

Het succes van de compact disc (een Nederlandse uitvinding!) is te danken aan allerlei technische innovaties, maar ook aan een flinke portie wiskunde. Op de openingslezing van de Nationale Wiskunde Dagen 1998 ging **Jack van Lint** in op de wiskundige principes die hierbij een rol spelen.

## Wiskunde en de compact disc

### 0-1 Rijen met $d - k$ beperking

Als aanloop beschouwen we een bekend probleem uit de combinatoriek. We hebben een rij van  $n$  kooien. In iedere kooi kan wel of niet een leeuw zitten. Als twee leeuwen in aangrenzende kooien zitten, irriteren ze elkaar en gaan brullen. Dat willen we voorkomen. Een configuratie van kooien waarbij het niet gebeurt dat twee leeuwen in naast elkaar gelegen kooien zitten, noemen we 'toegestaan'. We representeren zo'n configuratie door een rij 0'en en 1'en (van lengte  $n$ ). Daarbij geeft een 1 een kooi met leeuw aan en een 0 een lege kooi. Laat  $a(n)$  het aantal toegestane rijen van lengte  $n$  zijn (en  $a(0) = 1$ ).

Voorbeeld: Voor  $n = 3$  zijn 000, 100, 010, 001 en 101 toegestaan. Dus is  $a(3) = 5$ .

#### Stelling

Voor  $n \geq 2$  geldt:  $a(n) = a(n-1) + a(n-2)$ .

#### Bewijs

Een toegestane rij die eindigt op een 1 heeft 01 als laatste twee symbolen en daarvoor een toegestane rij van lengte  $n-2$ . Er zijn  $a(n-2)$  zulke rijen. Is het laatste symbool een 0, dan komt daarvoor een willekeurige toegestane rij van lengte  $n-1$  en daarvan zijn er  $a(n-1)$ .

We zien dus dat de rij  $a(n)$  de bekende rij van Fibonacci is, namelijk 1, 2, 3, 5, 8, 13, ...

#### Definitie

Een 0-1 rij met  $d - k$  beperking is een rij met de eigenschap dat tussen twee opvolgende 1'en ten minste  $d$  0'en staan en er nergens meer dan  $k$  nullen achterelkaar staan. Ons eerste voorbeeld had  $d = 1$  en  $k = \infty$ .

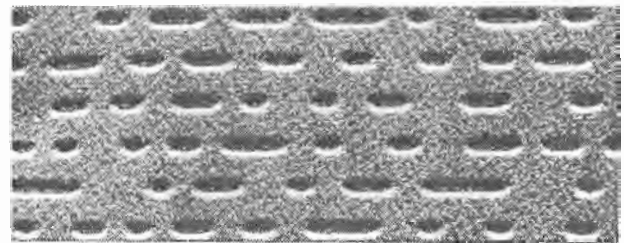
#### Opgave

Laat  $a(n)$  het aantal 0-1 rijen van lengte  $n$  zijn met  $d - k$  beperking waarin  $d = 2$ ,  $k = 3$ . Toon aan dat voor  $n \geq 6$  geldt:

$$a(n) = a(n-3) + a(n-4).$$

Aanwijzing: druk alles uit in  $b(n)$ : = het aantal zulke rijen beginnend met een 1.

Opmerking: Als we hier door kleine gevallen te analyseren een vermoeden proberen te krijgen, kunnen we op een dwaalspoor komen! Ga na dat  $a(n) = n + 1$  voor  $n \leq 6$  (maar verder niet).



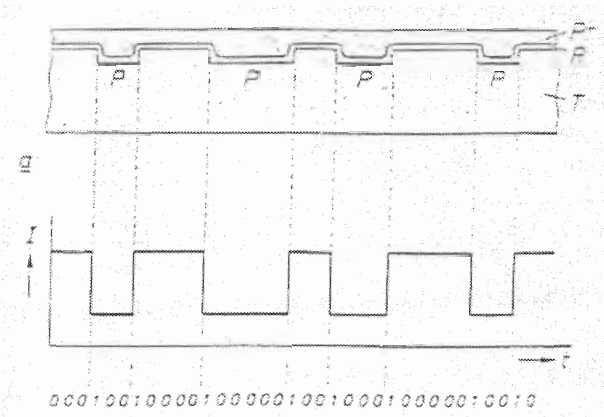
Sporen op een CD-plaat

Het spoor op een CD is een vijf kilometer lange spiraal met een spoorbreedte van  $0,6 \mu\text{m}$  (en spoed  $1,6 \mu\text{m}$ ). Het spoor bestaat uit zogenaamde *putten* (diepte  $0,12 \mu\text{m}$ ) met daartussen zogenaamde *dammen*. De lengte van een put of dam is maximaal  $3,3 \mu\text{m}$  en minimaal  $0,9 \mu\text{m}$ . Een *bit* (dat is een 0 of een 1) op de plaat heeft een lengte van  $0,3 \mu\text{m}$ . Bij het aflezen van de plaat (met een laserbundel) wordt een overgang van een put naar een dam (of omgekeerd) als een 1 gelezen en daarna 0'en tot de volgende overgang (= *flank*). Een put van lengte  $1,8 \mu\text{m}$  wordt dus geïnterpreteerd als 100000, de kortste dam als 100, enzovoort.

In de terminologie van hierboven hebben we dus te maken met een rij met  $d - k$  beperking met  $d = 2$  en  $k = 10$ .

Waarom deze beperkingen? De lichtbundel wordt door de plaat teruggekaatst. Idealiter is de diepte van een put een kwart van de golflengte van het gebruikte licht waardoor invallend en teruggekaatst licht elkaar uitdempen en de put donker lijkt. Dit is niet helemaal het geval, maar er is duidelijk verschil tussen putten en dammen. De lichtvlek ziet dus steeds flanken passeren, maar de zaak zou in de war raken als die vlek twee flanken tegelijk zag. Dit heet *intersymbol interference*. Daarom mogen twee 1'en

niet te dicht bij elkaar staan. De bit-klok, die de overgang van een bit naar de volgende bijhoudt, wordt steeds gelijkgezet als een flank wordt gepasseerd. Om te zorgen voor goede synchronisatie mag het na zo'n flank niet te lang duren voor de volgende flank langskomt. Daarvoor is  $k = 10$  gekozen.



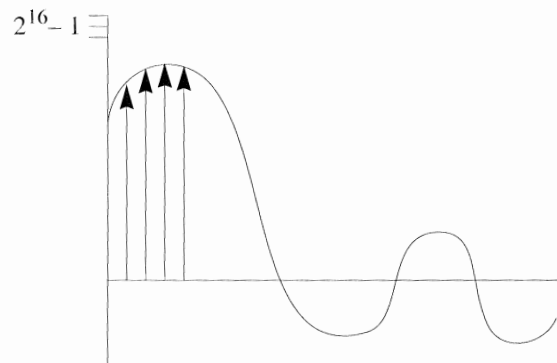
Codering van putten en dammen

We zullen zien dat bij de omzetting van muziek als analog signaal naar een digitale representatie en de daarop volgende codering uiteindelijk een rij symbolen ontstaat (die we overigens *letters* van ons *alfabet* zullen noemen) waarin elk van die symbolen (= letters) een rij van acht 0'en en 1'en is, een *byte*. Zo is 11011101 één van de letters van dat alfabet, dat dus uit  $2^8 = 256$  letters bestaat. Deze letters kunnen we *niet* op de plaat vastleggen, omdat ze niet aan de  $d - k$  beperking voldoen. In brochures over de CD vindt men vaak de kreet 'het EFM modulatie systeem' en dat staat heel duur. Het betekent 'Eight to Fourteen Modulation' en die kunnen we nu begrijpen. De 256 bytes moeten vertaald worden naar rijtjes die wel aan de beperkingen voldoen. Via een recurrente betrekking als we boven zagen (in voorbeeld en opgave), kunnen we uitrekenen dat de kleinste  $n$  waarvoor er ten minste 256 rijtjes zijn met  $2 - 10$  beperking  $n = 14$  is. Er zijn dan 267 toegestane rijtjes, waarvan we er (gelukkig) elf niet hoeven te gebruiken. Er is namelijk een *koppingsprobleem!* Een toegestaan rijtje eindigend op een 1 kan niet opgevolgd worden door een toegestaan rijtje dat met 1 of 01 begint, want dan staan tussen die twee 1'en niet ten minste twee 0'en. Als we de koppeling willen realiseren met een vast aantal bits, dan krijgen we problemen met rijtjes die op tien 0'en eindigen of beginnen. Daar moeten vier symbolen tussen om aan de eisen te voldoen. Gelukkig kunnen we de ergste boosdoeners weglaten en het blijkt dat we met drie koppingsbits de rijtjes aan elkaar kunnen plakken. Daarbij hebben we nog zo veel vrijheid, dat we ervoor kunnen zorgen dat uiteindelijk op de CD op een redelijk 'lang' stuk spoor (zeg een halve mm) de totale lengte van de putten respectievelijk dammen ongeveer gelijk is.

We onthouden: *De CD-speler 'weet' dat voortdurend ongeveer de helft van het gereflecteerde licht helder is en de helft donker.*

## Analoog-digitaal conversie

Een tweede technische term die men vaak tegenkomt bij het lezen over de CD is PCM (Pulse Code Modulation). Dit werkt als volgt. Een analog signaal (zie figuur) wordt bemonsterd met een frequentie van 44100 keer per seconde (44,1 kHz).



Deze frequentie is gekozen om aan te sluiten bij de televisie standaard (PAL) die ook voor videorecorders wordt gebruikt. Volgens de signaaltheorie heeft deze bemonsteringsfrequentie tot gevolg dat het op de plaat vastgelegde audiosignaal een bandbreedte van 20 kHz heeft (genoeg voor het menselijk oor).

De monsters worden gemeten op een schaal van 0 tot  $2^{16} - 1$  en binair geschreven. Aangezien we met stereomuziek te maken hebben (twee kanalen), betekent dit dat één bemonstering 32 bits oplevert en dat zijn vier bytes (de symbolen waarmee de wiskunde wordt bedreven). Bedenk dat bij het opnemen van muziek dus  $1,4 \cdot 10^6$  bits/sec worden gelezen. Dit zijn zogenaamde *audiobits* (nog niet geschikt om op de CD te worden geschreven).

## Foutenverbetering

Als er uiteindelijk op de CD geregistreerd wordt, komen er niet  $5 \cdot 10^9$  audiobits, maar ongeveer drie keer zoveel zogenaamde *kanaalbits* op. Door deeltjes en luchtbelletjes in de plaat, onnauwkeurigheden ontstaan bij het persen, vuil op de plaat, vingerafdrukken en krassen ontstaan afleesfouten. Bij de terugvertaling naar audiobits kunnen er wel 500.000 fout zijn (dat wil zeggen dat een 1 als 0 wordt gelezen of omgekeerd). Als hier niets aan gedaan wordt, is de muziek niet om aan te horen. Hier komt de theorie van *foutenverbeterende codes* ons te hulp.

### Voorbeeld

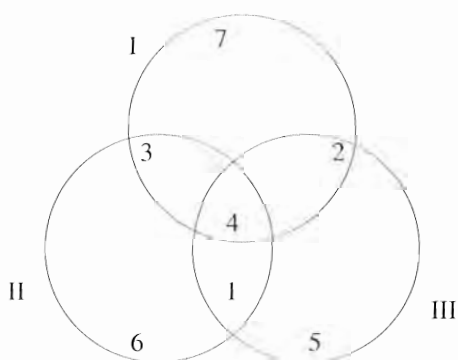
We geven eerst een heel eenvoudig voorbeeld van het toevoegen van *redundante symbolen* aan informatie om daarmee eventueel optredende fouten bij transmissie of opslag te kunnen verbeteren.

We beginnen met een alfabet van slechts twee symbolen,

namelijk 0 en 1. De informatie is een lange rij 0'en en 1'en (bits). We verdelen deze rij in blokken van vier bits en aan elk viertal voegen we drie extra symbolen (redundante bits) toe.

In onderstaand Venn-diagram zijn de zeven cirkeldelen genummerd van 1 tot en met 7. In de delen 1 tot en met 4 schrijven we de vier 'informatiebits'  $a_1, a_2, a_3, a_4$ . De drie redundante bits  $a_5, a_6, a_7$  voldoen aan de regel: *iedere cirkel heeft een even aantal 1'en*.

Het zevental dat zo wordt gevormd, noemen we een *codewoord*. Stel nu dat in zo'n zevental één van de bits wordt veranderd in de andere mogelijkheid ( $0 \rightarrow 1$  of  $1 \rightarrow 0$ ). Van ieder van de drie cirkels kunnen we vaststellen of de gemaakte fout wel of niet in die cirkel zit. Daarmee is het cirkeldeel waarin de fout zit, gelokaliseerd en dan kunnen we de fout verbeteren omdat het alfabet maar twee symbolen heeft.



Dit is een voorbeeld van wat een één fout verbeterende code heet. We zeggen dat de code lengte 7 heeft en *informatiedichtheid*  $4/7$  (omdat vier van de zeven symbolen van een woord de informatie bevatten en de andere drie redundant zijn).

De code die we in dit voorbeeld hebben behandeld, heet de *Hamming code*. In de terminologie van coderingstheorie is het een  $[7,4]$ -code over  $F_2$ .

Op de CD-plaat is de informatiedichtheid  $24/32$  (dus  $3/4$ ). Daar bestaat het alfabet niet uit bits maar uit bytes. Dat impliceert dat we niet alleen de plaatsen waar fouten optreden moeten opsporen, maar ook moeten bepalen welke symbolen er op die plaatsen wél moeten staan.

De codes van de CD berusten op het feit dat bytes kunnen worden opgevat als elementen van een eindig lichaam, te weten  $F_{2^8}$ . Dit betekent dat we met de bytes de rekenkundige operaties optellen, aftrekken, vermenigvuldigen en delen (behalve door 00000000) kunnen uitvoeren. Bij de decodering berekent een algoritme de plaats en de waarde van de fout. Als op de  $i$ -de plaats het symbool  $a$  is veranderd in  $b$ , dan zeggen we dat de waarde van de gemaakte fout  $b - a$  is. Deze waarde wordt eerst berekend en dan van  $b$  afgetrokken. Zo vinden we  $a$  terug.

Om dit uit te leggen, voert veel te ver. Daarom geven we een voorbeeld van het soort code dat voor de CD wordt

gebruikt, maar we maken het rekenwerk veel eenvoudiger door met een ander alfabet te werken. We kiezen hiervoor de getallen 0 tot en met 30 waarmee we modulo 31 zullen rekenen. Met andere woorden: we gebruiken het lichaam  $F_{31}$ . De letters  $a$  tot en met  $z$  van ons alfabet zullen we vertalen naar 1 tot en met 26 en we gebruiken de overgebleven getallen 0, 27 tot en met 30 voor de spaties en wat leesstekens. Dan is het voorbeeld wat minder abstract.

### Voorbeeld

We coderen gewone tekst met een  $[6,4]$ -code over  $F_{31}$ . Dat wil dus zeggen dat we steeds vier letters of leesstekens tegelijk nemen, deze omzetten in getallen uit de reeks 0 tot en met 30 en daaraan twee redundante getallen toevoegen volgens een nog te noemen voorschrift.

Laten we het woord CODE in een woord van onze code omzetten. Eerst worden de vier letters vertaald naar 3, 15, 4, 5. Dit zullen de letters  $a_2, a_3, a_4, a_5$  zijn van het codewoord  $(a_0, a_1, \dots, a_5)$ . Het wordt dus  $(*, *, 3, 15, 4, 5)$  waarvan de eerste twee symbolen moeten worden berekend. We kiezen als rekenregels:

$$\begin{aligned} a_0 + a_1 + a_2 + a_3 + a_4 + a_5 &\equiv 0 & (\text{mod } 31) \\ a_1 + 2a_2 + 3a_3 + 4a_4 + 5a_5 &\equiv 0 & (\text{mod } 31) \end{aligned}$$

Uit de tweede regel vinden we  $a_1 = 1$  en daarna uit de eerste regel  $a_0 = 3$ . Het codewoord voor CODE is dus  $(3, 1, 3, 15, 4, 5)$  of zo men wil CACODE.

Stel dat we het woord XGFOPT =  $(24, 7, 6, 15, 16, 20)$  zien staan. We zien direct dat hier een fout in zit (misschien wel meer). De som van de zes symbolen is namelijk  $26 \pmod{31}$  en niet 0. Als we geen redundante symbolen hadden, stond er  $(6, 15, 16, 20) = \text{FOPT}$  maar dit woord fopt ons niet! We weten al wat de waarde van de fout is, namelijk  $-5$ . Immers  $26 + 5 \equiv 0 \pmod{31}$ . Waar zit die fout? We nemen aan dat  $a_i$  is veranderd in  $a_i + e$  waarbij we al weten dat  $e = -5$ . We vullen de getallen in in de tweede regel en vinden:

$$\begin{aligned} 1 \cdot 7 + 2 \cdot 6 + 3 \cdot 15 + 4 \cdot 16 + 5 \cdot 20 &= \\ = (1 \cdot a_1 + \dots + 5 \cdot a_5) + ie &\equiv ie \equiv 11 \equiv -20 \pmod{31} \end{aligned}$$

Nu kunnen we door deling van  $ie$  door  $e$  vinden dat  $i = 4$ . Dus moet op de vierde plaats niet 16 staan, maar 21 en het goede woord is FOUT (in code XGFOUT).

Ook in dit voorbeeld is sprake van een één fout verbeterende code. (Het moet duidelijk zijn dat we altijd één fout kunnen verbeteren.)

Bij de CD wordt twee keer achterelkaar gecodeerd (en later gedecodeerd) en wel met een  $[28,24]$ -code over  $F_{2^8}$  en daarna met een  $[32,28]$ -code. Elk van die codes kan twee fouten verbeteren, maar door een samenwerking kunnen ze meestal zeer veel meer. Het voert te ver om dit hier uit te leggen.

### Opgave

Een [12,8]-code over  $F_{31}$  (dus dezelfde informatiedichtheid als in ons voorbeeld) wordt als volgt gedefinieerd. Een woord van acht letters (of leestekens) wordt eerst omgezet in acht getallen uit 0 tot en met 30. Dit worden de letters  $a_4$  tot en met  $a_{11}$  van het codewoord ( $a_0, a_1, \dots, a_{11}$ ). De regels om  $a_0$  tot en met  $a_3$  te berekenen zijn:

$$\begin{aligned} a_0 + a_1 + \dots + a_{11} &\equiv 0 & (\text{mod } 31) \\ 1^k \cdot a_1 + 2^k \cdot a_2 + \dots + 11^k \cdot a_{11} &\equiv 0 & (\text{mod } 31) \\ \text{voor } k &= 1, 2, 3. \end{aligned}$$

Nu zien we ergens het woord:

(25, 18, 3, 27, 2, 5, 18, 20, 5, 4, 5, 14).

Als we de vier redundante symbolen negeren, staat er BERTEDEN. Wat zou dit moeten zijn? Misschien BETREDEN? Kan ook BESTEDEN zijn of wellicht VERMEDEN (allemaal ten hoogste twee fouten). We hebben te maken met een code die twee fouten in een woord kan verbeteren.

Neem aan dat  $a_i$  is veranderd in  $a_i + e$  en dat  $a_j$  is veranderd in  $a_j + f$ . Bereken dan  $i, j, e$  en  $f$ . Hiertoe moeten vier vergelijkingen opgesteld worden van het type  $e + f \equiv \alpha_0 \pmod{31}$  en  $i^k \cdot e + j^k \cdot f \equiv \alpha_k \pmod{31}$  met  $k = 1, 2, 3$ . Hieruit kan men een vierkantsvergelijking voor  $i$  vormen (waarvan  $j$  uiteraard de andere wortel is).

### Opgave

(Voor de liefhebbers van waarschijnlijkheidsrekening.) Een boek (200 bladzijden van elk ongeveer 3000 letters) wordt gedrukt met een gebrekkig procédé dat een kans van 0,001 heeft dat een verkeerde letter wordt gedrukt. Dit betekent dat we gemiddeld drie drukfouten per bladzijde in het boek kunnen verwachten! We besluiten om het boek te drukken met de [6,4]-code van ons voorbeeld, maar we willen het boek niet dikker maken. Dan moeten we dus niet 3000 letters op een bladzijde drukken, maar 4500 symbolen van de code. Tot onze schrik vertelt de drukker ons dat het gebruik van deze kleinere letter tot gevolg heeft dat de kans op een drukfout verdubbelt! Nu staan er dus gemiddeld negen per bladzijde. Maar ... het is een code die één fout in een zestal kan verbeteren. Dus zullen na het decoderen alleen die woorden waarin meer dan één fout zat nog steeds fout zijn.

1. Ga na dat we na decoderen slechts één fout per vier bladzijden kunnen verwachten.
2. Nu gebruiken we de [12,8]-code van de vorige opgave. Deze heeft ook informatiedichtheid  $2/3$ . Dus weer verwachten we negen fouten per bladzijde vóór decoderen. Ga na dat we na decoderen in het hele boek slechts twee drukfouten verwachten.

N.B. Ook in allerlei praktische toepassingen van codering (zoals op de compact disc) moeten we er rekening mee houden dat het toevoegen van redundante symbolen vaak de kans op fouten verhoogt. Pas na decoding zien we de winst.

## Het lezen van de plaat

We hebben al gezien dat één bemonstering van het analoge signaal (de muziek) vier bytes oplevert. Uit zes bemonsteringen wordt een informatiewoord van 24 letters (= bytes) gemaakt. Via de twee coderingen die we eerder noemden, worden hieruit codewoorden van 32 letters gemaakt. Daar komt nog één letter bij die voor 'control and display' informatie zorgt. Na EFM en toevoeging van de koppelingsbits hebben we dan  $33 \cdot 17$  kanaalbits. Daar worden nog 27 bits aan toegevoegd die voor wordsynchronisatie zorgen (met behulp van rijtjes die herkenbaar zijn omdat ze niet uit de codering kunnen ontstaan).

Bij het lezen (met een AlGaAs halfgeleider-laser) worden de overgangen van licht naar donker en omgekeerd (bij het teruggekaatste licht) gebruikt om de kanaalbits in de speler te registreren. Via een opgeslagen 'woordenboek' worden de codewoorden (nog met fouten erin) gereconstrueerd. Een algoritme verbetert de fouten. Daar waar de boel toch nog misgaat, worden interpolatie en maskeringstechnieken gebruikt. Ten slotte wordt de muziek weer in analoge vorm uitgezonden.

Bedenk dat al het rekenwerk zó snel moet gebeuren dat de luisteraar niet hoeft te wachten tot de muziek eindelijk begint!

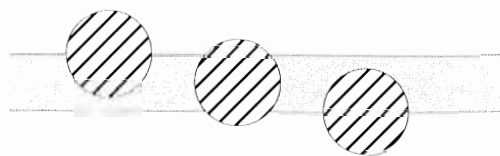
Bij een grammofoon zit de naald in het spoor op de plaat. Bij de CD is er slechts licht dat wordt teruggekaatst. Dat brengt heel wat problemen met zich mee. De afstand tot de plaat wordt constant gehouden doordat een deel van het teruggekaatste licht wordt afgebogen naar een servomechanisme dat via de afstand van twee lichtvlekjes ziet of de laser te dicht bij de plaat zit of te ver ervandaan. Dit is allemaal fysica. Er zijn twee andere regelmechanismen die sterk afhangen van het gebruik van de  $d-k$  begrensde rijen. Zoals gezegd, is een deel van het gereflecteerde licht helder en een deel donker. De lezer moet een beslissingsniveau hebben waarboven het 'helder' heet en waaronder 'donker'. Maar een vlek of vingerafdruk kan de lichtsterkte zo beïnvloeden dat alles donker wordt genoemd! Dit gebeurt niet, omdat de lezer weet dat de helft steeds licht moet zijn (ongeveer) en de helft donker. Dus kan het beslissingsniveau steeds worden bijgesteld om aan die voorwaarde te blijven voldoen (ondanks de vingerafdruk).

Van hetzelfde feit wordt gebruik gemaakt om iets nog verrassenders te regelen. De laservlek springt niet van een spoor naar een buurspoor dat toch slechts  $1,6 \mu\text{m}$  er naast ligt. Een van de systemen die hiervoor worden gebruikt, werkt als volgt.

In plaats van één lichtvlek zijn er drie: de middelste om het spoor af te lezen, één ( $A$ ) iets daarvoor en iets naar rechts en één ( $B$ ) iets daarachter en iets naar links. Als we 'licht' waarde 1 geven en 'donker' waarde 0, dan ziet de middelste vlek dus gemiddeld waarde  $1/2$ . De andere twee lopen voor de helft over volledig reflecterend mate-

riaal en zien dus een gemiddelde sterkte van  $\frac{3}{4}$ . Het mechanisme kijkt naar  $A - B$ . Dit moet waarde 0 hebben (gemiddeld). Wordt het positief, dan zit het drietal te ver naar rechts (en bij negatief, te ver naar links). Hiermee kan een servomechanisme voortdurend corrigeren en zo de middelste vlek op het te volgen spoor houden.

We hebben hiermee enkele van de technische hoogstandjes die het grote succes van de CD mogelijk maakten, beschreven. De lezer die meer details wil lezen, verwijzen we naar 'Compact Disc Digital Audio', *Philips Technisch Tijdschrift* 40, 1981/82, (9).



$A$  én  $B$  gemiddeld  $\frac{3}{4}$  en  $A - B = 0$

*Spoor volgen*

Prof. dr. J.H. van Lint, Stan Ackermans Instituut, TU Eindhoven

## Uitslag Escher prijsvraag

Op 7 november zijn tijdens de Ars et Mathesisdag te Baarn de prijzen uitgereikt van de Escherprijsvraag die georganiseerd was door de stichting Ars et Mathesis en het wiskundetijdschrift *Pythagoras*.

De resultaten waren overweldigend: 130 inzendingen bestaande uit ruim vijfhonderd werkstukken.

### De jurering

De jury bestond uit Jos de Mey (Ars et Mathesis), Joke Mestdagh (vormgeefster *Pythagoras*), Zsafia Ruttkay (CWI), Fred van der Blij (Ars et Mathesis), Eduard Looijenga (UU) en Hans van Lint (juryvoorzitter).

Opvallend was dat in alle categorieën bijzonder mooie en originele werkstukken zijn ingezonden.

De jury was onder de indruk van de kwaliteit van het vele werk. Bij de beoordeling is gelet op de wiskundige en de esthetische kanten van ieder werk en originaliteit was van groot belang; inzendingen die sterk leken op reeds bestaande werken van Escher werden daarom terzijde gelegd.

### Virtuele tentoonstelling en poster

De winnende inzendingen worden gepubliceerd in het februari-nummer van *Pythagoras*. Zoveel mogelijk inzendingen zullen worden gefotografeerd en tentoongesteld op de homepage van *Pythagoras*. Op dit moment worden de mogelijkheden onderzocht om de mooiste inzendingen (prijswinnaars en genomineerden) in het land tentoon te stellen. De inzendingen zullen aan de makers geretourneerd worden, maar niet eerder dan in het voorjaar van 1999.

Van de winnende inzending in de open groep zal een Pythagoras-poster worden gemaakt. Gereed: begin schooljaar 1999-2000.

### Sponsors

De organisatie van de prijsvraag is mogelijk gemaakt door een subsidie van de Stichting Weten.

### De prijzen

**Eerste prijs.** Een geldbedrag van vijfhonderd gulden (klassenprijs duizend gulden) en een kunstwerk: 'Onmogelijke kubus'. Ontwerp: Dick Baas Becking, Popke Bakker en Koos Verhoeff, 1986. Vervaardigd door Popke Bakker.

**Tweede prijs.** Een geldbedrag van tweehonderdvijftig gulden (klassenprijs vijfhonderd gulden) en een kunstwerk: 'Dwarsverstek met 7, 12 of 15 stukjes'. Ontwerp: Koos Verhoeff. Vervaardigd door Hans de Koning.

**Derde prijs.** Een geldbedrag van honderd gulden (klassenprijs tweehonderdvijftig gulden) en een kunstwerk: 'Boomspiegeling'. Ontwerp: Ton Schotten.

### De uitslag

**Categorie Onderbouw (t/m 14 jaar).** Prijs ter beschikking gesteld door de NVVW.

1. Daan Juttman, Haarlem
2. Ciry Boon, Maassluis
3. Joris Brakke, Amsterdam

**Categorie Bovenbouw.** Prijs ter beschikking gesteld door het Freudenthal Instituut.

1. Toni Jonkers, Zoetermeer
2. Geoffrey de Smet, Oostakker (Gent), België
3. Daan de Jong, Utrecht

**Open categorie.** Prijs ter beschikking gesteld door het wiskundetijdschrift *Pythagoras*, via een subsidie van de stichting de Wageningse Methode.

1. Simon Biesheuvel, Weesp
2. Ton Schotten, Heerhugowaard
3. C. van der Stelt, Bloemendaal

**Categorie Schoolklassen.** Prijs ter beschikking gesteld door de M.C. Escherstichting.

1. 1e klas bovenbouw, Onbevleete Ontvangenis, Tongeren, België
2. Klas 1A, Katholieke Scholengemeenschap Etten-Leur (K.S.E.), Etten-Leur
3. Klas 4/5, St. Michael College (S.M.C.), Zaandam