Priemgetallen

van nutteloos tot staatsgevaarlijk?

Wieb Bosma

Nijmeegse Tweedaagse
Radboud Universiteit

Nijmegen
oktober 2008

Priemgetallen

Voorwoord

Dit zijn de aantekeningen bij één van de twee onderwerpen uit de **Nijmeegse Wiskunde Tweedaagse** van oktober 2008. Het onderwerp, priemgetallen, vormt aan de ene kant een klassiek voorbeeld uit de getaltheorie, en is aan de andere kant van belang voor moderne toepassingen. Je kunt hier zowel iets vinden over de manier waarop wiskundigen al eeuwen over priemgetallen redeneren, als over de wijze waarop, vooral sinds de komst van de computer, met priemgetallen gerekend wordt in toepassingen. Natuurlijk laten tijd en ruimte niet meer toe dan het laten zien van het topje van een ijsberg, maar hopelijk krijg je iets te zien van de wiskundige methode en avn de fascinatie van wiskundigen voor dit onderwerp.

Wieb Bosma, Nijmegen, oktober 2008.

Priemgetallen

1 Priemgetallen

1.1 Wat is een priemgetal

In deze aantekeningen gaat het voornamelijk over de natuurlijke getallen $\mathbb{N} = \{0, 1, 2, \ldots\}$. Zulke natuurlijke getallen, die we met d, e, k, m, n, p, q enz., aangeven, zijn gemaakt door herhaaldelijk 1 bij zichzelf op te tellen: $2 = 1 + 1$, $3 = 2 + 1 = 1 + 1 + 1$ enzovoorts. Maar je kunt ze ook met elkaar vermenigvuldigen: $m \times n$ of $m \cdot n$, en het resultaat is ook altijd weer een natuurlijk getal. De merkwaardige verschijning van priemgetallen vindt zijn oorzaak in het feit dat je *niet* alle natuurlijke getallen krijgt door herhaalde vermenigvuldiging met één natuurlijk getal (of met enkele natuurlijke getallen).

Wanneer je twee natuurlijk getallen van elkaar aftrekt, is het resultaat $m - n$ niet altijd een natuurlijk getal, maar het zit wel altijd in de gehele getallen, $\mathbb{Z}$. Als je twee natuurlijke getallen op elkaar probeert te delen, is het resultaat m/n niet altijd weer een natuurlijk getal. Voor ons is van belang wanneer dat wél zo is.

1.1.1 Definities. We zeggen dat d een *deler* is van n, of dat d het getal n *deelt*, wanneer er een natuurlijk getal k is zodat $k \cdot d = n$. In dat geval heet n ook wel een *veelvoud* van d, en heten k en d ook *factoren* van n.

Volgens deze definitie heeft elk natuurlijk getal n in ieder geval twee delers, namelijk 1 en n (alleen voor $n = 1$ zijn die twee gelijk!). Nul is deelbaar door elk natuurlijk getal.

1.1.2 Definities. Een natuurlijk getal n heet een *priemgetal* wanneer $n > 1$ is en 1 en n de enige delers van n zijn. We zeggen dan ook wel: *n is priem.* Als $n > 1$ is en niet priem, dan heet n *samengesteld.* Het getal 1 is niet priem en ook niet samengesteld.

1.1.3 Stelling. *Elk natuurlijk getal $n > 1$ is te schrijven als product van eindig veel priemgetallen: $n = p_1 \cdot p_2 \cdots p_t$. Bovendien is deze priemfactorontbinding uniek als we opleggen dat $p_1 \leq p_2 \leq \cdots \leq p_t$.*

1.1.4 Opmerkingen. Kijk als voorbeeld eens naar $n = 12$. Natuurlijk kun je n op verschillende manieren *in factoren ontbinden*, bijvoorbeeld $12 = 4 \cdot 3$ of $12 = 1 \cdot 6 \cdot 2$. Maar 4 en 1 en 6 zijn niet priem, en als we ons tot priemgetallen beperken blijven alleen $12 = 2 \cdot 2 \cdot 3$ en $12 = 2 \cdot 3 \cdot 2$ en $12 = 3 \cdot 2 \cdot 2$ over. Slechts één mogelijkheid blijft over

als we eisen dat de priemgetallen oplopen (eigenlijk: niet àflopen) in grootte, namelijk $12 = 2 \cdot 2 \cdot 3$, en dat schrijven we meestal als $2^2 \cdot 3$.

Op grond van deze stelling, die we natuurlijk nog wel moeten bewijzen, kunnen we dus spreken van *de priemfactorontbinding* van een natuurlijk getal.

Voor $n = 1$ er helemaal geen priemfactor van n, en zouden we de priemfactorontbinding kunnen lezen als product van $t = 0$ termen, wat per definitie 1 oplevert.

Er zijn verschillende manieren om deze stelling te bewijzen. Het bewijs dat ik hier geef is tamelijk eenvoudig, en maakt eigenlijk alleen gebruik van een resultaat dat ik eerst afzonderlijk formuleer (en bewijs).

1.1.5 Lemma. *Laat $n > m$ zijn. Als d een deler is van n èn van m dan is d ook een deler van $n - m$. Als d een deler is van m maar niet van n (of omgekeerd), dan is d geen deler van $n - m$.*

Bewijs De eerst bewering volgt uit:

$$n - m = h \cdot d - j \cdot d = (h - j) \cdot d.$$

Veronderstel, voor de tweede bewering, dat $m = j \cdot d$, dat d geen deler is van n, maar dat d wél een deler is van het natuurlijke getal $v = n - m$, dus er is een k zodat $v = k \cdot d$. Dan is

$$n = (n - m) + m = v + m = k \cdot d + j \cdot d = (k + j) \cdot d$$

dus is n wél deelbaar door d. Dit is in tegenspraak met het gegevene. De veronderstelling dat d een deler is van v kan dus niet juist zijn.

Het ander geval (d deelt wel n maar niet m) gaat net zo.

Bewijs (van de Stelling) Veronderstel eens dat er getallen zijn die verschillende ontbindingen in priemfactoren hebben. Eén van die getallen met die eigenschap moet de kleinste zijn; die noemen we n. Veronderstel verder dat $n = p_1 \cdot p_2 \cdots p_s$ en $n = q_1 \cdot q_2 \cdots q_t$ ontbindingen in priemfactoren in niet-dalende volgorde zijn.

Als $s = 1$ of $t = 1$ dan is n priem en kan er maar 1 ontbinding zijn.

Het kan ook niet zo zijn dat voor zekere i en j geldt dat $p_i = q_j$, want anders zou $n/p_i = n/q_j$ een natuurlijk getal kleiner dan n zijn met twee verschillende priemfactorontbindingen, in tegenspraak met de aanname dat n de kleinste was.

Laten we aannemen dat $p_1 < q_1$ (het geval $p_1 > q_1$ gaat net zo). Bekijk nu het natuurlijke getal $m = (q_1 - p_1) \cdot q_2 \cdots q_t$. Vanwege $q_1 - p_1 < q_1$ is $m < n$, en dus is m op een unieke manier als product van priemfactoren te schrijven. Maar

$$m = (q_1 - p_1) \cdot q_2 \cdots q_t = n - p_1 \cdot q_2 \cdots q_t = p_1(p_2 \cdot p_3 \cdots p_s - q_2 \cdots q_t).$$

Door de twee factoren tussen haakjes beide in priemfactoren te ontbinden krijgen we twee priemfactorontbindingen voor m, die verschillend zijn omdat p_1 geen deler is van $q_1 - p_1$ (volgens het lemma!), en dus wel rechts maar niet links onder de priemfactoren voorkomt.

Dat is een tegenspraak, dus de veronderstelling (dat er getallen met verschillende priemfactorontbindingen bestaan) is fout.

De stelling geeft dus het belang aan van priemgetallen; ze vormen de bouwstenen voor alle natuurlijke getallen onder vermenigvuldiging.

♣ **Opgave 1.** *Je kunt aan een getal eenvoudig zien of het deelbaar is door 2, of door 5, of door 10. Hoe zit het met deelbaarheid door 4? En door 3, 9, 11? Kun je elk van die beweringen ook bewijzen?*

♣ **Opgave 2.** *Om te laten zien dat er echt iets te bewijzen valt over unieke ontbinding in priemfactoren, kun je kijken naar de verzameling E van even natuurlijk getallen. Daarin valt 10 niet te onbinden, maar 12 wel. Laat* e-priemen *de getallen uit E zijn die niet te ontbinden zijn in E. Vind een getal in E met twee verschillende ontbindingen in e-priemen en laat daarmee zien dat er geen unieke e-priemfactorontbinding in E bestaat.*

♣ **Opgave 3.** *We zagen dat je na het sorteren van de priemgetallen, natuurlijke getallen op maar één manier als product van priemgetallen kunt schrijven. Je kunt n op 2^{n-1} manieren als som van natuurlijke getallen schrijven, maar als je de summanden sorteert kan dat op $p(n)$ manieren, waar p de partitie-functie heet. Bereken $p(2), p(3), p(4), p(5), p(6)$.*

1.2 Delen met rest

Als n niet deelbaar is door m, blijft er een rest over.

1.2.1 Stelling. *Bij elk tweetal natuurlijke getallen $n \geq m$ bestaan natuurlijke getallen q en r zodat $n = q \cdot m + r$, en die zijn uniek als we voor de rest r eisen dat $0 \leq r < m$.*

Bewijs Het idee van een bewijs is heel simpel: omdat $m \leq n$, kunnen we m aftrekken van n en een natuurlijk getal of 0 als rest overhouden; dit kunnen we herhalen als de rest groter dan of gelijk is aan m, en het stopt zodra de rest kleiner dan m is. Het aantal keren dat je m van n hebt afgetrokken noem je q, en de rest die overblijft r.

De eigenschap dat d een deler is van n betekent dus precies dat de deling van n door d rest 0 oplevert. Deling met rest speelt een grote rol bij het bepalen van grootste gemene delers.

1.2.2 Definitie. Laten n en m natuurlijke getallen zijn. Een *gemene deler* van n en m is een getal d dat zowel n als m deelt. De *grootste gemene deler* van n en m geven we aan met $\text{ggd}(n, m)$.

1.2.3 Opmerkingen. Het is duidelijk dat de grootste gemene deler bestaat en uniek is: elk paar natuurlijke getallen heeft in ieder geval 1 als gemene deler, en omdat elk getal maar eindig veel delers heeft, zijn er ook maar eindig veel gemene delers, waaronder er één de grootste is.

Het is ook vrij eenvoudig in te zien dat de grootste gemene deler g van n en m niet alleen groter is dan alle andere gemene delers, maar zelfs een veelvoud van alle andere gemene delers.

Een andere belangrijke opmerking is dat elke gemene deler van n en m (en dus de grootste gemene deler) het verschil van n en m moet delen (zoals we zagen in Lemma 1.1.5), dus $n - m$ (neem even aan dat n de grootste van de twee is), en dan ook $n - 2 \cdot m$, enzovoorts, tot en met $n - q \cdot m = r$.

1.2.4 Gevolg. *De grootste gemene deler van $k + 1$ en k is 1.*

Bewijs Volgens de laatste opmerking moet de grootste gemene deler het verschil $(k + 1) - k = 1$ delen.

1.2.5 Voorbeeld. Om de ggd van 462 en 315 te berekenen zou je eerst alle delers van 462 kunnen bepalen:

$$1, 2, 3, 6, 7, 11, 14, 21, 22, 33, 42, 66, 77, 154, 231, 462$$

en die van 315

$$1, 3, 5, 7, 9, 15, 21, 35, 45, 63, 105, 315,$$

en dan van de gemeenschappelijke delers $1, 3, 7, 21$ de grootste, dus 21 in dit geval.

Ook zou je $462 = 2 \cdot 3 \cdot 7 \cdot 11$ en $315 = 3^2 \cdot 5 \cdot 7$ in priemfactoren kunnen ontbinden, en opmerken dat de grootste gemene deler product moet zijn uit de grootste priemmachten die in beide voorkomen, dus van 3 en 7.

Maar de beste methode (in termen van complexiteit, zie verderop) is de methode van *Euclides*: hiervoor doe je steeds deling met rest op twee getallen. Je begint met $n = 462$ en $m = 315$ (de grootste voorop), dus $n = q \cdot m + r$, vervangt dan n door m en m door de rest r. Hier dus: $462 = 1 \cdot 315 + 147$, en daarna $315 = 2 \cdot 147 + 21$, en tenslotte $147 = 7 \cdot 21 + 0$. Je stopt zodra je (zoals hier) als rest 0 overhoudt, de waarde van m (hier 21) is dan de ggd. Dit werkt volgens het principe van de laatste opmerking.

Dat het zo efficiënt werkt, komt doordat je in tegenstelling tot de eerste twee methoden n en m niet eerst in factoren hoeft te ontbinden (want om alle delers op te schrijven moet je ook alle priemfactoren vinden).

Een aardige eigenschap die niet direct volgt uit het bovenstaande, maar ook niet moeilijk is af te leiden, staat in de volgende stelling; hierin zullen de getallen x en y in het algemeen uit de verzameling van gehele getallen $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ komen. Bovendien is de methode van Euclides eenvoudig zo aan te passen dat deze niet alleen de grootste gemene deler maar ook de x en y oplevert.

1.2.6 Stelling. *De grootste gemene deler van n en m is het kleinste positieve getal dat is te schrijven als $x \cdot n + y \cdot m$.*

Een gevolg hiervan is weer de volgende nuttige eigenschap.

1.2.7 Stelling. *Als p een priemgetal is, en p is een deler van het product $n \cdot m$, dan is p een deler van n of van m.*

Bewijs Als p geen deler is van n, dan is $\text{ggd}(p, n) = 1$, want p heeft geen andere delers. Maar dan kun je getallen x en y vinden zodat $x \cdot p + y \cdot n = 1$. Vermenigvuldig beide zijden met m, dan vinden we $x \cdot p \cdot m + y \cdot n \cdot m = m$. Maar p is een deler van $n \cdot m$ en daarom van $x \cdot m \cdot n$, en natuurlijk ook van $y \cdot p \cdot m$. Dan is p ook een deler van hun som m.

Net zo volgt uit de aanname dat p geen deler is van m dat p wel n moet delen.

Let op dat de stelling niet geldt als p niet priem is: 15 deelt het product van 6 en 10, maar 15 deelt 6 noch 10.

♣ **Opgave 4.** *Wat is* $\text{ggd}(k + 2, k)$*? En* $\text{ggd}(k + 4, k)$*?*

♣ **Opgave 5.** *Vind x en y zodat $x \cdot 462 + y \cdot 315 = 21$.*

♣ **Opgave 6.** *Wat is* $\text{ggd}(463, 31)$*? En kun je hierbij ook x, y vinden als bedoeld in Stelling 1.2.6?*

1.3 Hoeveel priemgetallen zijn er

Het is heel natuurlijk je af te vragen *hoeveel* priemgetallen er zijn. Wanneer je begint te tellen vind je al gauw de vier priemgetallen kleiner dan 10, (namelijk $2, 3, 5, 7$) en de 25 priemgetallen kleiner dan 100. In het tabelletje wordt dat rijtje aantallen voortgezet tot 10^{21}; met $\pi(x)$ wordt het aantal priemgetallen kleiner dan of gelijk aan x aangegeven.

x	$\pi(x)$
10^1	4
10^2	25
10^3	168
10^4	1229
10^6	78498
10^{12}	37607912018
10^{18}	24739954287740860
10^{21}	21127269486018731928

Het maken van een lijst doe je met een *zeefmethode*, zoals die van *Eratosthenes*. De zeef van Eratosthenes werkt als volgt.

Schrijf, om de priemgetallen tot B te vinden, de getallen van 2 tot en met B op. Begin met 2 en streep daarna elk tweede getal door. Neem vervolgens het eerste niet doorgestreepte getal (3), en vanaf daar streep je elk veelvoud daarvan door. Herhaal dit (met 5) enzovoorts. De niet doorgetreepte getallen zijn de priemgetallen.

1.3.1 Voorbeeld. Met $B = 40$ strepen we uit de lijst getallen van 2 tot en met 40 eerst alle even getallen door, daarna de veelvouden $3, 6, 9, 12, 15, 18, 21, 24, 27, 30, 33, 36, 39$ van 3 (sommige waren ook al even natuurlijk), dan de veelvouden van 5, namelijk $10, 15, 20, 25, 30, 35, 40$ en we zijn klaar! Na $2, 3, 5$ zijn $7, 11, 13, 17, 19, 23, 29, 31, 37$ niet doorgestreept, dus priem.

Als je goed oplet, zie je in het voorbeeld dat je al klaar bent nadat je de veelvouden van 5 hebt weggestreept. In het algemeen kun je ophouden zodra je tot $\sqrt{B}$ bent gegaan, op grond van de volgende stelling.

1.3.2 Stelling. *Een natuurlijk getal $n > 1$ is een priemgetal dan en slechts dan als n geen deler groter dan 1 heeft die kleiner dan of gelijk is aan $\sqrt{n}$.*

Bewijs Als $n > 2$ een priemgetal is dan heeft n geen enkele andere deler dan 1 die kleiner dan of gelijk is aan $n - 1$, laat staan kleiner of gelijk $\sqrt{n}$.

Laat, voor de omkering, n geen delers kleiner dan of gelijk $\sqrt{n}$ hebben. Veronderstel dat n niet priem is, dan zijn er $d, e > 1$ met $n = d \cdot e$. Maar zowel d als e moeten groter dan $\sqrt{n}$ zijn, dus is hun product groter dan n, een tegenspraak met $n = d \cdot e$. Daarom moet n wel priem zijn.

Bij het maken van lijstjes krijg je sterk de indruk dat je steeds grotere priemgetallen kunt vinden.

1.3.3 Stelling. *Er zijn oneindig veel priemgetallen.*

Bewijs Stel er zijn maar eindig veel priemgetallen $p_1, p_2, \ldots, p_s$. Neem het product $p_1 \cdot p_2 \cdots p_s$ en tel er 1 bij op. Het resulterende getal is niet deelbaar door p_1, en niet door p_2, enzovoorts (op grond van Lemma 1.1.5). Maar het is groter dan 1 en bezit dus een priemfactorontbinding, waarin minstens één priemgetal voorkomt: tegenspraak.

Nu zijn er nog gradaties in oneindigheid. De fameuze *priemgetalstelling* zegt dat het aantal priemgetallen tot x, dus de functie $\pi(x)$, ongeveer groeit als $x/\log x$. Ruw gezegd betekent dit dat in de buurt van x ongeveer 1 op de $\log x$ getallen priem is. Een gevolg van de priemgetalstelling zegt het volgende.

1.3.4 Stelling.

$$\sum_{\substack{p \text{ priem} \\ p<N}} \frac{1}{p} > \log\log N, \qquad \textit{en de som is in het bijzonder onbegrensd.}$$

Het bewijs van deze resultaten voert veel te diep in de analytische getaltheorie.

♣ **Opgave 7.** *Vind de 25 priemgetallen onder de 100 met de zeefmethode.*

♣ **Opgave 8.** *Kijk naar de k opeenvolgende getallen $(k+1)!+2$, $(k+1)!+3$, …, $(k+1)!+k+1$ en concludeer dat je zo k opeenvolgende samengestelde getallen kunt maken: dus er zitten willekeurig grote gaten tussen priemgetallen.*

1.4 Curiosa

Sommige families getallen van een speciale vorm hebben een speciale rol in de geschiedenis van de (priem)getaltheorie. Hieronder zijn de *Fermatgetallen* en de *Mersennegetallen*.

De Fermatgetallen F_k zijn alle getallen van de vorm $2^{2^k}+1$. Fermat sprak in 1637 het vermoeden uit dat F_k een priemgetal zou zijn voor alle $k \geq 0$. Tot dusverre zijn alleen $F_0 = 3$, $F_1 = 5$, $F_2 = 17$, $F_3 = 257$, en $F_4 = 65537$ priem gebleken, en tenminste alle getallen $F_5, \ldots, F_{32}$ samengesteld. Een 'toepassing' van Fermatpriemgetallen is een prachtige stelling van Gauss die zegt dat de een regelmatige veelhoek die je met passer en lineaal kunt construeren een aantal hoeken moet hebben waarvan het oneven deel alleen uit verschillende Fermatpriemgetallen mag bestaan.

Van iets groter belang voor priemtheorie zijn de Mersennegetallen $M_k = 2^k - 1$, met k zelf een priemgetal. Marin Mersenne gaf in het begin van de 17e eeuw een lijst van priemgetallen k waarvoor M_k priem zou zijn; er zaten wat fouten in, maar sinds die tijd is bijna altijd het grootst bekende priemgetal een Mersennegetal geweest. We komen hier nog op terug. Men vermoedt dat er oneindig veel Mersenne-priemgetallen zijn. Op dit moment is het grootste bekende priemgetal $2^{43.112.609} - 1$.

Een groot raadsel is de vraag of er oneindig veel *priem-tweelingen* zijn. Een priemtweeling bestaat uit een paar $p, p+2$ van priemgetallen, zoals $3, 5$ en $5, 7$. Ondanks dat is wel bekend dat de som

$$\sum_{p \text{ priem en } p+2 \text{ priem}} \frac{1}{p}$$

begrensd is (constante van Brun). Het beste resultaat in deze richting laat zien dat er oneindig veel paren $p, p+2$ zijn waarvoor p priem is en $p+2$ priem of het product van twee priemgetallen. Dat lijkt weer erg op het beste resultaat in richting van het vermoeden van Goldbach uit 1742, dat elk even getal de som van twee priemgetallen is; dat resultaat laat namelijk zien dat er maar hoogstens eindig veel uitzonderingen zijn op de regel dat elk even getal de som is van twee priemgetallen of van een priemgetal en het product van twee priemgetallen.

Echte curiosa komen we tegen op het gebied van formules voor priemgetallen, en dergelijke.

1.4.1 Stelling. *Er is een reëel getal α zodat het n-de priemgetal*

$$p_n = \lfloor 10^{2^{n+1}} \alpha \rfloor - 10^{2^n} \lfloor 10^{2^n} \alpha \rfloor.$$

Er is ook een reëel getal β zodanig dat $\lfloor \beta^{3^n} \rfloor$ priem is voor elk natuurlijk getal n.

1.4.2 Stelling. *Er is een polynoom in 26 variabelen en met gehele coëfficiënten dat de eigenschap heeft dat de positieve waarden van het polynoom bij het invullen van natuurlijke getallen en 0 voor de variabelen precies alle priemgetallen zijn.*

De laatste stelling is minder frivool dan je misschien zou denken, en heeft wel degelijk een diepere betekenis, waarop we hier verder niet in kunnen gaan. Een expliciet voorbeeld met de genoemde eigenschap is dit polynoom:

$$\begin{aligned} P = (k+2)\cdot(1 \;&- (w\cdot z + h + j - q)^2 \\ &- ((g\cdot k + 2\cdot g + k + 1)\cdot(h+j) + h - z)^2 \\ &- (2\cdot n + p + q + z - e)^2 \\ &- (16\cdot(k+1)^3\cdot(k+2)\cdot(n+1)^2 + 1 - f^2)^2 \\ &- (e^3\cdot(e+2)\cdot(a+1)^2 + 1 - o^2)^2 \\ &- ((a^2-1)\cdot y^2 + 1 - x^2)^2 \\ &- (16\cdot r^2\cdot y^4\cdot(a^2-1) + 1 - u^2)^2 \\ &- (((a + u^2\cdot(u^2-a))^2 - 1)\cdot(n + 4\cdot d\cdot y)^2 + 1 - (x + c\cdot u)^2)^2 \\ &- (n + l + v - y)^2 \\ &- ((a^2-1)\cdot l^2 + 1 - m^2)^2 \\ &- (a\cdot i + k + 1 - l - i)^2 \\ &- (p + l\cdot(a - n - 1) + b\cdot(2\cdot a\cdot n + 2\cdot a - n^2 - 2\cdot n - 2) - m)^2 \\ &- (q + y\cdot(a - p - 1) + s\cdot(2\cdot a\cdot p + 2\cdot a - p^2 - 2\cdot p - 2) - x)^2 \\ &- (z + p\cdot l\cdot(a-p) + t\cdot(2\cdot a\cdot p - p^2 - 1) - p\cdot m)^2). \end{aligned}$$

Het is niet eenvoudig om positieve waarden te vinden ...

♣ **Opgave 9.** *Laat zien: als $m = d\cdot e$ met d oneven en e even, dan is $2^m + 1$ deelbaar door $2^b + 1$. Dus $2^m + 1$ kan alleen priem zijn als m een macht van 2 is.*

♣ **Opgave 10.** *Laat zien: als $k = d\cdot e$ dan is $2^k - 1$ deelbaar door $2^d - 1$. Dus $2^k - 1$ kan alleen priem zijn als k priem is.*

♣ **Opgave 11.** *Laat zien dat $F_0\cdot F_1\cdots F_4 = M_{32}$.*

♣ **Opgave 12.** *Het grootst bekende priemgetal heeft meer dan 10 miljoen cijfers. Hoeveel bits heeft het in de binaire schrijfwijze?*

2
Ontbinden

2.1 Priemgetallen en algoritmen

In dit hoofdstuk vertellen we iets over methoden om van een natuurlijk getal een niet-triviale factor te vinden. We zoeken een *algoritme*: een methode die stapsgewijze instructies geeft om een bepaalde berekening te doen. Dat willen we graag omdat we die instructies dan vervolgens door een computer uit kunnen laten voeren. Het begrip algoritme is echter al veel ouder dan de computer, en bestaat eigenlijk al zolang als er gerekend wordt. (Vergeet ook niet dat *computer* niets anders dan *rekenaar* betekent, en pas sinds korte tijd wordt daarmee eigenlijk 'electronic computer' bedoeld!)

Er is een methode die erg voor de hand ligt: probeer systematisch of n door de opeenvolgende getallen $2, 3, 4, 5, 6, \ldots, n-1$ deelbaar is. Als n door d deelbaar is, deel je n zo vaak door d als mogelijk is, en met het quotient ga je verder.

Dit lijkt een redelijke formulering van het resulterende algoritme:

2.1.1 Algoritme. Probeer voor $d = 2, 3, \cdots, n-1$ of d een deler is van n; als d inderdaad n deelt dan vervang je n door n/d en begint opnieuw. Als je tot en met $n-1$ geen delers vindt, stop je.

Dit is een *recursief* algoritme: je past het herhaaldelijk toe op een getal n dat steeds kleiner wordt.

Als je er iets beter over nadenkt, zie je dat er een aantal dingen mis zijn met dit algoritme, of liever gezegd, dat het veel beter kan. In de eerste plaats hoef je niet steeds opnieuw te beginnen met $d = 2$, maar kun je verder gaan met de d waar je gebleven was. Dat levert vooral winst op als je het combineert met de tweede verbetering: je hoeft niet door te gaan met d tot $n-1$, maar je mag als stoppen bij $\sqrt{n}$. Tenslotte hoef je niet alle d te proberen, maar slechts de priemgetallen d (probeer dat zorgvuldig te beredeneren!).

2.1.2 Voorbeeld. Laten we het algoritme toepassen op $n = 8211$. Omdat $8100 = 90^2 < n < 91^2 = 8281$ gaan we proberen of n deelbaar is door $d = 2, 3, \ldots, 90$. Het getal n is niet even, maar blijkt wel deelbaar door 3, want $8211 = 3 \cdot 2737$. We gaan dus verder met de nieuwe $n = 2737$ en kunnen nu al ophouden bij $d = 52$ omdat $52^2 < n < 53^2$. Deze n is niet meer deelbaar door 3, en natuurlijk ook niet door 4 want al niet door 2. Het is duidelijk dat n ook niet deelbaar is door 5 (kijk naar het laatste cijfer), en 6

hoeven we niet te proberen. Bij 7 vinden we $n = 2737 = 7 \cdot 391$, dus vervangen we n door $n = 391$. We hoeven alleen nog $d = 7, 8, \ldots, 19$ te proberen omdat $19^2 < n < 20^2$. Maar n is niet meer deelbaar door 7, en $8, 9, 10$ hoeven we niet te proberen, net als 12; door 11 delen blijkt niet te gaan, evenmin als 13, en $14, 15, 16$ mogen we weer overslaan. Bij 17 vinden we $391 = 17 \cdot 23$. Dan zijn we klaar, want voor $n = 23$ hoeven we maar tot $d = 4$ te gaan en daar zijn we al lang voorbij: $n = 23$ moet wel priem zijn! Uiteindelijk is de conclusie, met niet al te veel werk, dat $8211 = 3 \cdot 7 \cdot 17 \cdot 23$.

In het voorbeeld zie je dat we alleen maar proberen te delen door priemgetallen d zonder eerst een tabel van priemgetallen te hebben. In feite *genereren* we een lijst van kleine priemgetallen onderweg, als in de zeef die we zagen.

Toch werkt de methode van test-delen voor het vinden van alle factoren niet als n te groot wordt, simpelweg omdat het te lang duurt om alle mogelijkheden te proberen. Zelfs met de snelste computers is het onbegonnen werk om zo het product van twee priemgetallen van zeg 50 cijfers in factoren te ontbinden. Ga maar na hoeveel d tot 10^{50} je moet proberen, en er 10^9 per seconde proberen is al veel te optimistisch...

Tussen twee haakjes: afgezien van de vraag of je zo'n berekening in een dag of een jaar zou kunnen volbrengen, rijst de vraag welke waarde je eraan hecht wanneer de computer je na zoveel tijd alleen maar vertelt dat er geen deler bestaat. Het liefst zou je zo'n berekening willen kunnen verifiëren zonder deze helemaal opnieuw te hoeven doen!

In het volgende zal ik een methode uitleggen die, paradoxaal genoeg, in staat is redelijk efficiënt te bewijzen dat een getal *niet priem* is, zonder een deler van dat getal te vinden. Eigenlijk geeft dat algoritme je heel weinig informatie: het kan je met zekerheid vertellen dat een getal delers heeft zonder iets over die delers te zeggen, maar het kan je niet met zekerheid zeggen dat een getal priem is.

Het is een *probabilistisch algoritme*, wat betekent dat er af en toe aselecte keuzes (willekeurige keuzes) gemaakt moeten worden. Als je niet van die willekeurige keuzes hoeft te maken, spreek je van een *deterministisch* algoritme.

2.2 Probabilistische algoritmen

Om uit te leggen wat een probabilistisch algoritme is, maak ik eerst een zijsprong.

Veronderstel eens dat gevraagd wordt voor het volgende probleem een oplossing te vinden:

> *In een vaas zitten gekleurde knikkers. Je kunt de knikkers slechts één voor één uit de vaas halen om te bekijken. Beslis welke kleuren onder de knikkers in de vaas daadwerkelijk voorkomen.*

Het is duidelijk wat je te doen staat: bekijk alle knikkers en meldt welke kleur ze hebben.

Zonder verdere informatie geeft pas de allerlaatste knikker je uitsluitsel over de kleuren die voorkomen. Maar als je meer weet, bijvoorbeeld dat er maar hoogstens twee kleuren voorkomen kan het zijn dat je al na twee knikkers bekeken te hebben klaar bent.

Als er heel veel knikkers in de vaas zitten zou je misschien al na het bekijken van een relatief klein aantal knikkers *iets* willen zeggen – bijvoorbeeld: 'het is *waarschijnlijk* dat er alleen maar knikkers van één kleur in de vaas zitten'. Maar is dat wel gerechtvaardigd?

Daarvoor zou het handig zijn om twee dingen te weten:

(1) Hoeveel knikkers van beide kleuren kunnen er in de vaas zitten?
(2) Hoe haal je de knikkers uit de vaas?

Wat het laatste betreft: je zou misschien willen weten dat de trekking *aselect* is, dat wil zeggen dat elke knikker in de vaas dezelfde kans heeft om getrokken worden. Je wilt

niet systematisch meer kans hebben om bijvoorbeeld een witte te trekken, zeg omdat ze naar boven drijven. We zullen steeds aannemen dat inderdaad steeds een willekeurige knikker zonder voorkeur getrokken wordt.

Hoe meer je weet over (1) hoe beter het is. Bijvoorbeeld, als je tevoren weet dat er N knikkers in de vaas zitten, en

> *als* er een knikker van een bepaalde kleur voorkomt, dan zijn er *minstens* k van die kleur

dan weet je dat je nooit meer dan $N-(k-1)$ knikkers hoeft te trekken om een juiste conclusie met zekerheid te trekken. Maar in het ergste geval zal dat ook inderdaad betekenen dat je er $N-k+1$ moet trekken.

2.2.1 Voorbeeld. Stel eens dat je weet dat er alleen zwarte en witte knikkers zijn, èn dat

> *als* er knikker van een bepaalde kleur voorkomt, dan heeft *minstens de helft van alle knikkers* die kleur

Dan zijn er dus maar 3 mogelijkheden over: alle knikkers zijn wit, ze zijn allemaal zwart, of de helft is wit en de helft zwart. Na het zien van één knikker valt er al een mogelijkheid af. Laten we zeggen dat de eerste zwart is; bij het trekken van de tweede knikker zijn er weer twee mogelijkheden: we trekken een witte en weten dan zeker dat beide kleuren voorkomen. (We zeggen wel dat de witte dan *getuige* is voor het feit dat er twee kleuren voorkomen.) De tweede mogelijkheid is dat we weer een zwarte trekken, en weten dan niets nieuws (tenzij we weten dat er niet meer dan 3 knikkers zijn!). Bij de derde net zo, enzovoort.

Als er maar 1 kleur knikkers in de vaas zit, weet je dat pas *zeker* na het trekken van $\frac{N}{2}+1$ knikkers. Omdat we de knikkers willekeurig trekken, kunnen we in het geval dat er 2 kleuren knikkers in de vaas zitten *uitrekenen* hoe groot de kans is dat we na het bekijken van t knikkers *weten* dat er 2 kleuren in de vaas zitten, namelijk als volgt.

Na het trekken van 1 knikker weten we nog niks. Bij de tweede knikker hebben we kans van ongeveer $\frac{1}{2}$ dat we er één van dezelfde kleur trekken , en ook kans ongeveer $\frac{1}{2}$ dat deze de andere kleur heeft. In het laatste geval weet je zeker dat beide kleuren voorkomen – dus dat weet je met kans $\frac{1}{2}$ al na twee trekkingen! Als de tweede ook zwart was, ga je door met trekken van een derde knikker: er is weer een kans van ongeveer $\frac{1}{2}$ dat die wit is, en alleen als die ook zwart is weet je nog niets zeker. De kans dat de eerste drie allemaal dezelfde kleur hebben is (ongeveer) een kwart (ga na!), en in dat geval ga je door. Je ziet misschien zo al dat na het trekken van t knikkers de kans dat je het nog niet zeker weet $(\frac{1}{2})^{t-1}$ is, en dat je het wél zeker weet dus ongeveer $1-(\frac{1}{2})^{t-1}$. De kans dat je het zeker weet wordt dus steeds groter.

De uitspraak dat beide kleuren voorkomen zul je steeds met zekerheid kunnen doen. En je bent geneigd om dus na pakweg $t=100$ keer trekken van een zwarte knikker te denken dat er wel *alleen* zwarte knikkers in de vaas zullen zitten. Toch weet je dat dan nog steeds niet zeker (tenzij t groter wordt dan $N/2$ natuurlijk). Het is al beter om dan te zeggen dat er *hoogstwaarschijnlijk* alleen zwarte knikkers in de vaas zitten.

Je moet wel een beetje uitkijken met zulke uitspraken: het is bijvoorbeeld niet erg zinvol om te zeggen dat *er een grote kans is* dat er alleen zwarte knikkers in de vaas zitten: er zitten namelijk ofwel uitsluitend zwarte in of zwarte en witte, en het is niet zinnig over kansen te spreken in dit verband. Wat je bedoelt te zeggen is dat als je dit experiment heel vaak zou herhalen dan zou het maar uiterst zelden voorkomen dat je achter elkaar 100 zwarte knikkers uit de vaas trekt als er evenveel zwarte als witte in de vaas zitten!

We schreven boven over kansen van *ongeveer* $\frac{1}{2}$ omdat die kans een klein beetje verandert als je eenmaal knikkers hebt getrokken — maar als N groot is blijft zelfs nadat je een paar knikkers uit de vaas hebt gehaald vrijwel de helft van alle knikkers wit als oorspronkelijk precies de helft wit was.

2.3 Priem of niet?

We gaan nu een probabilistische methode beschrijven om onderscheid te maken tussen priemgetallen en samengestelde getallen.

Een voor de hand liggende methode is om willekeurig getallen tussen 1 en n te kiezen en te kijken of je een deler vindt. Eigenlijk is dat een probabilistische variant van het test-delen dat we beschreven, waar je achtereenvolgens $d = 2, 3, 4, \cdots$ als deler probeert. De reden om d willekeurig te kiezen en het algoritme zo dus probabilistisch te maken, is dat je vervolgens iets kunt *bewijzen* over de kans van slagen.

Maar helaas zijn er te weinig delers om dit tot een succesvolle methode te maken. Een iets betere methode krijg je al wanneer je niet bekijkt of d een deler van n is, maar of d en n een deler gemeen hebben. Voor een veel betere methode bekijken we een wat ingewikkelder eigenschap van priemgetallen.

2.3.1 Stelling. *Laat $n > 2$ een oneven natuurlijk getal zijn en a een natuurlijk getal met $1 < a < \sqrt{n}$. Schrijf $n - 1 = 2^k \cdot r$, met r oneven, en schrijf $b = a^r$.*

Als n een priemgetal is, dan is tenminste één van de getallen

$$b - 1, b + 1, b^2 + 1, b^{2^2} + 1, b^{2^3} + 1, \cdots, b^{2^{k-1}} + 1$$

deelbaar door n.

Helaas is het *niet* waar dat als één van de getallen

$$b - 1, b + 1, b^2 + 1, b^{2^2} + 1, b^{2^3} + 1, \cdots, b^{2^{k-1}} + 1$$

deelbaar door n is, dat n dan een priemgetal moet zijn. Dat zou een prachtige test opleveren om priemgetallen te vinden!

Maar uit de stelling volgt *wel* onmiddellijk dat als we bij een oneven natuurlijk getal n, met $n - 1 = 2^k \cdot r$ een getal a kunnen vinden zodat met $b = a^r$ géén van de getallen $b - 1, b + 1, b^2 + 1, \cdots, b^{2^{k-1}} + 1$ deelbaar door n, we zeker weten dat n niet priem is! Zo'n a noemen we dan wel een *getuige voor het niet-priem zijn* van het getal n.

2.3.2 Voorbeelden. Neem $n = 35$. Dan is $n - 1 = 34 = 2 \cdot 17$, dus $k = 1$ en $r = 17$. Neem ook $a = 2$, dan is $b = a^{17} = 131072$. De Stelling zegt dan dat als $n = 35$ een priemgetal is, tenminste één van de getallen

$$b - 1 = 131071, \quad b + 1 = 131073$$

deelbaar is door 35. Maar noch 131071, noch 131073 is deelbaar door 35 *dus* kan $n = 35$ geen priemgetal zijn!

Doen we hetzelfde met $n = 37$, zodat $n - 1 = 2^2 \cdot 9$ en $k = 2$, $r = 9$. Met $a = 2$ en dus $b = 2^9 = 512$ moeten we kijken naar

$$b - 1 = 511, \quad b + 1 = 513, \quad b^2 + 1 = 262145 = 37 \cdot 7085.$$

De Stelling vertelt ons nu alleen maar dat het mogelijk is dat n een priemgetal is, we weten nog niets zeker.

Er is een heel practische manier om te zien of een getal als $a^r - 1$ (zoals in dit kleine voorbeeld $2^{17}-1 = 131071$) deelbaar is door een relatief klein getal (zoals hier 35) zonder eerst die grote macht helemaal uit te schrijven. Die methode heet *modulo rekenen* en illustreren we in een voorbeeld.

2.3.3 Voorbeeld. In plaats van 2^{17} helemaal uit te rekenen en dan te kijken of $2^{17} - 1$ deelbaar is door 35, gaan we ervoor zorgen dat ons tussenresultaten steeds kleiner dan 35 worden met behulp van deling met rest.

Eerst bereken je een paar machten totdat je getal voor het eerst groter dan 35 wordt. Dus $2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16$ en $2^5 = 32, 2^6 = 64$. Voordat we weer met 2 vermenigvuldigen maken we het tussenresultaat kleiner dan 35 door er 35 af trekken, zoveel keer als maar mogelijk is zonder een negatief getal te krijgen. Hier vervangen we $2^6 = 64$ dus door $64 - 35 = 29$. We schrijven $2^6 \equiv 29 \bmod 35$. Dan weer met 2 vermenigvuldigen tot we 35 overschrijden: $2^7 \equiv 58$ en we trekken er weer (een veelvoud van) 35 af, $2^7 \equiv 23 \bmod 35$, en $2^8 \equiv 46 \equiv 11, 2^9 \equiv 22, 2^{10} \equiv 44 = 9 \bmod 35$ en zo door: $2^{11} \equiv 18 \bmod 35$, $2^{12} \equiv 36 \equiv 1, 2^{13} \equiv 2 \bmod 35$, $2^{14} \equiv 4, 2^{15} \equiv 8, 2^{16} \equiv 16 \bmod 35$, en uiteindelijk $2^{17} \equiv 32 \bmod 35$. Het was ons er om te doen of $2^{17} - 1$ deelbaar is door 35: het antwoord is *nee*, omdat $2^{17} - 1 \equiv 32 - 1 = 31 \bmod 35$. Het antwoord zou alleen *ja* geweest zijn als $2^{17} - 1 \equiv 0 \bmod 35$. Omdat $2^{17} + 1 \equiv 32 + 1 = 33 \bmod 35$ is ook $2^{17} + 1$ niet deelbaar door 35.

Je kunt nog meer trucjes bedenken om sneller je antwoord te vinden: toen we eenmaal hadden dat $2^8 \equiv 11 \bmod 35$ en $2^9 \equiv 22 \bmod 35$ hadden we sneller kunnen zien dat $2^{17} = 2^8 \cdot 2^9 \equiv 11 \cdot 22 = 242 \equiv 32 \bmod 35$, omdat $242 - 6 \cdot 35 = 32$.

Wat de vorige Stelling zo bruikbaar maakt is dat er bij elk getal dat samengesteld is ook veel getallen a te vinden zijn die daar getuige voor zijn! Dat is wat de volgende stelling uitdrukt.

2.3.4 Stelling. *Voor elke oneven getal $n > 1$ dat niet priem is, is tenminste $\frac{3}{4}$ van de getallen $1, 2, 3, \cdots, n-2, n-1$ getuige voor het niet-priem zijn.*

Door nu voor a willekeurige keuzes uit $1, 2, 3, \cdots, n-2, n-1$ te maken geeft dit een prachtige probabilistische test die grote kans van slagen heeft om voor een samengesteld getal een getuige voor dat samengesteld zijn te vinden. Voor priemgetallen kun je er nooit meer mee laten zien dan dat het (na zeg twintig willekeurige keuzes voor a die falen) erg *waarschijnlijk* is dat het getal priem is; bewijzen kun je dit er niet mee.

De werking van de methode berust op een stelling die we later weer zullen gebruiken (en die naar Fermat wordt genoemd).

2.3.5 Stelling. *Als p een priemgetal is, en $1 < a < p$, dan deelt p het getal $a^{p-1} - 1$.*

2.4 En verder?

Zoals gezegd, geeft de test uit het vorige hoofdstukje heel merkwaardige informatie. Je kunt er (vrij snel) mee *aantonen* dat een getal samengesteld is, zoals je van een vaas met evenveel witte als zwarte knikkers erin door willekeurige trekking meestal vrij snel aantoont dat dat zo is. Maar je kan er niet mee *bewijzen* dat een getal priem is, netzomin als herhaald trekken van dezelfde kleur knikkers bewijst dat er alleen maar knikkers van één kleur in een vaas zitten.

Voor een samengesteld getal blijft dan de taak over om de priemfactoren te vinden, terwijl voor een getal dat *hoogstwaarschijnlijk priem* is de taak overblijft om te bewijzen

dat dat inderdaad zo is! Voor beide taken zijn algoritmen verzonnen, die aanzienlijk gecompliceerder zijn dan het bovenstaande. De beste algoritmen (die probabilistisch zijn, dus van aselecte trekkingen gebruik maken) kunnen op snelle computers van getallen van enkele duizenden cijfers bewijzen dat ze priem. De grenzen van de beste factorisatie-algoritmen liggen bij het ontbinden van producten van priemgetallen van 60 of 70 cijfers.

♣ **Opgave 13.** *Schrijf alle delers op van 2^3, van 3^4 en van $2^3 \cdot 3^4$. Kun je een formule raden voor het aantal delers van een natuurlijk getal $n = p_1^{k_1} \cdot p_2^{k_2} \cdot \cdots \cdot p_t^{k_t}$?*

♣ **Opgave 14.** *Gebruik $a = 2$ in stelling 2.3.5 om te laten zien dat $n = 21$ niet priem is.*

Als we ons beperken tot de kwestie van het vinden van delers lijkt het redelijk om te zeggen dat een getal erg groot is als een computer er erg veel moeite mee heeft om een deler te vinden. Maar kijk eens naar de getallen

$$m = 80346902212949513777098104617058130126110149689139641765068 7,$$

en

$$n = 80346902212949513777098104617058130126110149689139641765068 8.$$

Dat zijn allebei getallen van 60 cijfers (en n is 1 groter dan m) en toch is het veel moeilijker van de kleinste een deler te vinden dan van de grootste: het is zelfs voor een mens gemakkelijk een deler van n aan te wijzen, want n is *even*. Maar m is oneven en het vinden van een deler kost ook een snelle computer nog enige moeite. Hoe moeilijk het is een deler van een getal te vinden hangt dus niet uitsluitend af van de vraag hoe groot het getal is in absolute zin. Het getal n is wel heel simpel, want in feite geldt dat $n = 2 \cdot 2 \cdot 2 \cdot \cdots \cdot 2$ (een product van 199 tweeën), en dat schrijven we dan

$$n = 2^{199}$$

terwijl

$$m = 164504919713 \cdot 4884164093883941177660049098586324302977543600799.$$

Dus m is veel 'moeilijker' te *ontbinden* dan n.

2.5 Priem of niet?

Als je getal n een speciale mooie vorm heeft kun je soms nog sneller zien of een getal priem is of niet. Dit geldt bijvoorbeeld voor de Mersennegetallen van de vorm $2^k - 1$, waarvan m een voorbeeld was (met $k = 199$). Voor zulke getallen hoef je alleen maar uit te rekenen (als $k > 2$):

$$\begin{aligned} w_1 &= 4, \\ w_2 &\equiv w_1^2 - 2 \bmod (2^k - 1), \\ &\vdots \\ w_{k-1} &\equiv w_{k-2}^2 - 2 \bmod (2^k - 1), \end{aligned}$$

en dan is $2^k - 1$ priem dan en slechts dan als $w_{k-1} = 0$. Met mod geven we steeds aan dat je door deling met rest de getallen klein(er dan $2^k - 1$) houdt. Voor $k = 7$ vinden we bijvoorbeeld:

$$
\begin{aligned}
w_1 &= 4,\\
w_2 &\equiv 4^2 - 2 \equiv 14 \bmod (2^7 - 1),\\
w_3 &\equiv 14^2 - 2 \equiv 67 \bmod (2^7 - 1),\\
w_4 &\equiv 67^2 - 2 \equiv 42 \bmod (2^7 - 1),\\
w_5 &\equiv 42^2 - 2 \equiv 111 \bmod (2^7 - 1),\\
w_6 &\equiv 111^2 - 2 \equiv 0 \bmod (2^7 - 1),
\end{aligned}
$$

en dus is $2^7 - 1 = 127$ een priemgetal. Met deze methode zijn ook de grootste thans bekende priemgetallen gevonden.

2.6 Pollard-ρ

Nu kunnen we de factorisatiemethode beschrijven die Pollard-ρ genoemd wordt (naar de ontdekker en de vorm van het plaatje dat bij de methode hoort en op de Griekse letter ρ lijkt). De methode is bedoeld om priemfactoren te vinden van een getal n. Om te zien wat er gebeurt zullen we aannemen dat $n = p \cdot q$, het product van twee priemgetallen $p \neq q$ (maar de methode werkt ook als er meer priemfactoren zijn). Het idee is om een functie f te maken die aan een getal modulo n (dat wil zeggen: een rest tussen 0 en $n - 1$) een nieuw (min of meer willekeurig) getal modulo n toevoegt. We beginnen met een of andere waarde en gaan vervolgens steeds f loslaten op het resultaat. Omdat er maar eindig veel verschillende getallen modulo n zijn zal na eindig veel stappen een waarde worden aangenomen die al eerder werd aangenomen. Vanaf dat moment worden eerdere stappen herhaald: de functie bijt zichzelf als het ware in de staart, vandaar het plaatje van de ρ. Hoewel je p en q niet weet als je begint, bepaalt de functie f in elke stap een waarde modulo p en een waarde modulo q. Net als dat het geval was modulo n zal na eindig veel stappen een herhaling modulo p optreden. Hoewel je p niet kent, kun je toch deze herhaling opsporen!

Laten we de beginwaarde met x_0 aangeven; na één stap vinden we $x_1 \equiv f(x_0) \bmod n$, dan $x_2 \equiv f(x_1) \bmod n$, enzovoorts. Veronderstel eens dat na 5 stappen een herhaling modulo de onbekende p optreedt, bijvoorbeeld modulo p is x_5 hetzelfde als x_2. Ook al weten we p niet, als $x_5 \equiv x_2 \bmod p$, dan betekent dit dat p een deler is van $\text{ggd}(x_5 - x_2, n)$. Het kan zijn dat er ook een herhaling modulo q optrad, zodat de $\text{ggd}(x_5 - x_2, n) = n$. In dat geval schieten we niets op. Maar als dat niet zo is, zal $\text{ggd}(x_5 - x_2, n) = p$, en dus precies de gezochte priemdeler p opleveren!

Als we nu vertellen wat de functie f is, dan kunnen we naar een voorbeeld kijken: $f(x) = x^2 + 1 \bmod n$. Hoewel deze functie natuurlijk allerminst 'willekeurig' lijkt, blijkt in de praktijk dat de afbeelding zich willekeurig genoeg gedraagt. Als beginwaarde wordt vaak $x_0 = 1$ genomen. Laten we $n = 77$ nemen, dan berekenen we:

$$
\begin{aligned}
x_1 &= 1^2 + 1 \equiv 2 \bmod 77\\
x_2 &= 2^2 + 1 \equiv 5 \bmod 77\\
x_3 &= 5^2 + 1 \equiv 26 \bmod 77\\
x_4 &= 26^2 + 1 \equiv 61 \bmod 77\\
x_5 &= 61^2 + 1 \equiv 26 \bmod 77
\end{aligned}
$$

zodat $x_5 \equiv x_3 \bmod 77$. Maar als we in staat waren modulo de priemdeler 7 van 77 te kijken dan zouden we al eerder een herhaling vinden, want modulo 7:

$$
\begin{aligned}
x_1 &= 1^2 + 1 \equiv 2 \bmod 7\\
x_2 &= 2^2 + 1 \equiv 5 \bmod 7\\
x_3 &= 5^2 + 1 \equiv 5 \bmod 7.
\end{aligned}
$$

Inderdaad, door x_3 en x_2 modulo n met elkaar te vergelijken, en zien we dat $\text{ggd}(x_3 - x_2, n) = \text{gcd}(26 - 5, n) = 7$. Het blijkt dat om de factor p in de ggd op te sporen, het

efficient is om de waarden voor $x_0, x_1, x_2, \ldots$ modulo n uit te rekenen, en $\text{ggd}(x_j - x_i, n)$ te berekenen op een manier die er voor zorgt dat $j - i$ steeds groter wordt. Daartoe beschouwen we de ggd met n van $x_3 - x_1$, dan $x_6 - x_3$ en $x_7 - x_3$, daarna $x_{12} - x_7, x_{13} - x_7, x_{14} - x_7, x_{15} - x_7$ en dan $x_{24} - x_{15}$, enzovoorts.

Hier is een voorbeeld, voor $n = 893$:

$$\begin{aligned}
x_1 &= 1^2 + 1 \equiv 2 \bmod 893, \\
x_2 &= 2^2 + 1 \equiv 5 \bmod 893, \\
x_3 &= 5^2 + 1 \equiv 26 \bmod 893, \quad \text{ggd}(x_3 - x_1, n) = \text{ggd}(21, 893) = 1 \\
x_4 &= 26^2 + 1 \equiv 677 \bmod 893 \\
x_5 &= 677^2 + 1 \equiv 221 \bmod 893 \\
x_6 &= 221^2 + 1 \equiv 620 \bmod 893, \quad \text{ggd}(x_6 - x_3, n) = \text{ggd}(594, 893) = 1 \\
x_7 &= 620^2 + 1 \equiv 411 \bmod 893, \quad \text{ggd}(x_7 - x_3, n) = \text{ggd}(385, 893) = 1 \\
x_8 &= 411^2 + 1 \equiv 145 \bmod 893 \\
x_9 &= 145^2 + 1 \equiv 487 \bmod 893 \\
x_{10} &= 487^2 + 1 \equiv 525 \bmod 893 \\
x_{11} &= 525^2 + 1 \equiv 278 \bmod 893 \\
x_{12} &= 278^2 + 1 \equiv 487 \bmod 893, \quad \text{ggd}(x_{12} - x_7, n) = \text{ggd}(76, 893) = 19.
\end{aligned}$$

We vinden een factor 19, en inderdaad $893 = 19 \cdot 47$.

Met precies dezelfde methode vinden we na 491279 stappen in enkele minuten op een gewone PC de factorisatie

$$2^{199} - 1 = 164504919713 \cdot 488416409388394117766004909858632430297754360079,$$

die we al noemden.

2.7 De verjaardagsparadox

Waarom werkt de methode Pollard-ρ zo goed? Dat komt omdat je kunt laten zien dat als de functie f echt willekeurige waarden aanneemt modulo n, je mag verwachten dat er een herhaling in de waarden modulo p optreedt na gemiddeld ongeveer $\sqrt{p}$ stappen. De reden daarvoor is dezelfde als degene die ten grondslag ligt aan de *verjaardagsparadox*: vraag aan een groep van g mensen naar hun verjaardagen. Hoe groot denk je dat g moet zijn voordat je een kans van meer dan 50% hebt dat twee mensen op dezelfde dag jarig zijn? Het verrassende antwoord is: ongeveer 23. Ook hier geldt dat de kans dat twee waarden (verjaardagen) samenvallen meer dan een half wordt zodra het aantal waarden (g dus) groter wordt dan een getal dat ongeveer de wortel uit het totaal aantal mogelijkheden (de 365 dagen) is. Dat blijkt als je uitrekent hoe groot de kans is dat alle verjaardagen op verschillende dagen vallen. Als $g = 1$ is die kans natuurlijk 1. Voor $g = 2$ is de kans $364/365$ (want de tweede verjaardag mag op 1 dag niet vallen); voor $g = 3$ wordt de kans $(364/365) \cdot (363/365) \approx 0,992$, enzovoorts, en die kans wordt bij $g = 23$ voor het eerst kleiner dan een half.

3 Cryptografie

3.1 Complexiteit

In dit hoofdstukje wil ik een paar elementaire zaken uit de complexiteitstheorie noemen, omdat dit een belangrijke rol speelt in toepassingen van priemgetallen in de cryptografie.

Complexiteitstheorie poogt de efficiëntie van algoritmen te vergelijken. Eén van de belangrijkste methoden vergelijkt bovengrenzen voor de tijd die het kost om een algoritme uit te voeren. Meestal kun je algoritmen uitvoeren op problemen van heel verschillende omvang, en zo'n bovengrens zal dus afhangen van de omvang van het probleem. Voor verschillende problemen van vergelijkbare omvang kan het heel goed zijn dat een algoritme heel verschillende rekentijd nodig heeft; de bovengrens wordt geacht voor het slechtst denkbare geval nog een bovengrens voor de benodigde tijd te geven.

Hoewel in principe de reken*tijd* bekeken wordt, wil je niet dat de bovengrens (te veel) afhankelijk is van de computer of de software die gebruikt wordt. Als maat wordt daarom niet de werkelijk benodigde tijd gemeten, maar het aantal stappen dat het algoritme moet uitvoeren om het resultaat te vinden. Preciezer gezegd: de rekentijd wordt gemeten in het aantal bit-operaties dat in het slechtste geval nodig is, als functie van het aantal bits waaruit de invoer bestaat.

3.1.1 Voorbeeld. De belangrijkste operaties die we gezien hebben, waren de optelling en vermenigvuldiging van twee natuurlijke getallen. Een optelalgoritme neemt als invoer twee natuurlijke getallen, en levert ook weer een natuurlijk getal af. De invoer wordt gespecificeerd als de bits waarmee de natuurlijke getallen (in het binaire stelsel, met nullen en enen) worden genoteerd. Gemakshalve mag je aannemen dat de beide natuurlijke getallen evenveel bits tellen (neem het maximum van beide).

Voor het optellen van twee natuurlijke getallen die uit N bits bestaan zal niet meer dan ongeveer N bit-operaties vereist zijn; dat aantal is een beetje groter omdat met carries rekening gehouden moet worden.

Voor de vermenigvuldiging zijn heel wat meer operaties nodig, in essentie ongeveer N^2 (om alle paren bits te vermenigvuldigen).

In essentie is dit ook het aantal operaties dat nodig is om met het algoritme van Euclides de grootste gemene deler van twee getallen te bepalen.

We zagen hier voorbeelden van een lineair en van een kwadratisch algoritme. In het algemeen heet een algoritme *polynomiaal van orde k* wanneer er een constante C bestaat zodat het uitvoeren van het algoritme niet meer dan $C \cdot N^k$ bit operaties kost op invoer ter grootte N bits.

In het algemeen worden polynomiale algoritmen als efficiënt bestempeld. De klasse van alle problemen waarvoor een oplossing bekend is waarvoor een polynomiale bovengrens op het aantal bit-operaties bestaat (als functie van het aantal bits aan invoer) voor het uit te voeren algoritme heet de klasse P. De klasse NP (voor *non-deterministisch polynomiale tijd*) bestaat uit problemen waarvoor een algoritme bestaat dat in polynomiaal begrensde tijd (als functie van de grootte van de input) een voorgestelde oplossing kan controleren. Het is niet nodig dat zo'n oplossing in die tijd ook gecreëerd wordt.

Het soort problemen dat in de klasse NP zit maar niet in P heeft typisch een groot aantal potentiële oplossingen, die niet allemaal in polynomiale tijd afgewerkt kunnen worden; wel kan een enkele kandidaat-oplossing in polynomial tijd gecontroleerd worden.

Het zal duidelijk zijn dat P een deelklasse van NP is: polynomiale methoden zoeken *en* controleren oplossingen in polynomiale tijd.

De allergrootste uitdaging op het gebied van theoretische informatica is de vraag: is P= NP ?

In onze contekst is het van belang om te weten dat enkele jaren terug is bewezen dat 'primaliteit in P' zit. Dus er bestaat een efficiënt algoritme om van een gegeven getal onomstotelijk vast te stellen of het priem is of niet. Maar niemand is er nog in geslaagd een algoritme te maken dat in de *praktijk* beter werkt dan sommige veelgebruikte algoritmen die niet (bewezen) polynomiaal zijn. Zulke algoritme kunnen in het eindige gebied waarin wij geneigd zijn te rekenen een veel betere performance geven dan op grond van hun asymptotische rekentijdanalyse verwacht zou mogen worden. Van het probleem om voor een gegeven natuurlijk getal de priemfactorontbinding te vinden is duidelijk dat het in de klasse NP zit (om te controleren dat de factorisatie $n = d \cdot e$ correct is volstaat een enkele vermenigvuldiging). De meest gehoorde verwachting is dat het vinden van factoren *niet* in polynomiale tijd gedaan kan worden.

3.2 Public key cryptografie

Bij wijze van afsluiting geven we een toepassing van priemgetallen in de cryptografie, namelijk RSA.

Het doel van cryptografie is om het mogelijk te maken berichten uit te wisselen over een openbaar kanaal (telefoonlijn, bijvoorbeeld) zonder dat die berichten begrijpelijk worden voor een luistervink.

De voor de hand liggende methode is om een geheime code af te spreken. Een groot probleem is lange tijd geweest hoe deze code af te spreken zonder in elkaars nabijheid te verkeren. De public key cryptografie gaf daar een elegante oplossing voor. Een aardige metafoor voor het gebruikte model is: wanneer je een ander in de gelegenheid wilt stellen jou iets te geven zonder dat een derde hier aan kan komen, kun je op een openbare plaats een open brandkast neerzetten waarvan jij alleen zelf de sleutel hebt. Wie wil kan hieraan zijn boodschap aan jou plaatsen en de brandkast sluiten; alleen jij kunt de boodschap nog lezen omdat je de enige met een sleutel bent.

3.3 RSA

In RSA gaat het om uitwisselen van berichten in de vorm van reeksen bits; gemakshalve zullen we zeggen dat iemand jou een natuurlijk getal m wil sturen dat niemand anders mag zien.

Jij maakt daartoe twee getallen openbaar, namelijk n en e. Het getal m mag niet groter dan n zijn (anders moet het in stukjes worden verstuurd). De zender stuurt aan jou $r = m^e \bmod n$ (dus de rest bij deling door n van m^e). De bedoeling is dat jij nu wel m uit r kunt herleiden, maar anderen dat niet kunnen. De reden is dat je n op een speciale manier gekozen hebt, en informatie achterhoudt (die zelfs de zender van het bericht niet nodig heeft — zoals de sleutel van de brandkast). Het getal n is gekozen als product van twee grote priemgetallen p en q; zo groot, dat je ervan uit mag gaan dat niemand in staat is uit n de factoren p en q te vinden (behalve jij, want je hebt p en q zelf gekozen). Bovendien heb je bij het getal e een getal d gemaakt zodanig dat $(m^e)^d \equiv m \bmod n$, voor elke keus van m. Zo'n d bestaat altijd; dat volgt uit de kleine stelling van Fermat die we eerder noemden, en je kunt hem vinden uit e, p en q (maar niet uit e en n alleen)! In feite moet d eraan voldoen dat $d \cdot e - 1$ deelbaar wordt door $p - 1$ en door $q - 1$, en dat is te bereiken met het algoritme van Euclides.

Alle prettige eigenschappen volgen dus uit feiten over priemgetallen die we eerder zagen, en de veiligheid is gebaseerd op de aannamedat het moeilijk is om n in factoren te ontbinden ....

De Amerikaanse overheid achtte het exporteren van dergelijke algoritmen in de jaren 80/90 van de vorige eeuw even verderfelijk als illegale wapenexport.