hører til en kvadratisk rest. Vi reducerer nu alle tallene modulo 2, hvilket betyder at de ulige tal bliver til 1, og de lige tal bliver til 0. Så får vi følgende matrix:

$$\tilde{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Om et øjeblik bliver det klart, hvorfor vi gør sådan. Lad os nu finde nulrummet for $\tilde{M}$, f.eks. ved hjælp af Gauss-elimination. Alle beregninger undervejs foretages modulo 2, så vi har f.eks. $0-1=1$. Da matricen har størrelsen 9×8 og to af rækkerne er nulrækker, er matricens rang mindre end dimensionen af det tilsvarende definitionsrum. Ifølge dimensionssætningen indeholder nulrummet for den lineære afbildning med $\tilde{M}$ som afbildningsmatrix egentlige vektorer. Bemærk, at når vi regner modulo 2 har vi altid et endeligt antal vektorer i nulrummet.

8. (a) Find, ved håndregning eller simuleret håndregning modulo 2, trappeformen og rangen af $\tilde{M}$ (Gauss - Jordan elimintion modulo 2

(b) Brug den fundne trappeform til at finde kernen (nulrummet) for den lineære afbildning med $\tilde{M}$ som afbildningsmatrix, idet beregningerne foretages modulo 2.

(c) Maple indeholder en særlig pakke kaldet `LinearAlgebra[Modular]`, beregnet til moduloregning i lineær algebra. Kontrollér den fundne trappeform og kerne med kommandoerne `RowReduce` og `Basis` fra denne Maplepakke (Læs evt. mere om kommandoerne i Maples hjælp).

I opgaven ovenfor fandt I at en basis for kernen består af vektorerne $[1,0,0,1,1,1,0,0]^T$ og $[1,1,0,0,0,0,1,1]^T$. Bemærk at det betyder at kernen indeholder netop tre egentlige vektorer. Lad os se på vektoren $[1,1,0,0,0,0,1,1]^T$. At denne vektor ligger i nulrummet for $\tilde{M}$ betyder nemlig, at hvis vi multiplicerer 1., 2., 7. og 8. kvadratiske rest i listen ovenfor (både venstre og

højre side), så er alle eksponenter til tallene i faktorbasen lige, og dermed har vi en kvadratisk rest modulo n, hvor vi kender to kvadratrødder. Vi får

$$\begin{aligned} 358^2 \cdot 359^2 \cdot 381^2 \cdot 397^2 &\equiv (-1)^2 \cdot 2^{18} \cdot 3^4 \cdot 7^2 \cdot 11^2 \cdot 13^2 \cdot 19^2 \pmod{n} \Leftrightarrow \\ 19439852154^2 &\equiv (-87639552)^2 \pmod{n} \Leftrightarrow \\ 16209^2 &\equiv 62983^2 \pmod{n}. \end{aligned}$$

Da vi nu har $X = 16209$ og $Y = 62983$, og vi ser at $X \not\equiv \pm Y \pmod{n}$, så kan vi faktorisere n ved at beregne $\text{sfd}(X+Y,n) = \text{sfd}(16209+62983, 133897)$. I Afsnit 2.2 beskrev vi en procedure til beregning af største fælles divisor, som kan benyttes her. Man skulle i dette tilfælde gerne komme frem til resultatet 521. Når man dividerer 521 op i 133897 får man 257, og 133897 faktoriseres derfor som $521 \cdot 257$.

3.2 Ikke helt tilfældige kvadrater

Hvordan valgte vi de kvadratiske rester således at de er $\mathcal{B}_T$-smooth for et lille T? Det er klart, at der er større chance for at et lille heltal (absolut set) er $\mathcal{B}_T$-smooth end at et stort heltal er det.

Hvis vi sætter $m = \lfloor \sqrt{n} \rfloor$ (den heltallige værdi af $\sqrt{n}$, Maplekode `floor`), så får vi $m^2 \approx n$, og vi har derfor at $m^2 - n$ er et (negativt) tal relativt tæt på nul. Vi kan generelt betragte polynomiet

$$Q(x) = (x+m)^2 - n$$

for små heltalsværdier af x (også negative). Bemærk, at $Q(x)$ er en kvadratisk rest modulo n med kvadratrod $x+m$ for ethvert heltal x. Ved at ekspandere får vi $Q(x) = x^2 + 2mx + m^2 - n$, hvor $m^2 - n$ jo er tæt på nul. For x-værdier tæt på nul er $x^2 + 2mx + m^2 - n \approx x^2 + 2mx$ altså et relativt lille tal, og der er derfor en god chance for, at det faktoriserer over en lille faktorbase.

Nu har vi altså en kvadratisk rest $Q(x) \equiv (x+m)^2 \pmod{n}$, som med god sandsynlighed faktoriserer over en lille faktorbase. Hvis vi kan finde flere sådanne kvadratiske rester (ved at prøve flere forskellige værdier af x), så kan vi benytte lineær algebra til at finde et produkt af kvadratiske rester, som kan faktoriseres over en lille faktorbase på en sådan måde, at alle eksponenter til tallene i faktorbasen er lige. Når alle eksponenter er lige, kan vi let beregne en kvadratrod ved at dividere eksponenterne med 2. Nu kan vi håbe, at denne kvadratrod har den påkrævede relation til kvadratroden som beregnes ud fra venstresiderne. I eksemplet ovenfor fandt vi netop de kvadratiske rester ved at prøve forskellige heltalsværdier af x i polynomiet $Q(x)$.

Det er en god idé at finde lidt flere kvadratiske rester der faktoriserer over faktorbasen end antallet af elementer i faktorbasen, således at nulrummet indeholder flere vektorer. Dette giver flere par (X,Y) som hver især med sandsynlighed cirka $1/2$ kan have relationen $X \not\equiv \pm Y \pmod{n}$.

3.3 Implementation af faktoriseringsmetoden i Maple

Vi kan nu implementere Dixons faktoriseringsmetode i Maple. Vi tager til at starte med små skridt ad gangen.

9. Skriv en Maple-procedure `Factorbase` som givet et tal T danner faktorbasen $\mathcal{B}_T$ som en liste i Maple. Maple-kommandoerne `isprime` og/eller `nextprime` kan være nyttige. Bemærk desuden, at man kan indsætte et nyt element `x` i en eksisterende liste `B` med kommandoen `B := [op(B), x];`. Antallet af elementer i `B` kan i øvrigt findes med `nops(B)`.

Når faktorbasen $\mathcal{B}_T$ er dannet, kan man begynde at beregne $Q(x)$ for forskellige små heltalsværdier af x, og undersøge om $Q(x)$ er $\mathcal{B}_T$-smooth.

10. Skriv en Maple-procedure `ExponentVector` som givet et heltal $Q(x)$ og en faktorbase $\mathcal{B}_T$ returnerer en søjlevektor med alle eksponenterne til elementerne i $\mathcal{B}_T$ hvis $Q(x)$ er $\mathcal{B}_T$-smooth. Hvis $Q(x)$ ikke er $\mathcal{B}_T$-smooth, returneres f.eks. en `NULL`-værdi.

Vink til ovenstående opgave: Bemærk at et helt tal s går op i et helt tal t netop når t mod $s = 0$. En s gange t matrix, hvor alle elementerne er nuller oprettes med Maplekommandoen `Matrix(s,t)`.

Vi mangler nu blot at sætte ovenstående procedurer sammen og benytte dem til at danne den matrix, som vi i eksemplet ovenfor kaldte M, og dernæst finde nulrummet for matricen $\tilde{M} = M \bmod 2$, og til slut beregne X og Y. Bemærk, at man er nødt til at huske på hvilke kvadratrødder, der hører til hver søjle i matricen M, for disse skal man bruge når man skal beregne X.

I Maple kan man finde en basis for nulrummet for en matrix M modulo 2 ved hjælp af kommandoen

```
LinearAlgebra[Modular][Basis](2,M,row,false,column);
```

Denne kommando er en del af den tidligere omtalte "datter"-pakke `Modular` til `LinearAlgebra`-pakken.

11. Lav nu i Maple en fuld implementation af Dixons faktoriseringsmetode. Det er i orden, hvis faktoriseringsmetoden fejler med sandsynlighed omkring $1/2$. Sørg for, at det er nemt at ændre størrelsen på faktorbasen via parameteren T. Bemærk, at man kan indsætte en ny søjlematrix `v` i en eksisterende matrix `M` med kommandoen `M := Matrix([M, v]);`. Endvidere kan man slette en søjle j fra en matrix `M` med kommandoen `M := DeleteColumn(M,[j]);`. Kan jeres program faktorisere tallet $n = 23058430274673049 93$ nævnt i introduktionen? Kan det faktorisere endnu større produkter af to primtal? (Vent eventuelt med at prøve dette indtil I har læst og implementeret idéerne i næste afsnit!)

3.4 Optimering

Vi kigger nu på nogle muligheder for at forbedre effektiviteten af den beskrevne faktoriseringsmetode.

3.4.1 Overflødige elementer i faktorbasen

Vi så i eksemplet ovenfor, at to rækker i matricen M var nul-rækker. Dette betyder i eksemplet, at tallene 5 og 17 aldrig optrådte som faktorer i de kvadratiske rester. Vi vil nu undersøge hvorfor.

De kvadratiske rester blev beregnet som $Q(x) = (x+m)^2 - n$ for forskellige værdier af x. Hvis primtallet p går op i $Q(x)$, så har vi at $Q(x) \equiv 0 \pmod{p}$.

12. Brug sammenhængen $Q(x) \equiv 0 \pmod{p}$ til at vise, at n er en kvadratisk rest modulo p. Hvis p er faktor i den kvadratiske rest $Q(x) \bmod n$, så er n altså en kvadratisk rest modulo p. Forklar nu, hvorfor 5 og 17 aldrig optrådte som faktorer i de kvadratiske rester i eksemplet ovenfor (her kan en generalisering af jeres program fra Opgave 6 være nyttig).

Vi behøver altså kun at have et primtal p med i faktorbasen, hvis n er en kvadratisk rest modulo p. For $p > 2$ er kriteriet altså at $n^{(p-1)/2} \bmod p = 1$. Desuden bør primtallet 2 altid være med i faktorbasen, da vi kan antage at n er ulige, og da er n en kvadratisk rest modulo 2. Alle andre tal (på nær -1) kan udelades.

3.4.2 Størrelsen af faktorbasen

For et givet tal n som skal faktoriseres, skal man vælge størrelsen t af faktorbasen angivet ved det største element T. Dette valg er lidt af en videnskab for sig; jo flere elementer, der er med i faktorbasen, jo lettere er det at finde kvadratiske rester, som kan faktoriseres over faktorbasen. På den anden side, jo større faktorbasen er, jo flere smooth kvadratiske rester skal vi finde for at have nok søjler i matricen M.

Det viser sig, at det er bedre at vælge faktorbasen lidt for stor end lidt for lille. For et n på ca. 20 cifre bør T f.eks. være mindst 500.

13. Eksperimentér med forskellige størrelser af faktorbasen. Diskutér resultatet af jeres eksperimenter. Brug evt. Maples `time`-kommando til at teste med.

4 Tid tilovers?

De grupper, der har tid tilovers, og som er særligt ambitiøse, kan give sig i kast med én eller flere af følgende udfordringer.

14. Vis at ethvert heltal større end 1 på præcist én måde (når der ses bort fra rækkefølgen) kan skrives som et produkt af primtal. Vink: Benyt Euklids lemma.

15. Vis at hvis $r = a \bmod n$, så gælder $0 \leq r < |n|$. Vis også reglerne 2 og 3 i Sætning 1.

16. Euklids algoritme (Algoritme 1) stopper, når $r_i = 0$. Gør rede for at $r_{i-1} = \text{sfd}(a,b)$.
Vink: Vis først, at r_{i-1} går op i alle resterne r_j, $0 \leq j \leq i-1$, dvs. specielt er r_{i-1} en fælles divisor af a og b. Vis dernæst, at enhver fælles divisor c af a og b går op i alle resterne r_j, $0 \leq j \leq i-1$. Benyt til slut disse to oplysninger til at bevise at $r_{i-1} = \text{sfd}(a,b)$.

17. Vis Sætning 4.