Wanneer is een getal priem?

Benjamin Cottyn
Andreas De Lille

1 Inleiding

Dit document/deze les vergelijkt een aantal methodes om te testen of een getal priem is. Eerst worden de trage,naïve methodes beschreven. Vervolgens wordt een snellere probabilistische methode bestudeerd. Om af te sluiten wordt er een deteministische methode besproken.

2 Naïve methodes

De algemene werkwijze hier is alle kleinere getallen aflopen en kijken of het een deler is. Deze methodes zijn bijgevolg traag, maar zeker.

In de tekst hieronder verwijst t naar het getal dat we willen testen.

2.1 Alle kleinere getallen aflopen

Deze eerste methode is de eenvoudigste, hierbij wordt voor elk getal x $\in$ [2;t[getest of x —t. Indien x —t hebben is er een deler gevonden, verschillend van 1 en t. Bijgevolg is t niet priem.

2.2 Alle getallen aflopen tot aan de wortel

De voorgaande methode kan nog sterk geoptimaliseerd worden.
Als we aan de wortel gekomen zijn, zijn alle delers tot en met de wortel (niet) gevonden. Het quotiënt van deze delers is hier groter dan de wortel. Als we verder gaan zullen we dus delers vinden waarvan we het overeenkomstige quotiënt al getest hebben.

bijvoorbeeld het getal 36
De wortel is 6
Deler 2 ; $36/2 = 18$
Deler 3 ; $36/3 = 12$
Deler 4 ; $36/4 = 9$
Deler 6 ; $36/6 = 6$ Wortel
Deler 9 ; $36/9 = 4$ is reeds gevonden
Deler 12; $36/12 = 3$ is reeds gevonden
Deler 18; $36/18 = 2$ is reeds gevonden

We moeten dus maar lopen tot aan $\lfloor \sqrt{t} \rfloor$.

2.3 Alle priemgetallen kleiner dan de wortel aflopen

Deze procedure kan verder geoptimaliseerd worden, door enkel priemgetallen kleiner dan de wortel te gebruiken. Immers als een getal deelbaar is door een willekeurig getal x, hij automatisch ook deelbaar is door zijn priemfactoren. Met andere woorden als een getal niet deelbaar is door 2, kan het ook niet deelbaar zijn door 4, immers 2 is een priemfactor van 4. Deze methode is sneller, maar dan moeten we wel beschikken over al deze priemgetallen. Indien men zoekt naar priemgetallen in een bereik, kan men best deze methode gebruiken. Een nieuw priemgetal wordt dan toegevoegd aan de lijst van priemgetallen.

2.4 Besluit

De eerste methode is redelijk traag, maar door de bovengrens te verlagen (tot aan wortel lopen) en enkel priemgetallen te testen, kunnen we dit proces toch al versneller. Het voordeel van deze trage methodes is dat we 100% zekerheid hebben dat een getal priem is.

3 Probabilistische methodes

Deze methodes geven niet langer een zeker antwoord ze zijn juist tot op een zekere benadering.

3.1 Miller-Rabin

3.1.1 Beschrijving

Dit is een algoritme ontwikkeld door Miller en later verbeterd door Rabin, dat gebruik maakt van bekende vergelijkingen waaraan priemgetallen moeten voldoen om te testen of een gegeven getal priem is. Dit is vergelijkbaar met de methode van Fermat.

3.1.2 Wiskunde

Om het algoritme te kunnen uitleggen, hebben we onderstaande eigenschappen nodig.

Eigenschap 1 Zij p een priemgetal, dan geldt $x^2 \equiv 1 \bmod p$ als en alleen als $x \equiv \pm 1 \bmod p$.

Bewijs

$$
\begin{aligned}
x^2 \equiv 1 \bmod p &\Leftrightarrow (x+1)(x-1) \equiv 0 \bmod p && (1)\\
&\Leftrightarrow p|(x+1)(x-1) && (2)\\
&\Leftrightarrow p|(x+1)\ OF\ p|(x-1) && (3)\\
&\Leftrightarrow (x+1) \equiv 0 \bmod p\ OF\ (x-1) \equiv 0 \bmod p && (4)\\
&\Leftrightarrow x \equiv -1 \bmod p\ OF\ x \equiv 1 \bmod p && (5)
\end{aligned}
$$

Stap (1) volgt uit de nulpunten voor $x^2 - 1 = 0$. Bij stap (2) wordt gesteld dat p een deler moet zijn van $(x+1)(x-1)$, anders kan dit niet gelijk zijn aan 0 mod p. Stap (3) volgt uit de eigenschappen rond deelbaarheid met p priem (cursus blz. 77). Stappen (4) en (5) zijn dan logische gevolgen uit de voorgaande vergelijkingen.

Concreet Als $x^2 \equiv 1 \bmod p$, dan is $x \equiv \pm 1 \bmod p$.

Eigenschap 2 Laat $n = 1 + 2^s \cdot d$, met d oneven, priem zijn. Dan heeft de zogenaamde b-sequentie

$$\{b^d, b^{2d}, b^{4d}, b^{8d}, \cdots, b^{2^{s-1}d}, b^{2^s d}\} \bmod n$$

een van de volgende twee vormen

$$(1, 1, \cdots, 1, 1, 1, \cdots, 1)$$

$$(?, ?, \cdots, ?, -1, 1, \cdots, 1)$$

in modulo n, voor elke $1 < b < n$. (Het vraagteken "?"duidt een getal verschillend van ± 1 aan.) De correctheid van deze eigenschap volgt uit eigenschap 1, namelijk dat als n priem is, de oplossingen voor $x^2 \equiv 1 \bmod n$ enkel $x \equiv \pm 1$ kunnen zijn. Als de b-sequentie van n één van de volgende vormen heeft, dan is n zeker samengesteld.

$$(?, \cdots, ?, 1, 1, \cdots, 1)$$

$$(?, \cdots, ?, ?, ?, \cdots, -1)$$

$$(?, \cdots, ?, ?, ?, \cdots, ?)$$

3.1.3 Algoritme

Het algoritme bundelt deze eigenschappen in een primaliteitstest die met hoge waarschijnlijkheid kan zeggen of een gegeven getal n priem of samengesteld is. Daarvoor gaat men als volgt te werk.

1. Laat n een oneven getal zijn, en base b een random getal in het gebied $1 < b < n$. Zoek s en d door $n - 1$ zover mogelijk uit te delen met priemfactor 2 zodat $n - 1 = 2^s d$ met d oneven.
2. Stel $i = 0$ en $y = b^d \bmod n$
3. Als ($i = 0$ en $y = 1$) of $y = n - 1$, dan stopt het algoritme en is n waarschijnlijk priem. Hier is $y = n - 1 \equiv -1 \bmod n$.
4. Als $i > 0$ en $y = 1$, ga naar stap 6.
5. Doe $i = i + 1$ en als $i < s$, zet $y = y^2 \bmod n$ en ga terug naar stap 3.
6. Het algoritme stopt, met n is zeker niet priem, maar samengesteld.

In stappen 3 en 4 gebeurt het eigenlijke werk:

- Als de b-sequentie begint met een 1, is het getal waarschijnlijk priem.
- Als de b-sequentie een element $y = n - 1 \equiv -1 \bmod n$ heeft, maar niet op de laatste plaats, is n waarschijnlijk priem.
- Als $y = 1$ is op eender welke andere plaats, dan is n zeker samengesteld, want we hebben dan een b-sequentie van de eerste "slechte"vorm. De base b is hier een getuige voor n.

Concreet We nemen het startgetal $b^d \bmod n$ en kwadrateren het in iedere stap van het opstellen vande b-sequentie. n is dan priem als we een 1 tegenkomen met in de stap ervoor een -1 of allemaal 1 sinds de eerste stap.

Getuige Een base b is voor een getal n een getuige als bovenstaand algoritme n aanduidt als zijnde samengesteld.

Leugenaar Een base b is een leugenaar voor n als het n aanduidt als zijnde priem, maar dit na een test met een andere base niet zo blijkt.

3.1.4 Voorbeeld

Stel dat we het getal $n = 221$ willen testen op primaliteit. Schrijf dan $n - 1 = 220 = 2^2 \cdot 55$. Hier is $s = 2$ en $d = 55$. Kies een willekeurige base $b < n$, bijvoorbeeld $b = 174$. We bekijken de gelijkheden die de b-sequentie vormen.

- $b^d \bmod n = 174^{55} \bmod 221 = 47 \neq 1$
- $b^{2^s \cdot d} \bmod n = 174^{1 \cdot 55} \bmod 221 = 47 \neq n - 1$ voor $s = 0$
- $b^{2^s \cdot d} \bmod n = 174^{2 \cdot 55} \bmod 221 = 220 = n - 1$ voor $s = 1$

Aangezien $220 \equiv -1 \bmod n$, geldt ofwel dat 221 priem is, ofwel dat 174 een leugenaar is voor 221. Kies nog een willekeurige b, bijvoorbeeld $b = 137$. Bekijk weer de gelijkheden.

- $b^d \bmod n = 137^{55} \bmod 221 = 188 \neq 1$
- $b^{2^s \cdot d} \bmod n = 137^{1 \cdot 55} \bmod 221 = 188 \neq n - 1$ voor $s = 0$
- $b^{2^s \cdot d} \bmod n = 137^{2 \cdot 55} \bmod 221 = 205 \neq n - 1$ voor $s = 1$

Hieruit volgt dat 137 een getuige is voor het feit dat 221 samengesteld is, en dat 174 een leugenaar is voor 221.

3.1.5 Deterministisch?

Het is mogelijk deze methode om te vormen naar een deterministische methode, zodat het resultaat altijd correct is. Daarvoor moeten er genoeg bases b getest worden, zodat de kans dat n een samengesteld getal blijkt te zijn, nihil wordt. Het aantal te testen bases wordt gegeven in onderstaande stelling.

Stelling Stel $n > 1$ een oneven samengesteld getal, dan slaagt n de test als priem voor ten hoogste $(n-1)/4$ bases b met $1 \leq b < n$.

We moeten dus ten minste $(n-1)/4$ bases testen om met 100% zekerheid te kunnen oordelen of n een priemgetal is.

4 Deterministische methodes

Deze methodes om mogelijke priemgetallen te controleren gaan uit van een deterministische statenmachine en kunnen dus heel traag uitvallen, soms zelfs NP-compleet.
Maar in tegenstelling tot de probabilistische tests, ben je wel 100% zeker van het bekomen resultaat. Als een getal als priem aangeduid wordt door deze methodes, is het dat ook.

4.1 Miller test

4.1.1 Beschrijving

De Miller test is de originele versie van de Miller-Rabin test, en deterministisch. Ze steunt daarvoor echter op de nog onbewezen Riemann hypothese. Deze hypothese impliceert resultaten over de verdeling van priemgetallen.

De Miller test is echter achterhaald, en wordt daarom niet verder uitgelegd. De volgende test, AKS, is sneller en steunt bovendien niet op onbewezen aannamen.

4.2 AKS test

4.2.1 Beschrijving

De AKS test, wiens naam afkomstig is van de drie bedenkers: Manindra Agrawal, Neeraj Kayal en Nitin Saxena, is een redelijk nieuwe methode bedacht in 2004. De methode kan in polynomiale tijd, $O((\log n)^{12})$ met n het aantal cijfers van het getal, bepalen of een gegeven getal priem of samengesteld is. Het is ook het eerste snelle algoritme dat aan alle onderstaande eigenschappen voldoet.

- Algemeen, het werkt voor alle getallen;
- Polynomiaal, snelheid is beperkt door een polynomiale functie;
- Deterministisch, bij eenzelfde input zal de output altijd gelijk zijn;
- Zonder aannamen, het is niet gebaseerd op nog niet bewezen hypothesen.

Alle vorige snelle methodes voldoen maar aan drie van de vier eigenschappen.
Een recente verbetering heeft de performantie zelfs teruggebracht van $O((\log n)^{12})$ naar $O((\log n)^{6})$.

4.2.2 Wiskunde

Om het uiteindelijke algoritme te snappen, zijn deze eigenschappen nodig:

Eigenschap 1 Kleine stelling van Fermat, Voor p priem en $a \in \mathbb{N}$

$$a^p = a \bmod p$$

Eigenschap 2 Stelling van Fermat voor polynomialen

$$(x-a)^n \equiv (x^n - a) \bmod n \Leftrightarrow n \text{ is priem}$$

Dit is een afleiding van de kleine stelling van Fermat. De x is hier een symbool en wordt niet vervangen door een getal. Om de equivalentie te controleren, expandeer $(x-a)^n$ en vergelijk de coëfficiënten van de overeenkomstige machten van x.
De logica voor bovenstaande afleiding steunt op eigenschap 1 in combinatie met het binomium van Newton en de eigenschap van de binomiumcoëfficiënt die zegt dat

$$\binom{n}{k} \equiv 0 \bmod n \quad \forall k,\ 0 < k < n \Leftrightarrow n\ is\ priem$$

Eigenschap 3 Merk op dat eigenschap 2 zelf al een priemtest is, maar deze werkt in exponentiële tijd. Daarom gebruikt de AKS methode volgende equivalentie

$$(x-a)^p \equiv (x^p - a) \bmod ggd(p, (x^r - 1))$$

Wat is $(x^r - 1)$? De originele equivalentie van eigenschap 2 heeft in het slechtste geval n coëfficiënten a die moeten geëvalueerd worden. Een simpele manier om dit aantal te reduceren, is om te evalueren met modulo een polynomiaal van de vorm $x^r - 1$, met r een gepaste, kleine waarde.

Nu is de bovenstaande test nog steeds geldig voor priemgetallen, maar ook bepaalde samengestelde getallen kunnen er aan voldoen. Maar dit probleem kan opgelost worden door de vergelijking voor verschillende a $(0 < a < n)$ te testen. Als bij alle mogelijke a de vergelijking geldig is, is n zeker een priemgetal.

4.2.3 Algoritme

Het uiteindelijke algoritme wordt hieronder uitgelegd. We starten van een input $n > 1,\ n \in \mathbb{N}$.

Stap 1 Als $n = a^b$, dan is n samengesteld, voor $a, b \in \mathbb{N},\ 1 < b < \log_2 n$.
We controleren alle mogelijke waarden voor b en kijken of $a = n^{1/b} \in \mathbb{N}$ is. Als dit zo is, is n een samengesteld getal, want n is dan een veelvoud van a, en dus niet priem.

Stap 2 Zoek de kleinste r zodat $ord_r(n) > (\log_2(n))^2$. De uitdrukking $ord_r(n)$ staat voor de multiplicatieve orde van n modulo r en is het kleinste getal k zodat $n^k \equiv 1 \bmod r$. Hier zoeken we dus een gepaste waarde voor r om dan deze r later te gebruiken.

Voorbeeld voor $ord_r(n)$: $ord_7(4) = 3$ want

- $4^2 = 16 \equiv 2 \bmod 7 \neq 1 \bmod 7$
- $4^3 = 64 \equiv 1 \bmod 7 \rightarrow$ OK

Stap 3 Als $1 < ggd(a, n) < n$ voor alle $a \leq r$, dan is n samengesteld. Hier controleren we of er een $a \leq r$ bestaat die een deler is van n. Indien ja, is n samengesteld, want we hebben een deler ($\neq 1$ of $\neq n$) gevonden. Indien niet, kunnen we verder naar stap 4.

Stap 4 Als $n \leq r$, dan is n priem. Dit is enkel geldig voor $n \leq 5690034$. Meestal zal $n > r$ zijn en gaan we verder naar stap 5.

Stap 5 $\forall a,\ 1 \leq a < \lfloor \sqrt{\phi(r)} \log_2(n) \rfloor$ geldt:

$$(x + a)^n \neq (x^n + a) \bmod ggd(n, x^r - 1) \Rightarrow n \textit{ samengesteld}$$

In deze stap gebruiken we de test uit eigenschap 3 om alle a van 1 tot $\lfloor \sqrt{\phi(r)} \log_2(n) \rfloor$ te controleren, om zeker te zijn dat een samengesteld getal niet als priem wordt aangeduid.

Hoe? Het berekenen en testen van de vergelijking gebeurt niet helemaal zoals hierboven beschreven staat. Herinner je je $(x^r - 1)$? Deze wordt gebruikt om beide leden (na invullen van n en expansie van machten) te delen. Dit gebeurt via een polynomiaaldeling, oftewel een staartdeling van veeltermen.

Het resultaat is een veel eenvoudigere polynomiaal waarvan we de modulo kunnen berekenen met polymod, modulo voor veeltermen. Bij polymod wordt iedere coëfficiënt van de verschillende termen van de veelterm modulo gedaan. Een voorbeeld:

$$(5x^3 + 2x^2 + 3x + 1) \bmod 3 = (2x^3 + 2x^2 + 0x + 1)$$

Na deze stap moeten we enkel nog controleren of de equivalentie voldoet, maar dit kan ook wat sneller. Trek beide bekomen resultaten van elkaar af, vul a in en controleer of de uitkomst equivalent is met 0 mod n.

Stap 6 Als we alle iteraties in stap 5 doorlopen hebben, en het getal n is niet aangeduid als zijnde samengesteld, kunnen we met 100% besluiten dat n inderdaad priem is.

4.2.4 Voorbeeld