Van tientallig naar tweetallig en weer terug.

Wobien Doyer
Lieke de Rooij

In deze puzzel maken we rijtjes getallen $a(0)$ tot en met $a(i) = 1$.
Het algoritme om $a(j + 1)$ te krijgen uit $a(j)$ is als volgt:

1) Schrijf $a(j)$ in het 2-tallig stelsel, maar beschouw dat als rij cijfers in het 10-tallig stelsel.
2) Plaats naar keuze een aantal + tekens tussen die enen en nullen en bepaal de 10-tallige som $s(j)$ die je zo hebt verkregen.
3) Stel nu $a(j + 1) = s(j)$.

Zo ontstaat een rij $a(0), a(1), \ldots a(i)$ met $a(i) = 1$
We zoeken een rij, beginnend met een bepaalde $a(0)$, waarvoor i zo klein mogelijk is.

Voorbeeld: $a(0) = 19$, dus 2-tallig is dat 10011. Dit beschouwen we nu als een rij cijfers in het 10-tallig stelsel. Hier zetten we een aantal plus-tekens tussen de cijfers en bepalen de 10-tallige som.
Bijvoorbeeld $10 + 0 + 1 + 1 = 12$, of $1 + 0 + 0 + 1 + 1 = 3$ of $100 + 11 = 111$. Dus $s(0)$ kan o.a. worden 12 , 3 of 111.
$a(1)$ is daarvan de binair geschreven versie, dus 1100, 11 of 1101111
Met 1100 en 11 kunnen we hiervan de som 2 maken, dus $s(1) = 2$, zodat $a(2)$= 10. En dan $1 + 0 = 1$, dus $s(2) = 1$ en dus ook $a(3) = 1$. Dus $i = 3$.

Nu blijkt dat als we dit voor elk willekeurig startgetal $a(0)$ handig aanpakken, we altijd kunnen krijgen $a(i) = 1$ met $i \leq 3$.

Stelling:
Voor alle startgetallen $\boldsymbol{a(0)}$ is het mogelijk een $\boldsymbol{a(i) = 1}$ te vinden met $\boldsymbol{i \leq 3}$.

We zullen het aantal enen in $a(0)$ noteren als $e(0)$, of gewoon e. Voor de andere verkregen getallen $a(j)$ kan je het aantal enen noteren als $e(j)$

Voor alle opgaven vragen we een antwoord+bewijs of toelichting.

Opgave 1:

Bepaal de kleinst mogelijke i voor de getallen $a(0) = 1$ tot en met $a(0) = 15$.

Uitwerking opgave 1:

We geven eerst een lijstje met kleinst mogelijke waarden van i. Daarna volgen de bewijzen en toelichtingen.

$\boldsymbol{a(0)}$:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\boldsymbol{i}$:	0	1	2	1	2	2	3	1	2	2	3	2	3	3	2

In de uitwerking hieronder noteren we 2-tallig geschreven getallen tussen aanhalingstekens, dus 5=”101”

$a(0) = 1 =$ "1": We zoeken een $a(i) = 1$ en vinden die al bij $a(0)$. Dus $i = 0$.
Merk op dat $a(0) = 1$ het enige startgetal is dat $i = 0$ kan hebben. Als we dus voor een $a(0) > 1$ een oplossing vinden met $i = 1$ dan moet dat de optimale oplossing zijn.

$a(0) = 2 =$ "10": optellen van de cijfers levert $1 + 0 = 1$, dus $a(1) = 1$ en $i = 1$. kleiner kan niet.

$a(0) = 3 =$ "11": optellen van de cijfers levert $1 + 1 = 2$, dus $a(1) = 2$. We hebben dan nog één stap nodig naar $a(2) = 2$ en $i = 2$. De enige andere start is "11" op te tellen als 11 en dat levert een grotere i.

$a(0) = 4 =$ "100": optellen van de cijfers levert $1 + 0 = 1$, dus $a(1) = 1$ en $i = 1$. Kleiner kan niet.
Merk op dat de 2-machten 2 en 4 beide $i = 1$ opleveren. Dat geldt natuurlijk voor alle 2-machten: als $a(0) = 2^k$ dan is het binair geschreven getal een 1 met een of meer nullen erachter, en de optelling kan dan altijd 1 opleveren, zodat $a(1) = 1$. We kunnen in het vervolg van deze lijst bij tweemachten ineens $i = 1$ opschrijven.
Omgekeerd: als de optelling van de binaire cijfers van $a(0)$ van 1 oplevert dan moet $a(0)$ een tweemacht zijn. Dat betekent dat als $a(0)$ gèèn tweemacht is $i = 1$ niet mogelijk is. Vinden we dus een oplossing met $i = 2$ voor een niet-tweemacht dan moet dat de optimale oplossing zijn.

$a(0) = 5 =$ "101": optellen van de cijfers levert $a(1) = 2 =$ "10". Weer optellen levert $a(2) = 1$ en $i = 2$.
Omdat 5 geen tweemacht is is dat optimaal.
Merk op dat 3 en 5 beide sommen zijn van twee tweemachten, en dat ze beide $i = 2$ opleveren. Dat geldt natuurlijk voor alle sommen van 2 verschillende tweemachten. Die hebben altijd een binaire notatie met precies 2 enen. Dat kan altijd worden opgeteld tot $a(1) = 2$ en $a(2) = 1$ en dus kunnen we in het vervolg van deze lijst bij sommen van 2 verschillende tweemachten ineens $i = 2$ invullen.

$a(0) = 6 = 2 + 4$: Dit is de som van 2 tweemachten dus $i = 2$.

$a(0) = 7 =$ "111": Optellen van de cijfers levert $a(1) = 3$. We zagen al dat we vanaf 3 nog 2 stappen nodig hebben dus we krijgen zo $i = 3$. De 2 andere optellingen vanuit = "111" zijn 11+1=12 en 111. 12 is de som van twee verschillende tweemachten, en heeft vanaf $a(1) = 12$ ook nog 2 stappen nodig, met dus ook $i = 3$. Vanaf 111 zijn meer stappen nodig. $i = 3$ is dus optimaal.

$a(0) = 8$: Dit is een tweemacht dus $i = 1$.

$a(0) = 9$: Dit is de som van 2 tweemachten 8+1, dus $i = 2$.

$a(0) = 10$: Ook dit is de som van 2 tweemachten 8+2, dus $i = 2$.

$a(0) = 11 =$ "1011": De mogelijke optellingen hiervan zijn 3, 12, 21, 102 en 1011. Voor 3 en 12 zijn, net als bij 7, nog 2 stappen nodig, met $i = 3$. 21, 102 en 1011 zijn geen tweemachten en hebben dus minstens 2 stappen nodig. $i = 3$ is dus optimaal.

$a(0) = 12$: Dit is de som van 2 tweemachten 8+4, dus $i = 2$.

$a(0) = 13 =$ "1101": De mogelijke optellingen hiervan zijn 3, 12, 102,111 en 1101. Voor 3 en 12 zijn, net als bij 7, nog 2 stappen nodig, met $i = 3$. 102, 111 en 1101 zijn geen tweemachten en hebben dus minstens 2 stappen nodig. $i = 3$ is dus optimaal.

$a(0) = 14 = 1110$: De mogelijke optellingen hiervan zijn 3, 12, 21,111 en 1110. Voor 3 en 12 zijn, net als bij 7, nog 2 stappen nodig, met $i = 3$. 102, 21, 111 en 1110 zijn geen tweemachten en hebben dus minstens 2 stappen nodig. $i = 3$ is dus optimaal.

$a(0) = 15 =$ "1111": Optellen van de cijfers levert $a(1) = 4$. Dat is een tweemacht en er is dus nog 1 stap nodig, met $i = 2$. Omdat 15 geen tweemacht is is dat optimaal.

Voor $a(0) = 31$, binair is dat 11111 met 5 enen ($e = 5$), kunnen we kiezen voor $s(0) = 5$, dus $a(1) = 101$ dan kunnen we kiezen voor $s(1) = 2$, zodat $a(2) = 10$, zodat $s(2) = 1$, en $a(3) = 1$. Dus $i = 3$.

Maar er zijn ook getallen met 5 enen waarvoor $i < 3$ mogelijk is.

Opgave 2: Geef hiervan een voorbeeld.
Uitwerking opgave 2:

$a(0) = 87 =$ "1010111" met optelling $a(1) = 10 + 10 + 11 + 1 = 32$. Dit is een 2-macht dus $a(2) = 1$ met $i = 2$.

We kunnen ook direct kijken naar het aantal enen ($= e$) in het binair geschreven startgetal $a(0)$.

Opgave 3: Bepaal de kleinst mogelijke i voor alle startgetallen met binair geschreven e enen voor $e = 1$ tot en met $e = 15$.[1])
Het moet dan dus gelden voor alle mogelijke $a(0)$ met e enen.

Uitwerking opgave 3.
We geven weer eerst een lijstje, daarna de toelichtingen en bewijzen.

e:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
i:	0	2	3	2	≤ 3	3	2	2	3	≤ 3	2	3	2	2	3

Het probleem is nu dat we wel weten hoeveel enen er in de binaire weergave staan, maar niet waar er nullen tussen die enen staan. Met andere woorden, we weten niet welke enen in de optelling kunnen figureren als eenheid, tiental of andere 10-macht. Bij de kleinere waarden van $a(0)$ uit opgave 1 gebruikten we alleen eenheden (hoewel er wel gevallen waren waarin het gebruik van 10-tallen dezelfde waarde van i opleverde). Maar bij grotere aantallen enen in de binaire weergave van $a(0)$ zal blijken dat we ook machten van 10 moeten gebruiken.
We weten daarbij niet hoe de enen en nullen in $a(0)$ gegroepeerd zijn, maar wat we wel weten is de waarde modulo 9 van de optelling is. Er geldt natuurlijk altijd dat de modulo-waarde van de optelling gelijk is aan de modulo-waarde van het aantal enen. En dus geldt: $a(1) \equiv e$ mod 9.
Dat betekent dat als e een drievoud is $a(1)$ ook een drievoud is en dus nooit een tweemacht. Er zijn dus nog na $a(1)$ nog minstens 2 stappen nodig zodat $i \geq 3$.

Conclusie: als e een 3-voud is geldt $i \geq 3$. We zullen dit in de volgende lijst diverse malen gebruiken.

$e = 1$: de binaire notatie van $a(0)$ bevat één 1:
$a(0)$ is dan 1 of een tweemacht.
We zagen in opgave 1 dat dan geldt $i = 0$ (als $a(0) = 1$) of $i = 1$ (als $a(0) = 2^k$ met $k > 0$. Dus $i \leq 1$.
We zagen hiervan voorbeelden in opgave 1 bij $a(0) = 1, 2, 4$ en 8.

$e = 2$: de binaire notatie van $a(0)$ bevat 2 enen. Om a(1) te bepalen tellen we die 2 enen (+nullen) op dus $a(1) = 2$.
Dat is een 2-macht dus $a(2) = 1$ en $i = 2$.
We zagen hiervan voorbeelden in opgave 1 bij $a(0) = 3, 5, 6, 9, 10$ en 12.

$e = 3$: de binaire notatie van $a(0)$ bevat 3 enen. Dus $a(0) = 2^p + 2^q + 2^r$ (met p, q en r ongelijk).
Om a(1) te bepalen kunnen we die 3 enen (+nullen) optellen met $a(1) = 3$. We zagen al dat als a(0)=3 geldt: a(2)=1, dus als a(1)=3, dan a(3)=1, met $i = 3$.
Kan het beter door ook 10-tallen te gebruiken? Nee, zoals we hierboven zagen geldt: als als e een 3-voud is geldt $i \geq 3$. Dus $i = 3$.
We zagen hiervan voorbeelden in opgave 1 bij $a(0) = 7, 11, 13$ en 14.

$e = 4$: Tellen we de 4 enen op dan is a(1) een 2-macht., dus $a(2) = 1$, met $i = 2$.
Kan het beter door ook 10-tallen te gebruiken? Nee, $i = 1$ kan alleen als $a(0)$ een tweemacht is, en dan is e=1.
We zagen in opgave in opgave 1 het voorbeeld $a(0) = 15$.

$e = 5$: Tellen we de 5 enen op dan is $a(1) = 5$. Dat is de som van 2 tweemachten (1 en 4) dus we hebben nog 2 stappen nodig met $i = 3$.
Kan het beter door ook 10-tallen te gebruiken? Ja, heel vaak wel. Als we van die 5 enen er 3 kunnen gebruiken als 10-tal en 2 als eenheid, dan hebben we $a(1) = 32$, en dat is een 2-macht, zodat we na $a(1)$ nog maar één stap nodig hebben, met $i = 2$. Maar als $a(0) = 31 =$ "11111" lukt dat niet omdat je met 5 binaire cijfers hebt en je dus nooit 3 10-tallen kan maken. Er geldt dus voor $e = 5$: $i \leq 3$.

---

[1]) We doen dit vrij uitgebreid omdat dat kan helpen bij de extra opgaven 6 en 7.

Van hier af zijn er geen voorbeelden meer in opgave 1, omdat de waarden van $a(0)$ daar minder dan 5 binaire cijfers hebben.

$e = 6$: Tellen we de 6 enen op dan krijgen we $a(1) = 6$. Dat is de som van 2 tweemachten, dus we hebben nog 2 stappen nodig, met $i = 3$.
Kan het beter door ook 10-tallen te gebruiken? Nee, omdat 6 een 3-voud is is $i \geq 3$.
Dus $i = 3$.

$e = 7$: Met de tot nu toe gebruikte methode (steeds alle enen optellen) komen we op $i \leq 3$, maar dat kan beter:
We kunnen optellen (afhankelijk van de vraag of 10 in het rijtje enen en nullen voorkomt of niet): 10+6x1(+nullen) of 11+5x1 (+ nullen), met in beide gevallen $a(1) = 16$. Dat is een tweemacht dus we zijn na $a(1)$ in één stap klaar.
Dus $i = 2$.

$e = 8$: Hier kunnen we weer de 8 enen optellen zodat $a(1) = 8$. Dat is een 2-macht dus na $a(1)$ in één stap klaar.
Dus $i = 2$.

$e = 9$: 0ok hier kunnen we de 9 enen optellen met $a(1) = 9$. Dit is de som van 2 tweemachten 1+8, dus zijn we na $a(1)$ in twee stappen klaar, met $i = 3$. Kleiner kan niet omdat e een drievoud is.

$e = 10$: Als we de 10 enen optellen hebben we $a(1) = 10$. Dat is de som van 2 tweemachten 8+2, dus zijn we na $a(1)$ in 2 stappen klaar, met $i = 3$.
Kan het beter door ook 10-tallen te gebruiken? Ja, vaak wel. Als we de 10 enen kunnen optellen als 6 tientallen en 4 eenheden dan hebben we $a(1) = 64$. Dat is een tweemacht, dus zijn we na $a(1)$ in één stap klaar, met $i = 2$.
Maar dat lukt niet altijd, want als de binaire weergave van $a(0)$ minder dan 12 cijfers heeft kunnen we nooit 6 tientallen maken. Dus $i \leq 3$.

$e = 11$: We kunnen altijd de som 128 maken. Van de 11 enen moet er dan één een honderdtal worden, 2 enen moeten tientallen worden en de rest, dus 8 eenheden. Dat lukt als volgt:
Het meest linkse cijfer is altijd een 1. Omcirkel de eerste 3 cijfers (zie figuur). Dat kan zijn 100, 110, 101 of 111.
We hebben nu dus 1 honderdtal, 0 of 1 tientallen en 0 of 1 eenheden.
Als er nog geen tientallen zijn, omcirkel dan de 2 eerste groepjes van 2 cijfers beginnend met een 1. Als er wel al een 10-tal is, omcirkel dan één zo'n groepje.
Nu heb je één honderdtal, 2 tientallen en tussen de 0 en 3 eenheden.
Zet nu overal een plus tussen, behalve in de omcirkelde rijtjes. (zie figuur).
De waarde van de optelling is nu altijd 128: er is één hondertal, 2 tientallen, en de rest van de 11 enen, dus 8, zijn eenheden.
$a(1)$ is dus een tweemacht, dus na $a(1)$ zijn we in één stap klaar, met $i = 2$

1 0 1 + 0 + 0 + 1 0 +1+ 1 1 + 0 + 0 + 1 + 1 + 0 + 1 + 0 + 1 + 1

$e = 12$: Hier kunnen we weer alle enen optellen, met $a(1) = 12$. Dat is de som van 2 tweemachten 4 en 8, dus na $a(1)$ zijn we in 2 stappen klaar, met $i = 3$.
Minder dan 3 kan niet, omdat 12 een 3-voud is.

$e = 13$: Nu kunnen we altijd de som 256 maken. En omdat 256 een tweemacht is is dan $i = 2$.
We moeten dan twee groepjes van 3 cijfers beginnend met 1 maken voor de 2 honderdtallen.
Maar als we daarvoor , zoals bij $e = 11$, de eerste 2 groepjes van 3 nemen dan kan het fout gaan, zoals te zien is in de eerste (rode) figuur.

1 0 1 + 1 0 1 + 1 1 + 1 1 + 1 1 + 1 1 + 1

1 0 1 + 1 0 + 1 1 1 + 1 1 + 1 1 + 1 1 + 1

Als beide groepjes van 3 101 zijn en daarna staan er geen nullen meer zijn dan kunnen we maar 4 tientalllen maken.
Dat kunnen we als volgt oplossen:
Noteer de aantallen honderdtallen, tientallen en eenheden die we nodig hebben, dus 2 5 6. Bij het "vullen" van de honderdtallen houden we die aantallen bij en controleren of het aantal nog nodige eenheden niet kleiner worden dan het aantal tientallen.
Als het omcirkelde groepje van 3 cijfers 111, 110 of 100 is blijft het aantal nog nodige eenheden groter dan het aantal nog nodige tientallen. Bij de eerste keer 101 worden ze gelijk. Een 2e keer 101 mag dus niet.
Maar als we alleen de eerste 2 cijfers omcirkelen dan wordt het aantal nog nodige eenheden weer groter dan het aantal nog nodige tientallen.
Gevolg: als we de 2 honderdtallen "gemaakt" hebben dan is het aantal nog nodige tientallen ≤ het aantal nog nodige eenheden. En omdat het aantal enen in de rij precies genoeg zijn om 256 te maken kunnen we daarna groepjes van 2 cijfers omcirkelen tot er 5 tientallen zijn en houden precies genoeg losse enen over.

$e = 14$: Van 14 enen kunnen we altijd een optelling maken die 2 tientallen oplevert en 12 eenheden, samen 32. Dus $i = 2$.

$e = 15$: Nu kunnen we altijd 2 tientallen en 13 eenheden maken, samen 33=32+1, dus de som van twee 2-machten. Dus $i = 3$. Kleiner kan niet omdat e een drievoud is.

Voor bepaalde waarden van e zijn er startgetallen waarvoor we reeds met $i \leq 1$ in alle gevallen klaar zijn.
Opgave 4: Voor welke waarden van e geldt dit altijd?
Uitwerking opgave 4:
Zoals we al constateerden in opgave 2 en 4 is $i = 0$ als $a(0)$=1 en $i = 1$ als $a(0)$ een tweemacht is. In beide gevallen geldt $e = 1$.

Opgave 5a.
Idem voor startgetallen met e enen waarvoor het altijd lukt met $a(2) = 1$, dus $i = 2$.
Waar moet e dan aan voldoen?
Uitwerking opgave 5a:
Ook constateerden we dat $i = 2$ als $a(0)$ de som is van twee verschillende tweemachten. In dat geval geldt $e = 2$. Maar we vonden ook allerlei grotere waarden van e waarvoor geldt dat $i = 2$.

Bij opgaven 4 en 5a ging het om bepaalde eigenschappen van e. Maar of je daarmee alle mogelijkheden met $i < 3$ hebt beschreven is moeilijk te bewijzen. Er zijn echter wel bepaalde waarden van e waarvoor het zeker niet lukt om altijd een $a(i) = 1$ te vinden met $i < 3$.

Opgave 5b: Voor welke waarden van e is dat zeker altijd het geval?
Uitwerking opgave 5b: Ook constateerden we dat als e een drievoud is het onmogelijk is dat $i < 3$ (zie voor het bewijs de inleiding van de uitwerking van opgave 3).

Twee toegiften buiten de puntentelling:

Tip bij opgave 6: Dit is op te lossen met in de optellingen getallen met maximaal 2 enen en nullen.
Opgave 6. Bewijs de stelling dat er altijd een $i \leq 3$ is te vinden met $a(i) = 1$.
Tip bij opgave 7: Dit is op te lossen met in de optellingen getallen met maximaal 4 enen en nullen.
Opgave 7:
Geef een overzicht van de minimumwaarden van i bij alle $e(0)$ en bewijs dat. Het bewijs is lastig, maar elke poging daartoe stellen we zeer op prijs.

Er waren twee inzenders die goede resultaten gevonden hebben voor de toegiften.
Gerard Bouwhuis vond een volledige oplossing voor opgave 6. Zijn methode is, net als in onze

oplossing, om 2 tweemachten te zoeken die opgeteld mod 9 dezelfde waarde hebben als e (dus $e \equiv 2^p + 2^q$ mod 9). Maar waar Gerard tweemachten zoekt met $0 \leq p - q \leq 3$ kiezen wij $2^p > e$, maar met een zo klein mogelijk verschil, en q zo als mogelijk is gezien de gewenste waarde modulo 9.
Gerard vond ook in opgave 7 de conclusies die hieronder worden besproken, maar, zoals hij zelf aangeeft kreeg hij het bewijs niet helemaal rond.
Frans van Hoeve sloeg opgave 6 over, maar vond een volledige oplossing van opgave 7 (en daarmee bewijs je natuurlijk ook opgave 6). Zijn methode is net een beetje anders dan de onze, maar loopt er globaal gesproken parallel aan. Wel is zijn notatie en bewijsvoering een stuk abstracter dan de onze.

Uitwerking opgave 6:
Dit lijkt, als we ook opgave 7 uitwerken, overbodig, maar de uitwerking is, vergeleken bij die van opgave 7, relatief eenvoudig, en daarom toch de moeite waard.
We gaan proberen voor alle waarden van e een oplossing te vinden met $i \leq 3$. We hebben daarvoor een algoritme gevonden waarbij we kunnen bewijzen dat dat altijd lukt.
We tonen daarbij aan dat voor elke e er een som $s(0)$ is te maken die een 2-macht of de som van twee 2-machten is. En daarbij worden in die som alleen getallen van maximaal twee cijfers gebruikt, dus 0,1,10 of 11.

Hieronder stap voor stap het algoritme, met steeds zonodig het bewijs erbij dat die stap mogelijk is.
We gaan een p en een q zoeken zo, dat $2^p + 2^q \equiv e$ mod 9, met $e \leq 2^p + 2^q$, en een getal $t \leq \frac{1}{2}e$ zo dat $e + 9t = 2^p + 2^q$

1. Laat e tussen de twee 2-machten 2^{k-1} en 2^k in liggen. Als $e = 2^{k-1}, 2^{k-1} + 1, 2^{k-1} + 2$ of 2^k dan is e een tweemacht of de som van 2 tweemachten, en dan is $i = 2$ of $i = 3$ en zijn we klaar.
2. We moeten dus alleen nog bewijzen dat als $2^{k-1} + 2 < e < 2^k$ geldt $i \leq 3$.
   Als $2^k - e$ geen drievoud is kies dan $p = k$, anders $p = k + 1$.
   (ga na dat in geen van beide gevallen nu $2^p - e$ een drievoud is)
   Dan is $2^{k-1} \leq e - 3$, dus $2^{k+1} \leq 4e - 12$, en omdat $p \leq k$ ook $2^p \leq 4e - 12$,
3. Kies voor q een zo klein mogelijke waarde, zodanig, dat $2^p + 2^q \equiv e$ mod 9.
   (Omdat er reeksen 2-machten zijn met alle waarden modulo 9 behalve de 3-vouden en we ervoor gezorgd hebben dat $2^p - e$ geen drievoud is, lukt dat altijd. En omdat alle mogelijke waarden modulo 9 voorkomen tussen $2^0 = 1$ en $2^5 = 32$ is altijd $2^q \leq 32$.)
4. We gaan t keer een 10 of een 11 gebruiken in de optelling.
   We berekenen $t = \frac{1}{9}(2^p + 2^q - e)$, zodat $e + 9t = 2^p + 2^q$
   a. t is geheel en niet negatief: Omdat we gezorgd hebben dat $2^p + 2^q \equiv e$ mod 9 is $2^p + 2^q - e$ een 9-voud, zodat t een geheel getal is. Omdat $e < 2^p$ is t ook positief.
   b. Als er geen nullen tussen de e enen in de 2-tallige weergave van $a(0)$ staan kunnen we enen alleen de waarde 10 geven door een 11 te gebruiken. Dan mag t dus niet groter zijn dan de helft van e.
      Om dat te garanderen maken we een afschatting van t.
      In punt 1 zagen we dat $2^p \leq 4e - 12$ en in punt 2 dat $2^q \leq 32$. Dat vullen we in in de formule: $t = \frac{1}{9}(2^p + 2^q - e) \leq \frac{1}{9}(4e - 12 + 32 - e) = \frac{1}{9}(3e + 20)$.
      Als $\frac{1}{9}(3e + 20) \leq \frac{1}{2}e$ weten we dat t klein genoeg is, want dan is $t < \frac{1}{9}(3e + 20) \leq \frac{1}{2}e$.
      Dus moet $\frac{1}{9}(3e + 20) \leq \frac{1}{2}e$ dus $6e + 40 \leq 9e$ dus $3e \geq 40$ of $e \geq 14$.
5. Als $e \geq 14$ kunnen we dus in de optelling van de e enen in $a(0)$ er t als tiental gebruiken, zodat de optelling t tientallen en $e - t$ eenheden bevat, met $a(1) = e - t + 10t = e + 9t =$

$= e + 9\frac{2^p+2^q-e}{9} = e + 2^p + 2^q - e = 2^p + 2^q$. In dat geval is $a(1)$ de som van 2 tweemachten en geldt $i = 3$.

Voor $e \leq 15$ hebben we in opgave 3 al aangetoond dat voor $e \leq 15$ geldt $i \leq 3$, dus nu hebben we bewezen dat voor alle waarden van e geldt: $i \leq 3$. Q.E.D.

Uitwerking opgave 7:

In opgave 6 zagen we dat we in alle gevallen kunnen zorgen dat $i \leq 3$.
Er zijn echter ook veel waarden van e (zoals we zullen zien de meerderheid) waarbij het mogelijk is om te zorgen dat $i = 2$.

In opgave 7 gaan we precies uitzoeken welke waarden van e dat zijn.

We zullen daarbij gebruik maken van wat we in opgave 3 zagen bij e = 11 en 13.
Maar we zullen dan moeten vaststellen onder welke voorwaarden dat lukt.

Bij 11 en 13 schreven we op een handige manier de plussen tussen de 2-tallige voorstelling van een getal, zodanig dat er een 2-macht ontstaat. Als dat lukt is $a(1)$ een tweemacht en geldt $i = 2$.

In de inleiding van de uitwerking van opgave 3 lieten we zien dat $a(1) \equiv e$ mod 9.
Dat betekent dat voor de tweemacht 2^k die we kunnen gebruiken moet gelden: $2^k = e$ mod 9

We maken daarom een lijstje van de waarden van de tweemachten mod 9:

$2^0 \equiv 1$ mod 9	$2^1 \equiv 2$ mod 9	$2^2 \equiv 4$ mod 9	$2^3 \equiv 8$ mod 9	$2^4 \equiv 7$ mod 9	$2^5 \equiv 5$ mod 9
$2^6 \equiv 1$ mod 9	$2^7 \equiv 2$ mod 9	$2^8 \equiv 4$ mod 9	$2^9 \equiv 8$ mod 9	$2^{10} \equiv 7$ mod 9	$2^{11} \equiv 5$ mod 9
$2^{12} \equiv 1$ mod 9	$2^{13} \equiv 2$ mod 9	$2^{14} \equiv 4$ mod 9	$2^{15} \equiv 8$ mod 9	$2^{16} \equiv 7$ mod 9	$2^{17} \equiv 5$ mod 9

enzovoort.

Zoals te verwachten was worden de modulo waarden periodiek herhaald en komen alle andere waarden modulo 9 allemaal voor behalve de 3-vouden.
Als e geen drievoud is kunnen we dus kiezen uit een aantal verschillende tweemachten die steeds een factor 2^6 van elkaar verschillen.

We gaan bewijzen:
I: Als $e = 1$ en $a(0) = 1$ dan is $i = 1$
II: Als e een drievoud is geldt: $i = 3$
III: Bij de waarden $e = 5, 10, 17$ en 25 horen zowel waarden van $a(0)$ waarvoor $i = 3$ als waarden van $a(0)$ waarvoor $i = 2$.
IV: Voor alle andere gevallen geldt $i = 2$.

Inleiding

Het bewijs dat we hier presenteren is niet eenvoudig en erg lang. We hebben geprobeerd het zo toegankelijk mogelijk te maken.
We hoeven ons niet meer druk te maken om de drievouden. We zagen al in opgave 3 dat als e een drievoud is in elk geval geldt $i \geq 3$, en in opgave 6 dat voor elke waarde van e geldt $i \leq 3$. Dus is punt II hierboven al bewezen. Dat als $a(0) = 1$ geldt $i = 1$ is triviaal en zagen we al in opgave 1.

Wat we nog moeten bewijzen komt neer op: Op een paar uitzonderingen na geldt voor alle niet-drievouden $i = 2$. We zullen dat bewijzen door een algoritme te beschrijven waarmee je voor elke waarde van e (behalve drievouden en 4 uitzonderingen) met elke rij nullen en enen met e enen kan zorgen dat $i = 2$.

Als e een tweemacht is zijn we meteen klaar. We krijgen dan $a(1) = 2^k$ en dus $i = 2$. Maar er zijn veel meer manieren om $a(1) = 2^k$ te krijgen. Bijvoorbeeld $a(1) = 64$ kan met e = 64, maar ook met bijvoorbeeld e = 28. In opgave 6 zagen we al dat het altijd goed gaat als $t \leq \frac{1}{2}e$ en we willen t tientallen en de rest eenheden. Dus als je de 28 enen verdeelt in 4 tientallen en 24 eenheden dan heb je $a(1) = 64$. En zo zijn er natuurlijk nog veel meer. Het aantal mogelijkheden wordt nog veel groter als je ook honderdtallen en duizendtallen gaat gebruiken. Het wordt dan wel een stuk ingewikkelder om te bepalen welke eisen je moet stellen om te zorgen dat het altijd lukt.

In het voorbeeld hierboven (we maken 64 met 28 enen) is het van belang dat 28 ≡ 64 mod 9, dus kunnen we een lijst maken van alle waarden 64 mod 9 en uitzoeken bij welke daarvan een waarde van e hoort die we kunnen gebruiken om 64 te maken. Daarop is dit algoritme gebaseerd.

Die lijst noteren we als volgt:
Reeks van $64 = 2^6$ met $64 \equiv 1 \bmod 9$:

$[c_1][c_0] = [6][4]$ cijfersom $= 10 = e$
$[c_1][c_0] = [5][14]$ cijfersom $= 19 = e$
$[c_1][c_0] = [4][24]$ cijfersom $= 28 = e$
$[c_1][c_0] = [3][34]$ cijfersom $= 37 = e$
⋮

In de alinea hieronder hebben we enkele termen vet gedrukt. Ze worden toegelicht bij de Definities op blz.Met een 9.

In de reeks van 64 is [6][4] de **verdeling** van 64 in 6 tientallen en 4 eenheden. De volgende verdelingen ontstaan door steeds één tiental te wisselen voor 10 eenheden, dus **standaard verlenging**. De **cijfersommen** van die verdelingen zijn de waarden van e die we ermee kunnen maken. Alle verdelingen hebben dezelfde **getalwaarde** 64.
We hebben de eerste verdeling blauw gekleurd omdat $e = 10$ één van de 4 uitzonderingen e=5, 10, 17 of 25 is. Inderdaad voldoet [6][4] niet, omdat er dan meer tientallen zijn dan eenheden.

Voor alle getallen die geen 3-voud zijn, geen tweemacht en niet één van de 4 uitzonderingen kunnen we volgens onderstaand algoritme te werk gaan. Dat zal leiden tot een verdeling in groepjes van enen en nullen die we samen nemen bij de optelling in de binaire versie van $a(0)$.

Het algoritme bestaat uit 4 stappen die we in het hoofdstuk Algoritme op blz. 9 verder uitwerken:

Stap 1: Je kiest de tweemacht 2^m die past bij jouw waarde van e (geen drievoud of tweemacht).

Stap 2: Je vindt een klein stukje (tot en met de verdeling waarvan de cijfersom e gelijk is aan de kleinste waarde waarvoor 2^m gekozen kan worden) van de bij de tweemacht behorende reeks verdelingen in eenheden, tientallen, honderdtallen etc. die steeds voor één waarde van e geschikt zijn.
Als een waarde blauw is gemarkeerd dan is dat één van de uitzonderingen $e = 5, 10, 17$ of 25. Als die voorkomen zitten ze altijd in het stukje van de reeks dat in Stap 2 staat.
Als jouw waarde van e erbij staat en is hij niet blauw gemarkeerd dan noteer je de verdeling die bij e past en slaat stap 3 over.
Voor grotere waarden van e moet je de rest van de reeks zelf uitschrijven. Daarvoor noteer je de laatste verdeling en vindt je verdere aanwijzingen in stap 3.
Natuurlijk kunnen we niet voor alle tweemachten die gebruikt kunnen worden (dat zijn ze allemaal) een klein stukje uitschrijven in stap 2. Bij 4096 duikt een resultaat op waarmee alle grotere tweemachten kunnen worden behandeld. Het karwij waarmee we bezig zijn is dus (gelukkig!) eindig.

Stap 3: Vanaf de verdeling die je had genoteerd vervolg je de rij verdelingen met standaard verlenging tot jouw waarde van e en noteert die laatste verdeling.

Stap 4: In stap 4 ga je de reeks enen en nullen uit $a(0)$ gebruiken om de optelling te maken met som 2^k. Om bij te houden hoeveel eenheden van elke soort je nog nodig hebt trek je steeds de getallen die je maakt in de reeks af van de verdeling van de tweemacht.

definities

1. Met een **verdeling** van een getal bedoelen we hier een verdeling in eenheden, tientallen, honderdtallen etc, die kan afwijken van de gebruikelijke. Voor de notatie gebruiken we vierkante haken. Dus b.v. het getal 325 schrijven we [3][2][5], maar ook[2][12][5] etc.
   We kunnen verdelingen ook optellen of aftrekken. We doen dat altijd zonder wisselen, dus bijvoorbeeld [12][5]+[3][17]=[15][22]. Dan kunnen we de cijfersommen van de eerste 2 optellen om de cijfersom van de derde te krijgen.
2. De **getalwaarde** van de verdeling $[c_2][c_1][c_0]$ is dan $100 \cdot c_2 + 10 \cdot c_1 + x_0$.
3. Met de **cijfers** (die dus groter mogen zijn dan 9) van een verdeling bedoelen we de getallen tussen de haken.
4. De **cijfersom** van een verdeling is dan de som van die cijfers.
5. Een **reeks** verdelingen is een opeenvolging van verdelingen van hetzelfde getal die we onder elkaar noteren, en die ontstaat door steeds één eenheid te wisselen in 10 kleinere. De cijfersommen van de verschillende verdelingen (van hetzelfde getal) zijn dan steeds 9 groter dan hun voorganger.
6. **Standaard verlenging** van een reeks gaat door te beginnen met wisselen van één tiental, dan een honderdtal etc. ot de grootste eenheid, en dit te herhalen.

Het algoritme

Stap 1: Keuze van de tweemacht.

Laat $e = E$ de waarde van e zijn waarvoor je een oplossing zoekt met *i*=2. Dan mogen we aannemen dat E geen drievoud is, maar ook geen tweemacht (want dan hebben we dit algoritme niet nodig) en ook niet $E = 5, 10, 17$ of 25 (want dan kunnen we het bewijs dat *i*=2 niet geven)
Kies uit het lijstje op blz 7 de serie met waarden $\equiv E$ mod 9, en kies uit die serie de kleinste tweemacht $2^m > E$. Dan is dus $2^{m-6} < E < 2^m$ (want $2^{k+6} \equiv 2^k$ mod 9).

Voorbeeld: als $E = 109 \equiv 1$ mod 9. In het lijstje zie je dat $4096 = 2^{12} \equiv 1$ mod 9 de kleinste tweemacht is die voldoet, dus $2^m = 2^{12}$.

Laat 2^m de gekozen tweemacht zijn. Dan geldt dus $2^{m-6} < E < 2^m$ en $2^m \equiv E$ mod 9.
Merk op dat voor 2^m met $m < 4$ nooit gekozen kan worden: voor 1, 2, 4 en 8 zijn er geen waarden voor $E < 2^m$ met $2^m \equiv E$ mod 9.

Stap 2:

Hieronder vind je het startpunt voor de reeks die hoort bij de tweemacht 2^m die je in Stap 1 hebt gevonden.
Dat startpunt is een verdeling van 2^m als volgt (voorbeeld $2^m = 512$):

2^9 =512≡ 8 mod 9**:**
Deze kiezen we als E =26, 35 ⋯ ⋯503.
Reeks van $512 = 2^9$ met $512 \equiv 8$ mod 9:

$[c_2][c_1][c_0] = [5][1][2]$	cijfersom $= 8 = e$
$[c_2][c_1][c_0] = [4][11][2]$	cijfersom $= 17 = e$
$[c_2][c_1][c_0] = [4][10][12]$	cijfersom $= 26 = e$ ($c_2 < c_1 < c_0$)

De reeks van 512 bestaat uit verdelingen van de tweemacht in eenheden, tientallen en honderdtallen, in dit geval beginnend met de verdeling die we gewend zijn: $5 \times 100 + 1 \times 10 + 2$
De 2e en derde ontstaan door het wisselen van een honderdtal in tientallen, en van een tiental in

eenheden. [2]) De cijfersom wordt daardoor steeds 9 groter.
Voor alle cijfersommen geldt dus: $e \equiv 2^m$ mod 9 en dus ook $e \equiv E$ mod 9, en dus mogen we hopen in de reeks $e = E$ tegen te komen.

We kunnen de cijfersommen van een reeks laten lopen van de "gewone" cijfersom van de tweemacht tot de getalwaarde ervan. Over de cijfersom van een tweemacht valt moeilijk een algemene uitspraak te doen, dus moeten we bij elke tweemacht controleren of ook de kleinste waarde waarvoor we deze tweemacht kiezen in de reeks voorkomt. We vermelden daarom steeds boven de reeks welke waarden dat zijn.

We weten wel wat de grootste cijfersom is: dat is de getalwaarde van de tweemacht. En omdat we een tweemacht hebben gekozen groter dan de waarde van e kunnen we er zeker van zijn dat als de kleinste waarde er is ze er allemaal zijn.

We zullen de verdelingen rood kleuren als ze niet worden gebruikt en dus irrelevant zijn voor het algoritme. Dat kan zijn omdat e een tweemacht is (daarvoor heb je het algoritme niet nodig) of als $e \leq 2^{m-6}$ (we kozen Stap 1 voor $E > 2^{m-6}$).
We kleuren ze blauw als de cijfersom gelijk is aan één van de uitzonderingen 5, 10, 17 of 25. Voor die waarden geeft dit algoritme geen oplossing. De eerste zwarte verdeling voldoet aan een ongelijkheid van de waarden van c, die erachter vermeld wordt en e is steeds de kleinste waarde van E waarvoor we deze tweemacht kiezen.

Ga naar de tweemacht 2^m hieronder en noteer de laatste (zwarte) verdeling en cijfersom e.
Controleer en noteer ook de ongelijkheid.
Als E=e ga dan verder met Stap 4.
Als E<e ga dan verder met Stap 3.

A. **De gekozen tweemacht heeft 2 cijfers:**

- $2^4 = 16 \equiv 7$ mod 9
  Deze tweemacht kiezen we als $E = 7$.
  Reeks van $16 = 2^4$ met $16 \equiv 7$ mod 9:
  [c_1][c_0]=[1][6] cijfersom $= 7 = e$ ($c_1 < c_0$)

- $2^5 = 32 \equiv 5$ mod 9
  Deze kiezen we als E =14 of 23
  Reeks van $32 = 2^5$ met $32 \equiv 5$ mod 9:
  [c_1][c_0]=[3][2] cijfersom $= 5 = e$
  [c_1][c_0]=[2][12] cijfersom $= 14 = e$ ($c_1 < c_0$)

- 2^6 =64$\equiv$ 1 mod 9:
  Deze kiezen we als E =19, 28, 37, 46, of 55
  Reeks van $64 = 2^6$ met $64 \equiv 1$ mod 9:
  [c_1][c_0]=[6][4] cijfersom $= 10 = e$
  [c_1][c_0]=[5][14] cijfersom $= 19 = e$ ($c_1 < c_0$)

[2]) Je vraagt je misschien af hoe je bepaalt welke eenheid gewisseld wordt, want dit is géén standaard verlenging. Dat klopt. Voordat je met standaard verlenging begint moet eerst de rij cijfers in de verdeling dalend zijn. Het algoritme daarvoor is: Kies steeds van links naar rechts het eerste cijfer dat groter is dan zijn opvolger en splits daarvan één eenheid in 10 kleinere. In stap 2 hebben we dat al voor je gedaan.

B. **De gekozen tweemacht heeft 3 cijfers:**

- 2^7 =128≡ 2 mod 9:
  Deze kiezen we als e = 11 ,20,⋯ ⋯119.
  Reeks van 128 = 2^7 met 128 ≡ 2 mod 9:
  $[c_2][c_1][c_0]$=[1][2][8] cijfersom = 11 = e ($c_2 < c_1 < c_0$)

- 2^8 =256≡ 4 mod 9:
  Deze kiezen we als e = 13 ,22,⋯ ⋯247.
  Reeks van 256 = 2^8 met 256 ≡ 4 mod 9:
  $[c_2][c_1][c_0]$=[2][5][6] cijfersom = 13 = e ($c_2 < c_1 < c_0$)

- 2^9 =512≡ 8 mod 9**:**
  Deze kiezen we als e =26, 35 ⋯ ⋯503.
  Reeks van 512 = 2^9 met 512 ≡ 8 mod 9:
  $[c_2][c_1][c_0]$ = [5][1][2] cijfersom = 8 = e
  $[c_2][c_1][c_0]$ = [4][11][2] cijfersom = 17 = e
  $[c_2][c_1][c_0]$ = [4][10][12] cijfersom = 26 = e ($c_2 < c_1 < c_0$)

C. **De gekozen tweemacht heeft 4 cijfers en is ≤ 4096 (dus 1024, 2048 of 4096).**

- 2^{10} =1024≡ 7 mod 9**:**
  Deze kiezen we als e =34, 41⋯ ⋯1015.
  Reeks van 1024 = 2^9 met 1024 ≡ 7 mod 9:
  $[c_3][c_2][c_1][c_0]$ = [1][0][2][4] cijfersom = 7 = e
  $[c_3][c_2][c_1][c_0]$ = [0][10][2][4] cijfersom = 16 = e
  $[c_3][c_2][c_1][c_0]$ = [0][9][12][4] cijfersom = 25 = e
  $[c_3][c_2][c_1][c_0]$ = [0][9][11][14] cijfersom = 34 = e ($c_3 = 0$ en $c_2 < c_1 < c_0$)

- 2^{11} =2048≡ 5 mod 9**:**
  Deze kiezen we als e = 41 ,50,⋯ ⋯2039.
  Reeks van 2048 = 2^9 met 2048 ≡ 5 mod 9:
  $[c_3][c_2][c_1][c_0]$ = [2][0][4][8] cijfersom = 14 = e
  $[c_3][c_2][c_1][c_0]$ = [1][10][4][8] cijfersom = 23 = e
  $[c_3][c_2][c_1][c_0]$ = [1][9][14][8] cijfersom = 32 = e
  $[c_3][c_2][c_1][c_0]$ = [1][9][13][18] cijfersom = 41 = e
  ($c_3 < c_0 - c_1$ en.. $c_3 < c_1 - c_2$ en $c_3 < c_2$) [3])

- 2^{12} =4096≡ 1 mod 9**:**
  Deze kiezen we als e = 73, 82,⋯ ⋯4087.
  Reeks van 4096 = 2^{12} met 4096 ≡ 1 mod 9:
  $[c_3][c_2][c_1][c_0]$ = [4][0][9][6] cijfersom = 19 = e
  $[c_3][c_2][c_1][c_0]$ = [3][10][9][6] cijfersom = 28 = e
  $[c_3][c_2][c_1][c_0]$ = [3][9][19][6] cijfersom = 37 = e
  $[c_3][c_2][c_1][c_0]$ = [3][9][18][16] cijfersom = 46 = e
  $[c_3][c_2][c_1][c_0]$ = [3][9][17][26] cijfersom = 55 = e
  $[c_3][c_2][c_1][c_0]$ = [3][9][16][36] cijfersom = 64 = e
  $[c_3][c_2][c_1][c_0]$ = [3][8][26][36] cijfersom = 73 = e

[3]) Je verwacht misschien: $c_3 < c_2 < c_1 < c_0$. Dat is ook uit de 3 ongelijkheden af te leiden: als $c_3 < c_0 - c_1$ dan moet (als de cijfers niet < 0 zijn) $c_1 < c_0$ en als $c_3 < c_1 - c_2$ moet $c_2 < c_1$. De ongelijkheid die we gebruiken is dus strenger: de afstanden tussen c_0, c_1 en c_2 moeten groter zijn dan c_3

($c_3 < c_0 - c_1$ en.. $c_3 < c_1 - c_2$ en $c_3 < c_2$)

D. **De gekozen tweemacht is groter dan 4096**

In Stap 2 punt D bij $2^{12} = 4096$ vonden we de cijfersom 64 met verdeling:
$[c_3][c_2][c_1][c_0] = [3][9][16][36]$ cijfersom $= 64 = e$
Die is overbodig als we op zoek zijn naar een oplossing voor $e = 64$, maar voor de tweemachten 2^m met $m > 12$ kunnen we met deze cijfersom een klein wonder verrichten.
Deze tweemachten kunnen we schrijven als $4096 \cdot 2^k = 2^{k+12}$ met $k = m - 12$. De voorganger met dezelfde waarde mod 9 is een factor 2^6 kleiner dus die wordt geacht de waarden van e te leveren tot 2^{k+6}.
De waarden van e die we met 2^n hopen te maken beginnen dus bij $2^{k+6} + 9$ en eindigen bij $2^{k+12} - 9$.
Omdat [3][9][16][36] een verdeling is van 4096 is $[3 \cdot 2^k][9 \cdot 2^k][16 \cdot 2^k][36 \cdot 2^k]$ een verdeling van $4096 \cdot 2^k$.
De cijfersom is $3 \cdot 2^k + 9 \cdot 2^k + 16 \cdot 2^k + 36 \cdot 2^k = 2^k \cdot (3 + 9 + 16 + 36) = 2^k \cdot 64 = 2^{k+6}$
Die is overbodig, maar na één wisseling hebben we:
$[c_3][c_2][c_1][c_0] = [3 \cdot 2^k][9 \cdot 2^k][16 \cdot 2^k - 1][36 \cdot 2^k + 10]$ cijfersom $= 2^{k+6} + 9 = e$.
Aan het einde van de reeks is de cijfersom gelijk aan de waarde van de tweemacht, dus $4096 \cdot 2^k = 2^{k+12}$, en dus kunnen we precies de waarden van e maken die we nodig hebben.

Bereken $k = m - 12$ uit je gekozen tweemacht 2^m en bereken daarmee de waarden van c_3, c_2, c_1 en c_0 en de cijfersom.
Vul dat in in de 2e cijfersom (met cijfersom $2^{k+6} + 9 = e$) in de reeks van 2^k hieronder.

$2^k \equiv E$ mod 9:
Deze kiezen we als $e = E > 4096$.
Reeks van 2^k met $2^k \equiv E$ mod 9:
$[c_3][c_2][c_1][c_0] = [3 \cdot 2^k][9 \cdot 2^k][16 \cdot 2^k][36 \cdot 2^k]$ met cijfersom= $2^{k+6} = e$
$[c_3][c_2][c_1][c_0] = [3 \cdot 2^k][9 \cdot 2^k][16 \cdot 2^k - 1][36 \cdot 2^k + 10]$ met cijfersom= $2^{k+6} + 9 = e$
($c_3 < c_0 - c_1$ en.. $c_3 < c_1 - c_2$ en $c_3 < c_2$)

Stap 3:

De ongelijkheid die je in stap 2 gecontroleerd en genoteerd hebt noemen we de **strenge eis** (er staan alleen < tekens in).
Je gaat nu de reeks van 2^m waarvan je in stap 2 het begin hebt opgezocht verder voortzetten met **standaard verlenging** (zie definitie 4)tot je de cijfersom vindt die gelijk is aan E.
Je krijgt dan, met hetzelfde voorbeeld als in Stap 2:

2^9 =512≡ 8 mod 9:
Deze kiezen we als E =26, 35 ⋯ ⋯503.
Reeks van $512 = 2^9$ met $512 \equiv 8$ mod 9:
$[c_2][c_1][c_0] = [5][1][2]$ cijfersom $= 8 = e$
$[c_2][c_1][c_0] = [4][11][2]$ cijfersom $= 17 = e$
$[c_2][c_1][c_0] = [4][10][12]$ cijfersom $= 26 = e$
Vanaf hier begint de standaard verlenging:
$[c_2][c_1][c_0] = [4][9][22]$ cijfersom $= 35 = e$
$[c_2][c_1][c_0] = [3][19][22]$ cijfersom $= 44 = e$
⋮
Verdeling met cijfersom = E

Bij elke stap wordt de waarde van e groter met stappen van 9, en, zoals we al zagen in stap 2, kan

je dat voortzetten tot $e = E$.
Dan kan het gebeuren dat er verdelingen verschijnen die niet voldoen aan de strenge eis, maar wel aan de wat minder strenge eisen (met ook ≤ tekens) die je hieronder vindt:

Bij 4 cijfers $[c_3][\,c_2][\,c_1][\,c_0]$ geldt: $c_3 < c_0 - c_1$ en.. $c_3 \leq c_1 - c_2$ en $c_3 \leq c_2$
Bij 3 cijfers $[\,c_2][\,c_1][\,c_0]$: $c_2 < c_1 \leq c_0$ of $c_2 \leq c_1 < c_0$
Bij 2 cijfers $[\,c_1][\,c_0]$: $c_1 \leq c_0$
Dat bij standaard verlenging geldt dat als de eerste verdeling aan de strenge eisen voldoet alle volgende verdelingen ten minste aan de minder strenge eisen voldoen wordt bewezen in **stelling 1 op blz. 14**

Als je $e = E$ hebt gevonden noteer je weer de laatste verdeling met cijfersom en de ongelijkheid die hoort bij aantal cijfers hierboven (dat kan zijn verminderd tijdens de wisselingen). Daarmee ga je naar Stap 4.

Stap 4:
Je hebt nu een verdeling $[c_3][\,c_2][\,c_1][\,c_0]$ van de tweemacht 2^m met maximaal 4 en minimaal 2 cijfers en met cijfersom $e = E$. Ook hebben we een rij nullen en enen met in totaal E enen. Voor de cijfers geldt, afhankelijk van het aantal, een ongelijkheid (zie het eind van stap 3).

In die rij nullen en enen gaan we, van links naar rechts, groepjes enen en nullen omcirkelen en vormen zo getallen die we gaan optellen.

Het doel is om daarmee precies c_3 duizendtallen, c_2 honderdtallen, $c_1$tientallen en c_0 eenheden te maken. Als dat lukt is de uitkomst van de optelling die $a(1)$ bepaalt gelijk aan de getalwaarde van $[c_3][\,c_2][\,c_1][\,c_0]$, dus 2^m.

Om bij te houden hoe ver we zijn trekken we steeds de getallen die we maken af van $[c_3][\,c_2][\,c_1][\,c_0]$ (in het voorbeeld hieronder is dat [3][8][26][36]). Dat levert een reeks nieuwe verdelingen op waarvan de waarde en de cijfersom bij elke omcirkeld groepje daalt.
Je ziet aan de dalende cijfersommen hoeveel enen er nog over zijn in de reeks en in de verdelingen hoeveel enen, tientallen etc. er nog moeten worden omcirkeld.
De laatste verdeling geeft daarbij steeds aan hoeveel we nog van elke eenheid moeten maken (in het voorbeeld dus nog [7][24][34]). Als het lukt om [0][0][0][0] te maken zijn we klaar. Dan is de cijfersom, en dus het aantal ongebruikte enen in de reeks 0. De som van de omcirkelde en afgetrokken getallen is dan gelijk 2^m, want dat is de getalwaarde van de verdeling met cijfersom E (in het voorbeeld [3][8][26][36]).

$[c_3][\,c_2][\,c_1][\,c_0]$ = [3][8][26][36]	cijfersom = 73 = E	
$[c_3][\,c_2][\,c_1][\,c_0]$ = [2][8][25][35]	cijfersom = 70	omcirkeld [1][0][1][1]
$[c_3][\,c_2][\,c_1][\,c_0]$ = [1][7][24][35]	cijfersom = 67	omcirkeld [1][1][1][0]
$[c_3][\,c_2][\,c_1][\,c_0]$ = [0][7][24][34]	cijfersom = 65	omcirkeld [1][0][0][1]
⋮		

Het algoritme om de groepjes te omcirkelen is simpel: Je werkt van links naar rechts.
Je omcirkelt steeds de eerstvolgende 1, eventueel samen met 1 of meer cijfers die daar achter staan. Daardoor gebruik je in elk geval alle enen die je tegenkomt.

Je omcirkelt (bijna) steeds net zoveel cijfers als het aantal cijfers van je verdeling.
Daarop is één uitzondering: Als je verdeling 3 cijfers heeft, **en** daarvoor geldt $c_1 = c_0$, **en** het eerstvolgende groepje in de reeks enen en nullen begint met 10 , dan omcirkel je 2 cijfers.

Dat dit algoritme, altijd eindigt met [0][0][0][0] met cijfersom 0, (en niet bijvoorbeeld met [0][0][-1][1] met cijfersom 0) bewijzen we in stelling 2 op blz 16 .

Stellingen

(Zie voor stelling 2 blz. 16.)

Stelling 1:
Als we in een reeks vanaf een verdeling met maximaal 4 cijfers aanvullen met standaard verlenging, en de cijfers van de eerste verdeling voldoen aan:

- Bij 4 cijfers $[c_3][c_2][c_1][c_0]$: $c_3 < c_0 - c_1$ en.. $c_3 < c_1 - c_2$ en $c_3 < c_2$
- Bij 3 cijfers $[c_2][c_1][c_0]$: $c_2 < c_1 < c_0$
- Bij 2 cijfers $[c_1][c_0]$: $c_1 < c_2$

dan geldt voor alle verdelingen die ontstaan door standaard verlenging

:

- Bij 4 cijfers $[c_3][c_2][c_1][c_0]$: $c_3 < c_0 - c_1$ en $c_3 \leq c_1 - c_2$ en $c_3 \leq c_2$
- Bij 3 cijfers $[c_2][c_1][c_0]$: $c_2 < c_1 \leq c_0$ of $c_2 \leq c_1 < c_0$
- Bij 2 cijfers $[c_1][c_0]$: $c_1 < c_0$

De eisen voor alle verdelingen zijn dus iets minder streng dan de eisen waarmee we beginnen.

We kijken voor elk aantal cijfers p wat er gebeurt als we standaard verlenging toepassen, dus steeds eerst een tiental wisselen, dan een honderdtal en zo door tot het meest significante cijfer c_{p-1} en dat herhalen tot $c_{p-1} = 0$.

We zullen bewijzen dat als zo'n ronde begint met de strengere eisen alle verdelingen in de ronde voldoen aan de minder strenge eisen, terwijl de laatste weer voldoet aan de strengere eisen.
Omdat die laatste verdeling weer het beginpunt is voor de volgende ronde kunnen we dan dus steeds doorgaan en voldoen alle volgende verdelingen aan de minder strenge eisen.

Als $c_{p-1} = 0$ zijn we altijd aan het einde van een ronde (c_{p-1}wordt alleen kleiner bij de laatste stap van een ronde). De volgende ronde is dus met $p - 1$ cijfers, en dus moeten de strengere eisen voor $p - 1$ gelden.

We bewijzen daarom voor elk aantal cijfers ≤ 4:

1. Als we een ronde starten met een verdeling met p cijfers waarvoor strengere eisen voor p cijfers gelden, en we voeren één ronde standaardverlenging uit, dan gelden voor alle regels de minder strenge eisen en voor de laatste van de ronde weer de strengere eisen.

2. Als $c_{p-1} = 0$ dan gaan de strengere eisen voor p cijfers over in de strengere eisen voor $p - 1$ cijfers

Bewijs van Stelling 1.1:

Als we een ronde starten met een verdeling met p cijfers waarvoor strengere eisen voor p cijfers gelden, en we voeren één ronde standaardverlenging uit, dan gelden voor alle regels de minder strenge eisen en voor de laatste van de ronde weer de strengere eisen.

a. Met 2 cijfers: De strengere en de minder strengere eis zijn dan beide $c_1 < c_0$. De ronde bestaat dus uit één stap. In die stap wisselen we een tiental voor 10 eenheden, dus c_1 wordt kleiner en c_0 wordt groter. Dus geldt opnieuw $c_1 < c_0$.

b. Met 3 cijfers: De strengere eis is $c_2 < c_1 < c_0$ en de minder strenge is $c_2 < c_1 \leq c_0$ of $c_2 \leq c_1 < c_0$.
De ronde bestaat uit twee stappen: we wisselen eerst een tiental voor 10 eenheden, en dan een honderdtal voor 10 tientallen.
Bij 3 en 4 cijfers noteren we de cijfers in de eerste verdeling als v_i, en berekenen we daarmee de volgende verdelingen.
We krijgen dan:
$[c_2][c_1][c_0] = [v_2] \quad [v_1] \quad [v_0]$ begin met $v_2 < v_1 < v_0$.
$[c_2][c_1][c_0] = [v_2] \quad [v_1 - 1][v_0 + 10]$ daaruit volgt: $v_2 \leq v_1 - 1 < v_0 + 10$
$[c_2][c_1][c_0] = [v_2 - 1][v_1 + 9][v_0 + 10]$ en ook: $v_2 - 1 < v_1 + 9 < v_0 + 10$
Na de eerste stap geldt dus $c_2 \leq c_1 < c_0$, en dat voldoet zeker aan $c_2 < c_1 \leq c_0$ of $c_2 \leq c_1 < c_0$
En na de laatste stap geldt weer $c_2 < c_1 < c_0$

c. Met 4 cijfers: De strengere eis is dan: $c_3 < c_0 - c_1$ en.. $c_3 < c_1 - c_2$ en $c_3 < c_2$ en de minder strenge: $c_3 < c_0 - c_1$ en $c_3 \leq c_1 - c_2$ en $c_3 \leq c_2$.

De ronde bestaat dan uit 3 stappen, en we krijgen dan:
$[c_3][c_2][c_1][c_0] = [v_3] \quad [v_2] \quad [v_1] \quad [v_0]$
$[c_3][c_2][c_1][c_0] = [v_3] \quad [v_2] \quad [v_1 - 1][v_0 + 10]$
$[c_3][c_2][c_1][c_0] = [v_3] \quad [v_2 - 1][v_1 + 9][v_0 + 10]$
$[c_3][c_2][c_1][c_0] = [v_3 - 1][v_2 + 9][v_1 + 9][v_0 + 10]$

We starten vanaf de eerste regel, waarvoor geldt:
$v_3 < v_0 - v_1$ en $v_3 < v_1 - v_2$ en $v_3 < v_2$
Na één stap hebben we in de tweede regel:
$v_3 < v_0 - v_1 \Leftrightarrow a_3 < (v_0 + 10) - (v_1 - 1) - 11 \Leftrightarrow c_3 < c_0 - c_1 - 11 \Rightarrow \boldsymbol{c_3 < c_0 - c_1}$
$v_3 < v_1 - v_2 \Leftrightarrow a_3 < (v_1 - 1) - v_2 + 1 \Leftrightarrow c_3 < c_1 - c_2 + 1 \Rightarrow \boldsymbol{c_3 \leq c_1 - c_2}$
$v_3 < v_2 \Leftrightarrow \boldsymbol{c_3 < c_2}$

In de derde regel geldt:
$v_3 < v_0 - v_1 \Leftrightarrow a_3 < (v_0 + 10) - (v_1 + 9) - 1 \Leftrightarrow c_3 < v_0 - c_1 - 1 \Rightarrow \boldsymbol{c_3 < c_0 - c_1}$
$v_3 < v_1 - v_2 \Leftrightarrow a_3 < (v_1 + 9) - (v_2 - 1) - 10 \Leftrightarrow c_3 < c_1 - c_2 - 10 \Rightarrow \boldsymbol{c_3 < c_1 - c_2}$
$v_3 < v_2 \Leftrightarrow v_3 < (v_2 - 1) + 1 \Leftrightarrow c_3 < c_2 + 1 \Rightarrow \boldsymbol{c_3 \leq c_2}$

En in de vierde regel:
$v_3 < v_0 - v_1 \Leftrightarrow (v_3 - 1) < (v_0 + 10) - (v_1 + 9) - 2 \Leftrightarrow c_3 < c_0 - c_1 - 2 \Rightarrow \boldsymbol{c_3 < c_0 - c_1}$
$v_3 < v_1 - v_2 \Leftrightarrow (v_3 - 1) < (v_1 + 9) - (v_2 + 9) - 1 \Leftrightarrow c_3 < c_1 - c_2 - 1 \Rightarrow \boldsymbol{c_3 < c_1 - c_2}$
$v_3 < v_2 \Leftrightarrow (v_3 - 1) < (v_2 + 9) - 10 \Leftrightarrow c_3 < c_2 - 10 \Rightarrow \boldsymbol{c_3 < c_2}$

Dus geldt voor alle regels de minder strenge eisen: $c_3 < c_0 - c_1$, $c_3 \leq c_1 - c_2$, $c_3 \leq c_2$
En voor de laatste: $c_3 < c_0 - c_1$, $c_3 < c_1 - c_2$, $c_3 < c_2$, dus gelden daar weer de strengere eisen. QED

Bewijs van Stelling 1.2:

Als $c_{p-1} = 0$ dan gaan de strengere eisen voor p cijfers over in de strengere regels voor $p - 1$ cijfers

a. Met 2 cijfers is het triviaal, want er zijn geen eisen voor één cijfer.

b. Met 3 cijfers is de strengere eis: $c_2 < c_1 < c_0$. Als $c_2 = 0$ wordt dat $c_1 < c_0$ en dat is de strengere eis voor 2 cijfers.

c. Met 4 cijfers is de strengere eis: $c_3 < c_0 - c_1$ en.. $c_3 < c_1 - c_2$ en $c_3 < c_2$
Als : $c_3 = 0$ wordt dat: $c_1 < c_0$ en $c_2 < c_1$ en $c_2 > 0$ en dus $0 < c_2 < c_1 < c_0$

QED.

Stelling 2: We moeten bewijzen dat als een verdeling $[c_3][c_2][c_1][c_0]$ met getalwaarde 2^m en cijfersom E voldoet aan bepaalde voorwaarden, en we trekken er volgens het algoritme in stap 4 groepjes enen en nullen met in totaal E enen van af, we eindigen met [0][0][0][0] met cijfersom 0. Na elke aftrekking ontstaat een nieuwe verdeling met een kleinere cijfersom. We gaan aantonen dat ook die nieuwe verdelingen voldoen aan dezelfde voorwaarden.[4])

Het is triviaal dat de cijfersom na het aftrekken 0 is, omdat we beginnen met cijfersom E en in totaal E enen aftrekken. Als we aantonen dat bij het aftrekken er nooit een negatief cijfer ontstaat zijn we dus klaar.

De voorwaarden zijn:

- Bij 4 cijfers: $[c_3][c_2][c_1][c_0]$: $c_3 < c_0 - c_1$ en $c_3 \leq c_1 - c_2$ en $c_3 \leq c_2$.
- Bij 3 cijfers: $[c_2][c_1][c_0]$: $c_2 \leq c_1 < c_0$ of $c_2 < c_1 \leq c_0$
- Bij 2 cijfers: $[c_1][c_0]$: $c_1 \leq c_0$

En het algoritme zorgt ervoor dat:
Het aantal cijfers van het getal dat je aftrekt altijd gelijk is aan het aantal cijfers >0 van de verdeling
Daarop is één uitzondering: Als de verdeling 3 cijfers heeft, **en** daarvoor geldt $c_1 = c_0$, **en** het eerstvolgende groepje in de reeks enen en nullen begint met 10 , dan omcirkel je 2 cijfers.

We zullen voor elk aantal cijfers p aantonen dat:

1. Als we p cijfers hebben waarvoor de voorwaarden voor p cijfers gelden, en we volgen het algoritme tot er nog $p-1$ cijfers over zijn dan blijven die voorwaarden gelden.
2. Als we in de voorwaarden voor p cijfers voor het meest significante cijfer invullen $c_{p-1} = 0$ dan gaan die voorwaarden over in de voorwaarden voor $p-1$ cijfers.

Alle voorwaarden garanderen dat de cijfers in de verdeling een niet-dalende rij vormen. De getallen die we aftrekken bestaan alleen uit enen en nullen, en hebben nooit meer cijfers dan de verdeling. Dus kunnen er bij zo'n aftrekking nooit negatieve cijfers ontstaan.

We kunnen dus volstaan met een bewijs van de bovenstaande punten 1. en 2.

Bewijs van Stelling 2.1:

Als we p cijfers hebben waarvoor de voorwaarden voor p cijfers gelden, en we volgen het algoritme tot er nog $p-1$ cijfers over zijn dan blijven die voorwaarden gelden.

a. Als $p = 2$ is de voorwaarde: $c_1 \leq c_0$. Dan trekken we ofwel [1][1] ofwel [1][0] af. In beide gevallen blijft de voorwaarde gelden.

b. Als $p = 3$ is de voorwaarde: $c_2 \leq c_1 < c_0$ of $c_2 < c_1 \leq c_0$. We trekken af: ofwel [1][1][1], [1][1][0], [1][0][0] , [1][0][1] of [1][0].
In de eerste 3 gevallen blijft de voorwaarde gelden.
De vierde ([1][0][1]) trekken we alleen af als $c_2 \leq c_1 < c_0$ en daarna geldt dus: $c_2 < c_1 \leq c_0$
De vijfde ([1][0]) trekken we alleen af als $c_2 < c_1 \leq c_0$ en daarna geldt dus: $c_2 \leq c_1 < c_0$
In alle gevallen blijft de voorwaarde dus gelden.

c. Als $p = 4$ zijn de voorwaarden: $c_3 < c_0 - c_1$ en $c_3 \leq c_1 - c_2$ en $c_3 \leq c_2$
We trekken een verdeling af waarvan het eerste cijfer 1 is, gevolgd door 3 cijfers die 0 of 1 kunnen zijn, dus: $[1][d_2][d_1][d_0]$ met d_2, d_1, en $d_0 = 0$ of 1.

---

[4]) Misschien denk je: moet dat bewezen worden? Alle ongelijkheden garanderen dat de cijfers in de verdelingen een niet-dalende reeks vormen. Je ziet zo dat het dan zeker lukt als er geen nullen tussen de enen staan. Dan lukt het toch altijd als er wel nullen tussen staan? (je zou de enige niet zijn). Tegenvoorbeeld: voor e=6 kan je met 6 enen zonder nullen ertussen 111+111= 222 maken. Met 1111011 lukt dat niet.

Dan worden de 3 rechterleden van de ongelijkheden maximaal 1 kleiner en de linkerleden precies 1 kleiner. Dus blijven de voorwaarden geldig. [5])
Dus voor $p \leq 4$ geldt: Als we p cijfers hebben waarvoor de voorwaarden voor p cijfers gelden, en we volgen het algoritme tot er nog $p-1$ cijfers over zijn dan blijven die voorwaarden gelden. QED.

Bewijs van Stelling 2.2:

Als we in de voorwaarden voor p cijfers voor het meest significante cijfer invullen $c_{p-1} = 0$ dan gaan die voorwaarden over in de voorwaarden voor $p-1$ cijfers.

a. Als $p = 2$ is het triviaal want voor $p = 1$ zijn er geen voorwaarden.
b. Als $p = 3$ is de voorwaarde: $c_2 \leq c_1 < c_0$. Als $c_2 = 0$ wordt dat $c_1 < c_0$ en dus zeker $c_1 \leq c_0$.
c. Als $p = 4$ zijn de voorwaarden: $c_3 < c_0 - c_1$ en $c_3 \leq c_1 - c_2$ en $c_3 \leq c_2$. Als $c_3 = 0$ wordt dat:
$0 < c_0 - c_1$ en $0 \leq c_1 - c_2$ en $0 \leq c_2$, en dus $0 \leq c_2 \leq c_1 < c_0$.
QED.

---

[5]) Hier maken we gebruik van de strengere ongelijkheden bij 4 cijfers. We eisen dat de afstanden tussen de laatste 3 cijfers groter moeten zijn dan c_3. Dan kunnen, ook als er nullen staan tussen de enen, die afstanden niet negatief worden tijdens het maken van de duizendtallen.
Het algoritme voor het maken van de optelling met 4 cijfers heeft daardoor geen uitzondering zoals die voor 3 cijfers.