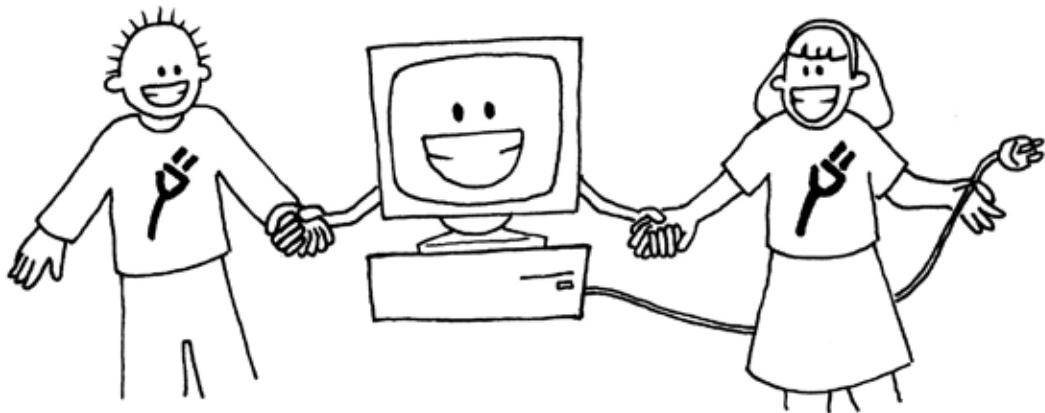


# **COMPUTER SCIENCE** *Unplugged*

**Voor leerkrachten  
Deel I, II en III**



**Een verdiepings- en uitbreidingsprogramma  
voor het lager onderwijs.**

Door:

Tim Bell, Ian H. Witten en Mike Fellows  
Bewerking voor onderwijs door Robyn Adams en Jane McKenzie

Illustraties: Matt Powell

Nederlandse vertaling:  
Suzanne van Kessel,  
Marijn van der Meer,  
Joek van Montfort en  
Cobie van de Ven



# Introductie

---

Computers zijn overal. We leren ze van jongen af aan te gebruiken en velen van ons gebruiken ze dagelijks. Maar hoe werken ze eigenlijk? Hoe denken ze? En hoe maken mensen ze iedere keer sneller en beter? Informatica (computerkunde) is het boeiende vakgebied dat deze vragen onderzoekt. Door de toegankelijke en leuke opdrachten in dit boek, bedacht voor leerlingen van alle leeftijden, leren de leerlingen hoe de computer in basis werkt - zonder achter een computer te moeten zitten!

Dit boek kan prima gebruikt worden in verdiepings- en uitbreidingsprogramma's of zelfs in het reguliere lesprogramma. Je hoeft echt geen computerexpert te zijn om dit programma te begrijpen of over te brengen aan de leerlingen. Dit boek bevat verschillende activiteiten met achtergrondinformatie. Antwoorden voor alle opdrachten zijn bijgevoegd en iedere opdracht eindigt met de vraag "waar gaat dit eigenlijk over?", een beknopte uitleg over de relevantie van de opdrachten.

Veel activiteiten behoren tot het rekendomein, bijvoorbeeld binaire getallen, kaarten en grafieken, patronen, sorteerproblemen en cryptografie (geheimschrift). Anderen vinden dit meer tot het technologiedomein horen, het leren begrijpen hoe computers werken. De kinderen worden actief betrokken in communicatie, problemen oplossen, creativiteit en denkvaardigheden in een betekenisvolle context.

In aanvulling op dit boek, biedt het "Unplugged" project veel gratis online bronnen, inclusief video's, foto's en extra materiaal op [csunplugged.org](http://csunplugged.org) of [csunplugged.nl](http://csunplugged.nl).

Dit boek is geschreven door drie informaticadocenten en twee leerkrachten en is gebaseerd op onze ervaringen in het klaslokaal. We hebben gemerkt dat veel belangrijke kennis overgebracht kan worden zonder computer - in sommige gevallen vormt de computer zelfs een te grote afleiding. Dus trek de stekker uit je computer en ga aan de slag om echt te leren waar informatica over gaat!

**Dit boek is gratis te downloaden voor persoonlijk en educatief gebruik dankzij een genereuze bijdrage van Google, Inc. Het wordt verspreid onder een Creative Commons Attribution-NonCommercial-ShareAlike licence, wat betekent dat je dit boek vrijelijk kan delen (kopiëren, distribueren en verzenden). Dit geeft je ook het recht het boek aan te passen of delen over te nemen. Maar alleen onder de volgende condities: er moet een verwijzing naar de auteurs gemaakt worden, je dit boek voor niet-commerciële doeleinden gebruikt en als je dit boek aanpast, verandert of stukken eruit haalt, moet je dit delen onder dezelfde of soortgelijke licentie. Meer specificaties over deze licentie vind je online door te zoeken naar: CC BY-NC-SA 3.0.**  
**We moedigen aan dit boek te gebruiken voor educatieve doeleinden. Je mag je eigen kopie printen en de werkbladen onder je leerlingen verspreiden. We ontvangen graag vragen en suggesties, welke naar de auteurs gezonden kunnen worden (kijk op [csunplugged.org](http://csunplugged.org)). Dit boek is vertaald in verschillende andere talen. Kijk op de website voor de beschikbaarheid.**

## Woord van dank

---

Veel kinderen en leerkrachten hebben ons geholpen dit programma beter te maken. De kinderen en de leerkrachten van de South Park School (Victoria, BC), Shirley Primary School, Ilam Primary School en Westburn Primary School (Christchurch, Nieuw Zeeland) waren de proefkonijnen voor veel opdrachten. We zijn bijzonder veel dank verschuldigd aan Linda Picciotto, Karen Able, Bryon Porteous, Paul Cathro, Tracy Harrold, Simone Tanoa, Lorraine Woodfield, en Lynn Atkinson die ons uitnodigden in hun klassen en ons opbouwende kritiek gaven waardoor wij de opdrachten verder konden verbeteren. Gwennda Bensemman heeft verschillende opdrachten uitgeprobeerd en heeft vervolgens goede suggesties voor verbetering gegeven. Richard Lynders en Sumant Murugesh hebben ons geholpen in de testfase. Delen van de cryptografie opdrachten zijn ontwikkeld door Ken Noblitz. Een aantal opdrachten zijn ontwikkeld onder auspiciën van de Victoria “Mathmania” groep, met hulp van Kathy Beveridge. Eerdere versies van de illustraties zijn gemaakt door Malcolm Robinson en Gail Williams, en we hebben ook geprofiteerd van de kennis van Hans Knutson. Matt Powell heeft ook welkome adviezen gegeven tijdens het ontwikkelen van het “Unplugged” project. We zijn het Brian Mason Scientific and Technical Trust dankbaar voor de sponsoring in de beginfase van dit boek.

Speciale dank gaat uit naar Paul en Ruth Ellen Howard, die veel opdrachten hebben getest en waardevolle suggesties gaven. En naar Peter Henderson, Bruce McKenzie, Joan Mitchell, Nancy Walker-Mitchell, Gwen Stark, Tony Smith, Tim A. H. Bell, Mike Hallett, en Harold Thimbleby die ook met waardevolle aanvullingen kwamen.

We zijn zeer veel dank verschuldigd aan onze families: Bruce, Fran, Grant, Judith, en Pam voor hun steun, en Andrew, Anna, Hannah, Max, Michael, en Nikki die een inspiratie waren voor dit werk, en veelal de eerste proefkonijnen waren.

We moeten ook Google Inc. bijzonder veel danken voor het sponsoren van het Unplugged project, en daarmee ons de mogelijkheid gaven om deze editie gratis via het web te verspreiden.

Aanvullingen en opmerkingen over de opdrachten zijn welkom. De auteurs kunnen benaderd worden via [csunplugged.org](http://csunplugged.org).

Ook voor de Nederlandse vertaling moeten wij Google Inc. danken voor een gift voor het vertalen in het kader van Codeweek.eu. Ook zouden we graag Marije van der Meer en Jeroen Guns bedanken voor de vertalingen bij de tekstcompressieopdrachten. En Marco Jordan, Harald van Brederode, Marcelle Peeters en Amber Beckers voor het nalezen van de vertaling.

Voor verbeteringen en suggesties voor een betere vertaling; ga naar [www.csunplugged.nl](http://www.csunplugged.nl).

# Inhoud

---

Introductie .....	2
Woord van dank .....	3
<b>Deel I - Data: het ruwe materiaal - informatie weergeven .....</b>	<b>5</b>
Activiteit 1 - Binaire getallen.....	7
Activiteit 2 - Kleur door getallen .....	18
Activiteit 3 - Zeg het opnieuw! .....	27
Activiteit 4 - De “Magische truc” .....	34
Activiteit 5 - Twintig keer raden .....	40
<b>Deel II - Computers aan het werk - algoritmes.....</b>	<b>47</b>
Activiteit 6 - Slagschepen .....	49
Activiteit 7 - Lichtste en zwaarste .....	69
Activiteit 8 - Sorteer netwerken.....	76
Activiteit 9 - Modderstad .....	81
Activiteit 10 - Het sinaasappelspel .....	86
<b>Deel III - computers vertellen wat te doen - weergeven van een actieplan.....</b>	<b>89</b>
Activiteit 11 - Schateiland.....	91
Activiteit 12 - Marsorders .....	105

# **Deel I**

---

## **Data: het ruwe materiaal**

informatie weergeven

---

# Data: Het ruwe materiaal

---

## Hoe slaan we informatie op in computers?

Het woord computer komt van het Latijnse woord computare, wat betekent rekenen of optellen, maar computers zijn tegenwoordig veel meer dan alleen grote rekenmachines. Ze kunnen een bibliotheek zijn, zij helpen ons met schrijven, zij vinden informatie voor ons, spelen muziek en laten films zien. Hoe slaan ze al deze informatie op? Geloof het of niet, de computer gebruikt er maar twee dingen voor: een nul en een één!

## Wat is het verschil tussen data en informatie?

Data zijn het ruwe materiaal, de getallen waar de computer mee werkt. Een computer zet zijn data om in informatie (woorden, getallen en afbeeldingen) die jij en ik kunnen begrijpen.

## Hoe kunnen getallen, woorden en afbeeldingen omgezet worden in nullen en enen?

In dit deel leren we over binaire getallen, hoe computers afbeeldingen tekenen, wat de meest effectieve manier is om heel veel data op te slaan, hoe we fouten kunnen vermijden en hoe we de grootte bepalen van een hoeveelheid informatie die we op willen slaan.



# Activiteit 1

---

## Tel de punten—Binaire Getallen

### Samenvatting

Data in de computer worden opgeslagen als een serie van nullen en enen. Hoe kunnen we woorden en getallen weergeven met alleen deze twee symbolen?

### Kerndoelen

- Rekenen: Getallen groep 4 en hoger. Het ontdekken van getallen in andere grondtallen. Het weergeven van getallen in grondtal 2.
- Rekenen: Algebra groep 4 en hoger. Een patroon voortzetten en regels bepalen voor een patroon. Patronen en relaties in de machten van 2.

### Vaardigheden

- Tellen
- Paren
- Reeksen

### Leeftijd

- 7 jaar en ouder

### Materiaal

- Maak een set van 5 binaire kaarten voor de demonstratie. Je kan bijvoorbeeld A4'tjes nemen en met ronde stickers of dikke zwarte stift de punten erop zetten.

Iedere leerling heeft nodig:

- Een set van 5 kaarten: Kopieer *Kopie Kaart: Binaire getallen* en knip 5 kaarten uit.
- *Werkblad: Binaire getallen*

Optionele uitbreidingsopdrachten:

- *Werkblad: Werken met binaire getallen.*
- *Werkblad: Sturen van geheime boodschappen.*
- *Werkblad: Fax machines en modems.*
- *Werkblad: Tellen tot boven 31.*
- *Werkblad: Meer over binaire getallen.*

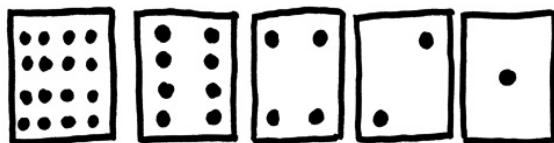
# Binaire getallen

---

## Introductie

Voordat je het werkblad uitdeelt, is het verstandig om de basisvaardigheden voor te doen aan de hele groep.

Voor deze opdracht heb je een set van 5 kaarten nodig, zoals hieronder, met punten aan de ene kant en blanco aan de andere zijde. Kies 5 leerlingen om de kaarten vast te houden voor de klas. De kaarten moeten in deze volgorde staan:



## Vraag

Wat valt je op aan het aantal punten dat er op de kaarten staat? (Iedere kaart heeft 2 keer zoveel punten als de kaart aan de rechterzijde.)

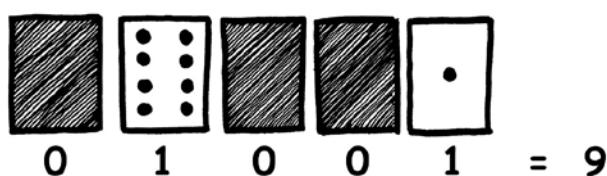
Hoeveel punten zouden er op de volgende kaart komen als we links nog een kaart zouden toevoegen? (32) En de volgende...?

We kunnen deze kaarten gebruiken om getallen te maken door de 5 leerlingen de kaarten om te laten draaien. Vraag een leerling om het getal 6 te maken, welke leerlingen in de rij moeten hun kaart omdraaien? (4-punten kaart en 2-punten kaart laten zien) En dan 15 (8-, 4-, 2-, en 1-punten kaart), en dan 21 (16, 4 en 1)...

Probeer nu te tellen vanaf 0...

De rest van de klas moet goed kijken hoe de kaarten wisselen om te zien of ze een patroon ontdekken in hoe de kaarten gedraaid worden (iedere kaart draait half zo vaak als degene rechts van hem). Je kan dit ook uitproberen met meer groepjes.

Wanneer een binaire getallen kaart niet getoond wordt staat dit voor een NUL. Wanneer deze wel getoond wordt staat dit voor een EEN. Dit is het binaire getallen systeem.



Vraag de leerlingen om 01001 te maken. Welk getal is dat in decimalen? (9) En wat is 17 in binair? (10001)

Probeer nog een aantal mogelijkheden tot ze het concept begrijpen.

Er zijn vijf optionele extra vervolgopdrachten, te gebruiken voor verdieping. Pas de moeilijkheidsgraad aan bij het niveau van de leerlingen.

# Werkblad: Binaire getallen

---

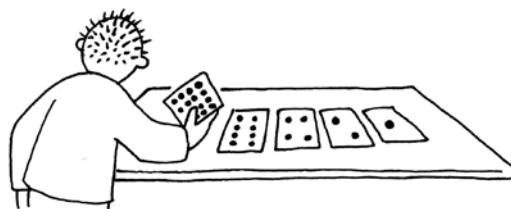
## Leer opnieuw tellen

Dus, je denkt te weten hoe je moet tellen? Hier is een nieuwe manier van tellen!

Wist je dat computers alleen de 0 en 1 gebruiken? Alles wat je ziet of hoort op de computer - woorden, foto's, getallen, films en zelfs geluiden worden opgeslagen met alleen deze twee getallen! De volgende opdrachten leren je hoe je geheime boodschappen naar je vrienden kunt sturen op precies dezelfde manier als een computer dat doet.

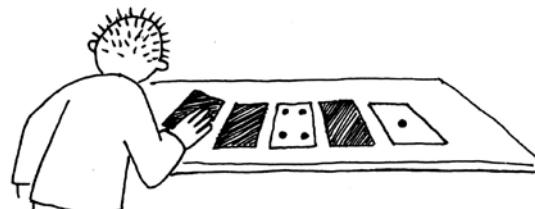
## Instructie

Knip de kaarten uit van je blaadje en leg ze voor je op tafel met de 16-punten kaart links en in deze volgorde:



Kijk goed of ze echt in dezelfde volgorde liggen.

Draai nu de kaarten om zodat er precies 5 punten zichtbaar zijn - hou de kaarten in dezelfde volgorde!

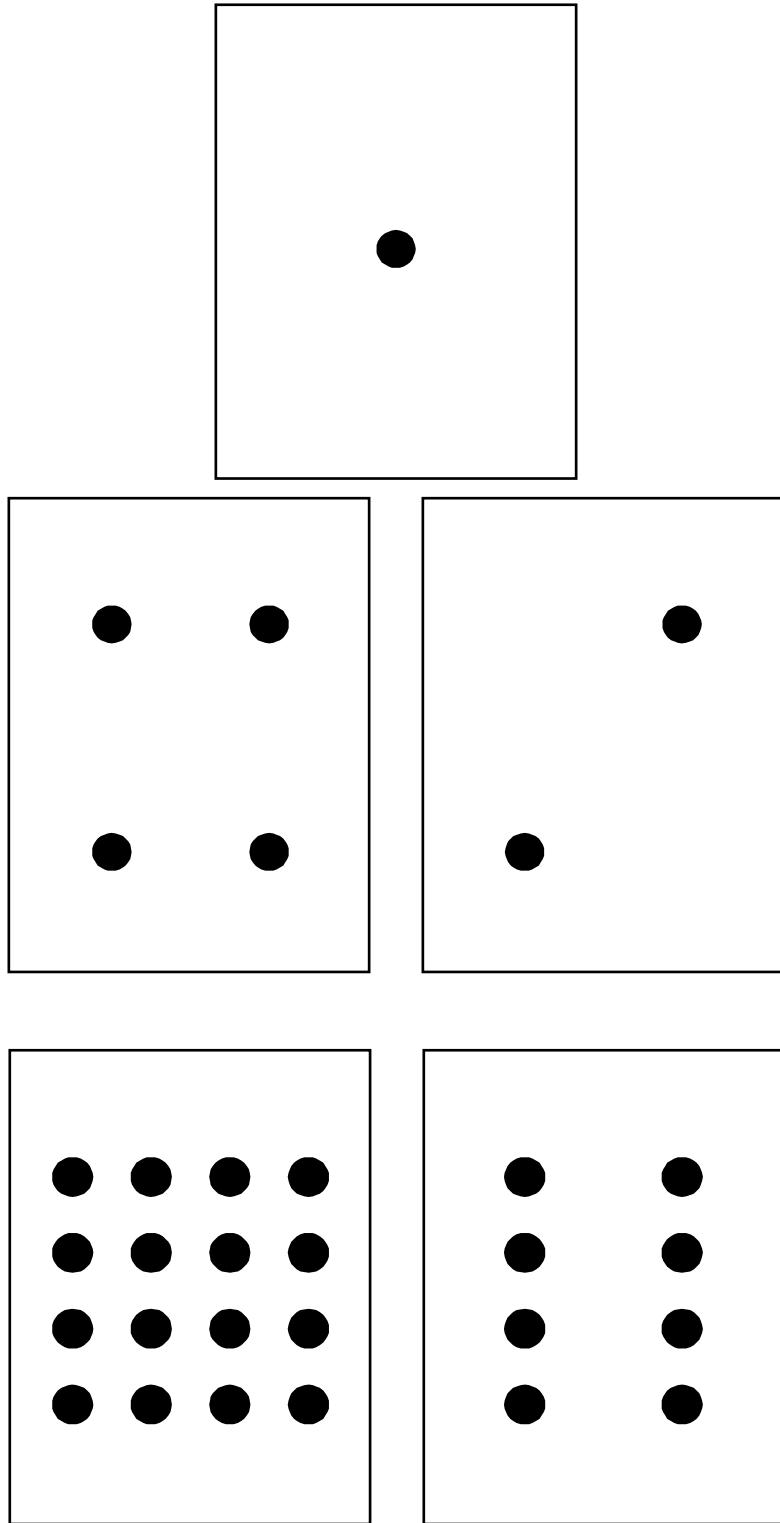


Zoek uit hoe je 3, 12 of 19 krijgt. Zijn er meer manieren om een getal te krijgen? Wat is het grootste getal dat je kan maken? Wat is het kleinste? Is er een getal dat je niet kan maken tussen het grootste en het kleinste?

**Extra vraag:** probeer de getallen 1, 2, 3, 4 in volgorde te maken. Zie je een logische manier om de kaarten om te draaien zodat er iedere keer een bij opgeteld wordt?

## Kopie Kaart: Binaire Getallen

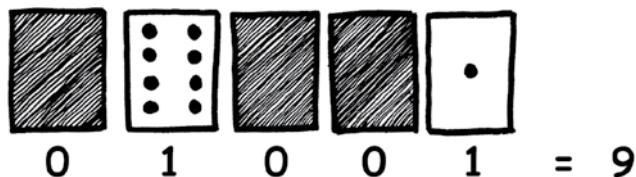
---



## Werkblad: Werken met binaire getallen

---

Het binaire systeem gebruikt nullen en enen om te bepalen of een kaart is omgedraaid of niet. Nul betekent dat de kaart ondersteboven ligt en één betekent dat je de punten ziet.  
Bijvoorbeeld:



Kan je nu uitvinden wat **10101** is? En **11111**?

Op welke dag van de maand ben je jarig? Schrijf dat in een binair getal. En wat zijn de verjaardagen van je vrienden in binair?

Probeer deze gecodeerde getallen te ontcijferen:

$$\boxed{\times} \ \checkmark \ \boxed{\times} \ \boxed{\times} \ \checkmark = \\ (\checkmark=1, \times=0)$$

$$\overset{\uparrow}{\square} \ \overset{\downarrow}{\square} \ \overset{\uparrow}{\square} = \\ (\uparrow=1, \downarrow=0)$$

$$\circlearrowleft \circlearrowright \circlearrowleft \circlearrowright \circlearrowleft = \\ (\circlearrowleft=1, \circlearrowright=0)$$

$$\overset{\uparrow}{\square} \ \overset{\uparrow}{\square} \ \overset{\uparrow}{\square} = \\ (\uparrow=1, \square=0)$$

$$\odot \odot \odot \odot \odot = \\ (\odot=1, \odot=0)$$

$$\leftarrow \rightarrow \leftarrow \rightarrow \leftarrow = \\ (\leftarrow=1, \rightarrow=0)$$

$$+ + \times + = \\ (+=1, \times=0)$$

$$\circlearrowleft \circlearrowright \circlearrowleft \circlearrowright \circlearrowleft = \\ (\circlearrowleft=1, \circlearrowright=0)$$

$$\blacktriangle \blacktriangledown \blacktriangle \blacktriangledown \blacktriangle = \\ (\blacktriangle=1, \blacktriangledown=0)$$

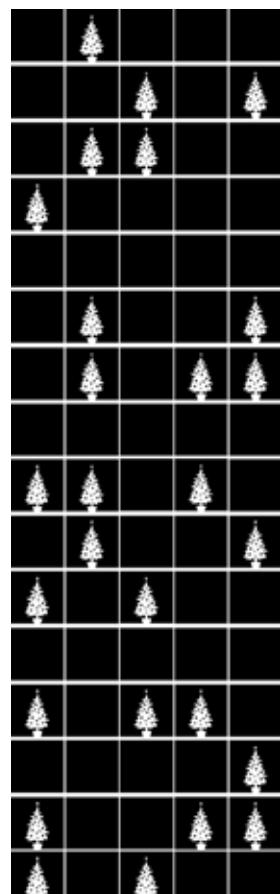
$$\spadesuit \spadesuit \spadesuit \spadesuit \spadesuit = \\ (\spadesuit=1, \clubsuit=0)$$

**Extra vraag:** Maak stokken of stroken papier van een lengte 1, 2, 4, 8, en 16 centimeter en laat zien dat je nu alles kan meten tot 31 centimeter. Of probeer een volwassenen te verrassen door hem of haar uit te leggen dat je voor het wegen op een balans van zware dingen toch maar een paar gewichten nodig hebt!

## Werkblad: Geheime boodschappen sturen

---

Tom zit opgesloten in een hoog gebouw. Het is bijna Kerst en hij wil snel naar huis voor zijn cadeautjes. Wat kan hij doen? Hij heeft geprobeerd te bellen en zelfs geschreeuwd, maar niemand neemt op of hoort hem. Aan de overkant ziet hij nog een vrouw achter haar computer zitten. Hoe kan hij haar een boodschap sturen? Tom kijkt rond om te zien wat hij kan gebruiken. Dan heeft hij een briljant plan - hij kan de kerstlampjes uit de boom gebruiken om haar een boodschap te sturen! Hij zet alle kerstbomen in iedere kamer voor het raam en stopt de stekkers van de lichtjes in het stopcontact. Nu kan hij ze aan- of uitzetten en kan hiermee binaire code versturen. Hij weet zeker dat de vrouw aan de overkant het begrijpt. Begrijp jij het ook?



a	b	c	d	e	f	g	h	i	j	k	l	m
1	2	3	4	5	6	7	8	9	10	11	12	13
n	o	p	q	r	s	t	u	v	w	x	y	z
14	15	16	17	18	19	20	21	22	23	24	25	26

## Werkblad: E-mail en Internet

---

Computers zijn met elkaar verbonden door het internet en gebruiken ook de binaire code om opdrachten en boodschappen naar elkaar te versturen. Het enige verschil is dat zij geluid of licht gebruiken. Bij geluid is een hoog piepje een 1 en een lage piep is een 0. Deze piepjess gaan razendsnel achter elkaar. Zo snel dat wij alleen een afgrijselijke "srkieeeeh" horen. Als je dit nog nooit gehoord hebt kijk dan of er nog iemand een fax heeft staan (tegenwoordig nog maar weinig gebruikt) en bel deze met een gewone telefoon. Je kan ook even naar [tinyurl.com/faxgeluid](http://tinyurl.com/faxgeluid) gaan om het geluid te horen. Tegenwoordig wordt er steeds vaker licht gebruikt (licht gaat sneller dan geluid), licht aan is 1, licht uit is 0 en dat heel snel achter elkaar.



Gebruik dezelfde code als Tom gebruikte met de kerstverlichting, stuur een e-mail naar een klasgenoot met een lamp of je stem, je hoeft niet zo snel te zijn als een echte computer.



## Werkblad: Tellen boven de 31

---

Kijk nog eens naar je binaire kaarten. Als we de volgende in de rij zouden maken, hoeveel punten zouden daarop staan? En die daarna? Welke regel gebruik je om nieuwe kaarten te maken? Zoals je kunt zien kun je met maar een paar kaarten tot grote getallen tellen.

Als je nog eens naar het patroon kijkt, vind je een interessante relatie tussen de kaarten:

**1, 2, 4, 8, 16...**

Tel deze eens op:  $1 + 2 + 4 = ?$  Wat krijg je dan?

En bij dit:  $1 + 2 + 4 + 8 = ?$

Wat gebeurt er als je alle getallen van de kaarten vanaf het begin optelt?

Je hebt vast bij de kleuters ooit gezongen: “Tien, tien, tien, laat al je vingers zien”? Maar nu heb je een manier om met je vingers tot veel meer te tellen dan 10, en daar hoef je geen Marsmannetje met heleboel vingers voor te zijn. Als je het binaire systeem gebruikt en iedere vinger gebruikt in plaats van een van de kaarten met de punten kan je met 1 hand van 0 tot 31 tellen. Dat zijn 32 getallen (Vergeet niet dat 0 ook een getal is).

Probeer te tellen met je vingers. Een vinger omhoog is een 1 en omlaag is een 0.

Met twee handen kom je zelfs van 0 tot 1023! Dat zijn 1024 getallen!

Als je je tenen goed zou kunnen buigen (nu zou het wel handig zijn als je een Marsmannetje was) kan je zelfs nog hoger komen. Als één hand tot 32 getallen kan tellen, en twee handen tot  $32 \times 32 = 1024$  getallen kan tellen, tot hoe ver kom je dan als je je tenen ook nog zou kunnen gebruiken?



## Werkblad: Meer over binaire getallen

---

Een andere interessante eigenschap van binaire getallen zie je als je rechts van de serie getallen een 0 zet. Als we met ‘gewone’ getallen (decimale getallen heet dat) een 0 rechts zetten wordt het getal 10 keer zo groot. Bijvoorbeeld 9 wordt 90 en 30 wordt 300.

Maar wat gebeurt er als we een 0 achter een binair getal zetten? Probeer dit eens:

$$\mathbf{1001 = 9} \quad \mathbf{10010 = ?}$$

Doe het met nog een aantal getallen om te zien of je idee klopt. Wat is de regel en waarom denk je dat dit gebeurt?

Elke kaart die we tot nu toe hebben gebruikt stelt een ‘bit’ voor op de computer (‘bit’ is een afkorting van ‘binary digit’, binair getal in het Engels). Dus onze alfabetcode die we gebruikten tot nu toe bestond maar uit 5 kaarten of ‘bits’. Maar een computer moet ook weten welke letters als hoofdletter moeten, komma’s en punten herkennen en bijzondere symbolen zoals € of /.

Kijk eens naar een toetsenbord en ga eens na hoeveel karakters een computer moet kunnen weergeven. Hoeveel bits heeft een computer dan nodig om al deze karakters te bewaren?

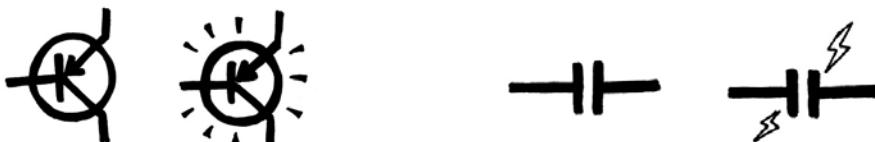
De meeste computers gebruiken tegenwoordig een weergave die ASCII (American Standard Code for Information Interchange = Amerikaanse Standaard Code voor Informatie Uitwisseling) heet, die gebaseerd is op dit aantal bits per karakter, maar er zijn ook landen die langere codes moeten gebruiken omdat ze meer karakters in het alfabet hebben.



## Waar gaat dit eigenlijk over?

---

Tegenwoordig gebruiken computers allemaal het binaire systeem om informatie weer te geven. Het heet binair omdat er maar twee getallen worden gebruikt. Het wordt ook wel het tweetallig stelsel genoemd (in het dagelijks leven gebruiken we het decimale, tientallige stelsel). Elke 0 of 1 heet een bit (binary digit). Een bit wordt in de computer meestal weergegeven in het geheugen door een transistor die aan of uit staat of een condensator die geladen of ontladen is.



Wanneer data door een telefoondraad worden gezonden, worden hoge en lage tonen gebruikt voor nullen en enen. Op magnetische schijven, zoals een harde schijf, worden de bits weergegeven door de richting van het magnetische veld op de oppervlakte van de schijf, Noord-Zuid of Zuid-Noord.

N S N S S N N S S N S N S

Audio CD's, CD-ROM's en DVD's bewaren bits optisch—het deel van het oppervlakte dat voor 1 bit staat reflecteert het licht of juist niet



Een bit alleen zegt niet zo veel, dus zijn bits meestal gegroepeerd in groepjes van 8 die de getallen 0 tot 255 kunnen weergeven. Een groep van 8 bits heet een byte.

De snelheid van een computer wordt bepaald door het aantal bits dat het in een keer kan verwerken. Een 32-bit computer bijvoorbeeld, kan 32-bits in een handeling verwerken, terwijl een 16-bit computer de 32-bit getallen eerst in kleinere stukken moet delen wat het dus langzamer maakt.

Uiteindelijk zijn het alleen bits en bytes die een computer gebruikt om tekst, getallen en alle andere informatie op te slaan en te verzenden. In een aantal latere opdrachten zullen we zien hoe de andere soorten informatie worden opgeslagen.

# **Oplossingen en hints**

---

## **Binaire getallen**

Voor 3 gebruik je kaart 2 en 1

Voor 12 gebruik je kaart 8 en 4

Voor 19 gebruik je kaart 16, 2 en 1

Er is maar één manier om alle cijfers te maken.

Het grootste getal dat je kan maken is 31. Het kleinste is 0. En je kan ieder getal er tussen in en ieder getal op maar één manier maken.

**Extra vraag:** Om ieder willekeurig getal met één te verhogen, draai je vanaf rechts alle kaarten om de beurt om tot je een kaart met de punten omhoog draait.

## **Werken met binaire getallen**

$10101 = 21, 11111 = 31$

## **Geheime boodschappen sturen**

Gecodeerd bericht: HELP IK ZIT VAST

## **Tellen boven de 31**

Als je alle getallen vanaf het begin optelt, is de uitkomst altijd één minder dan het volgende getal in de rij.

Met tenen erbij kom je tot:  $1024 \times 1024 = 1.048.576$  getallen—van 0 tot 1.048.575!

## **Meer over binaire getallen**

Wanneer je een nul toevoegt aan de rechterkant van een binair getal wordt het getal twee keer zo groot.

Alle plekken waar een één staat in de rij zijn nu twee keer zoveel waard als op hun vorige positie en daarom wordt het totaal verdubbeld. (Als je een nul rechts toevoegt bij decimale getallen wordt het getal tien keer zo groot).

Een computer heeft zeven bits nodig om alle karakters op te slaan. Met zeven bits kunnen er tot 128 karakters worden opgeslagen. Normaal gesproken worden de zeven bits opgeslagen in een 8-bit byte, waarbij één bit niet gebruikt wordt.

# Activiteit 2

---

## Kleuren met getallen — Afbeeldingen weergeven

### Samenvatting

- Computers slaan tekeningen, foto's en andere afbeeldingen op door het gebruik van getallen. De volgende opdracht laat zien hoe.

### Kerndoelen

- Rekenen: Meetkunde groep 4 en hoger. Vormen en ruimte ontdekken.

### Vaardigheden

- Tellen
- Tekenen

### Leeftijd

- 7 jaar en ouder

### Materialen

- Voorbeeld voor in de klas: Kleur met getallen*

Iedere leerling heeft nodig:

- Werkblad: Pixels tekenen*
- Werkblad: Maak je eigen afbeelding*

Optionele uitbreidingsopdracht:

- Werkblad: Maak je eigen afbeelding 2*

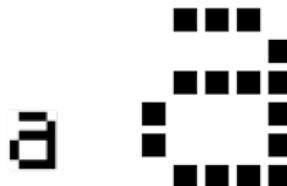
# Kleur door getallen

---

## Introductie

### Discussie vragen

1. Wanneer moeten computers beelden opslaan? (Een tekenprogramma, voor games en foto's en films.)
2. Hoe kunnen computers beelden opslaan met alleen nullen en enen?



Computer schermen zijn verdeeld in een raster van hele kleine puntjes die pixels (komt van picture elements) worden genoemd.

In een zwart-wit afbeelding is iedere pixel zwart of wit.

De letter ‘a’ is hier vergroot om de pixels te laten zien. Als een computer een beeld moet opslaan onthoudt hij precies welke puntjes zwart moeten worden en welke wit.

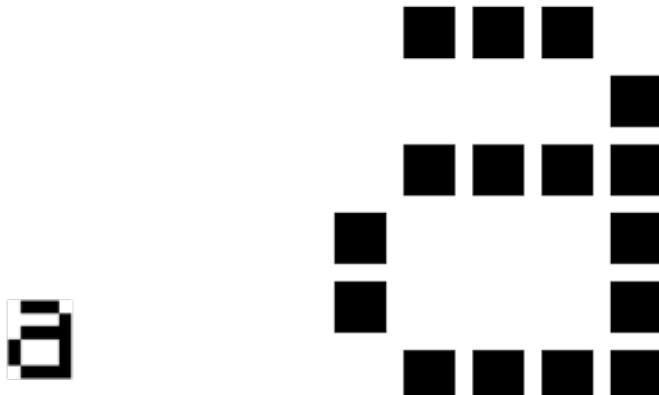
					1,3,1
					4,1
					1,4
					0,1,3,1
					0,1,3,1
					1,4

De afbeelding hierboven laat zien hoe een beeld weergegeven kan worden door getallen.

De eerste regel bestaat uit 1 witte pixel, dan 3 zwarten en dan weer een witte. Dus de eerste regel wordt bewaard als 1,3,1. Het eerste nummer geeft altijd het aantal witte pixels weer. Als de eerste pixel zwart is begint de regel met een nul. Op het werkblad *Maak je eigen afbeelding* staat een aantal beelden die de leerlingen kunnen ontcijferen met de bovenstaande methode.

## **Voorbeeld voor in de klas: kleur door getallen**

---



Een letter “a” van een computerscherm en dan vergroot zodat de pixels te zien zijn die de letter maken.

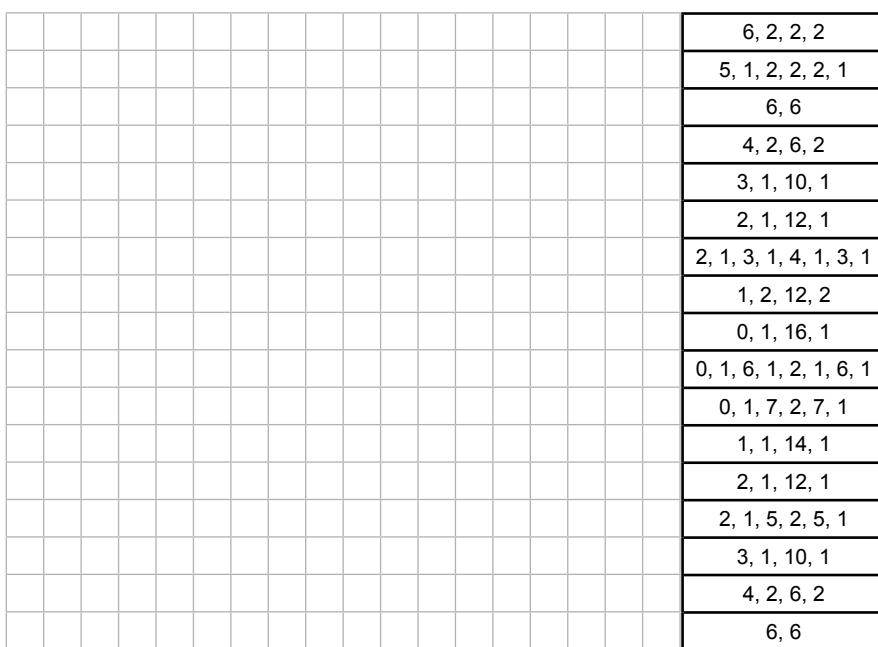
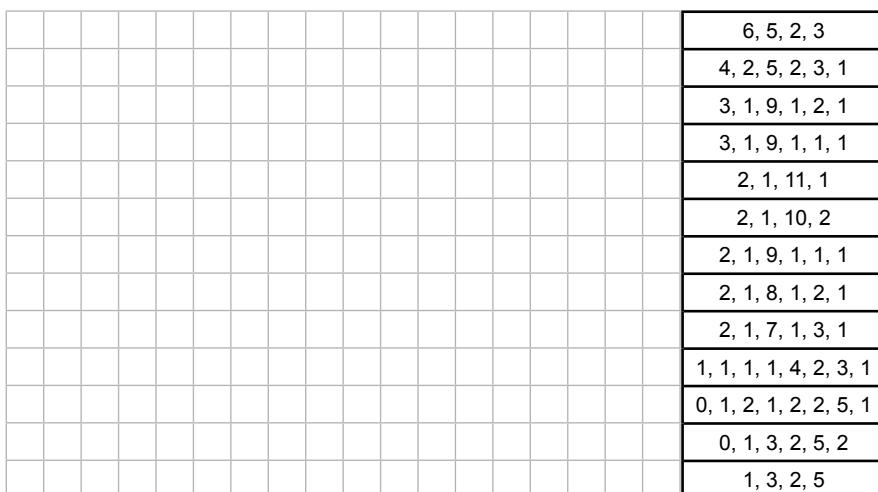
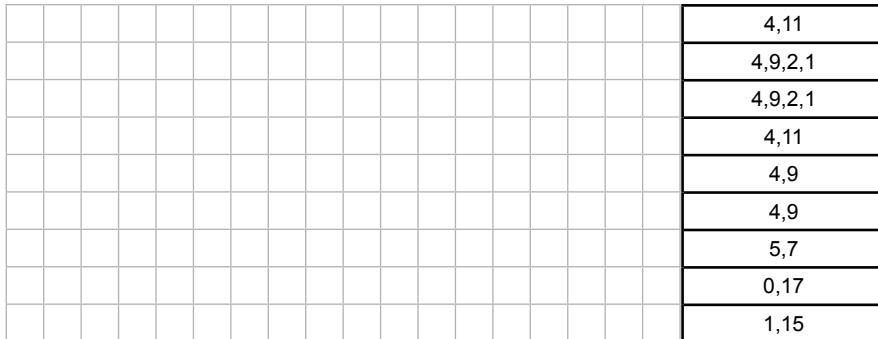
					1,3,1
					4,1
					1,4
					0,1,3,1
					0,1,3,1
					1,4

Dezelfde afbeelding nu in getallen.

# Werkblad: Pixels tekenen

---

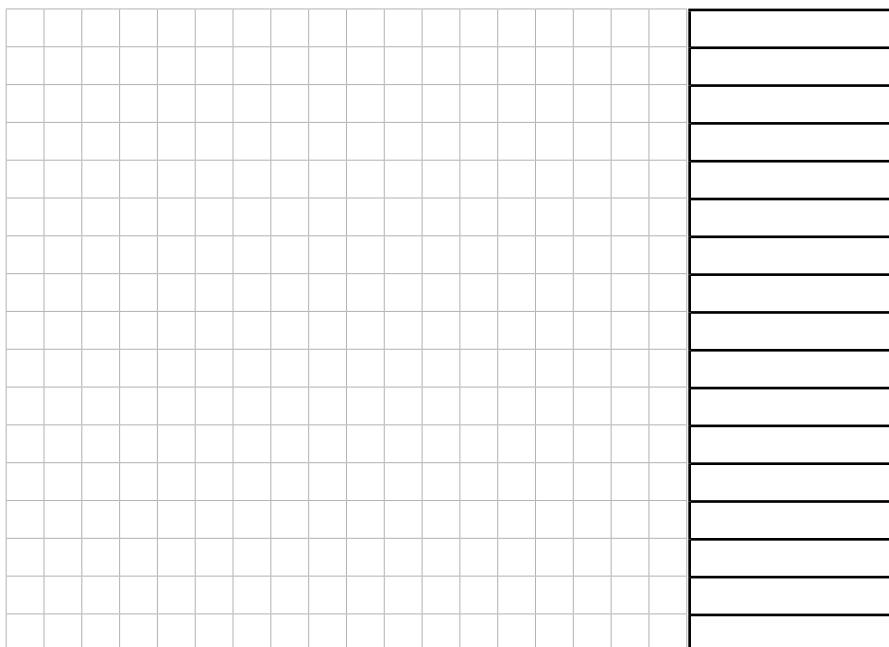
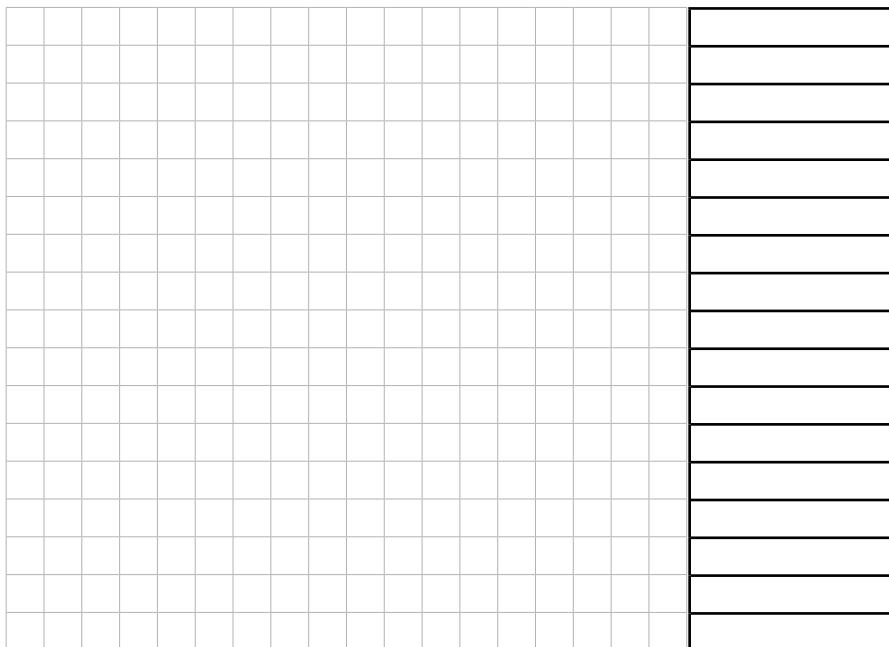
**Tip:** Een fout is zo gemaakt dus doe het met potlood en houd een gum in de buurt!



## **Werkblad: Maak je eigen afbeelding**

---

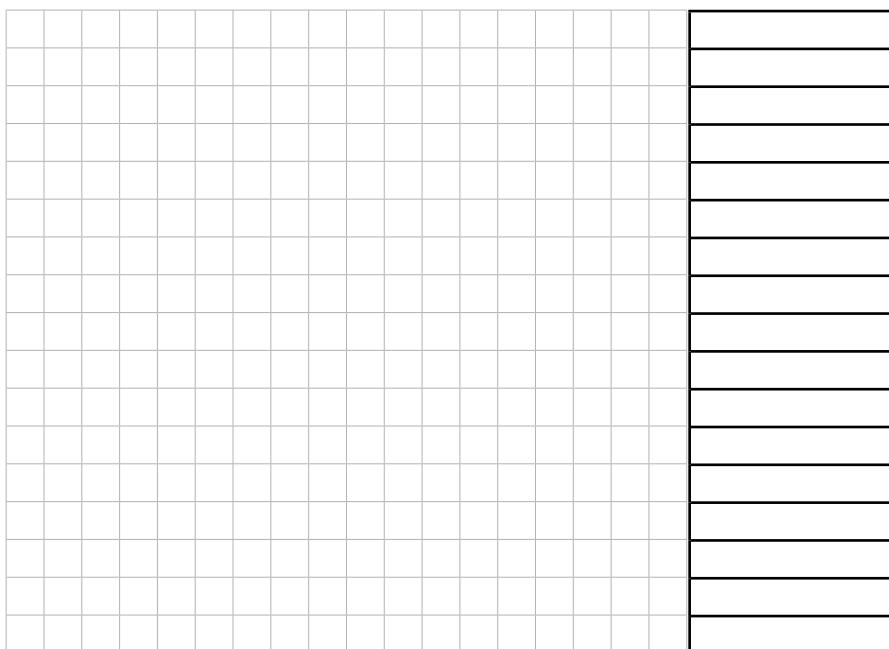
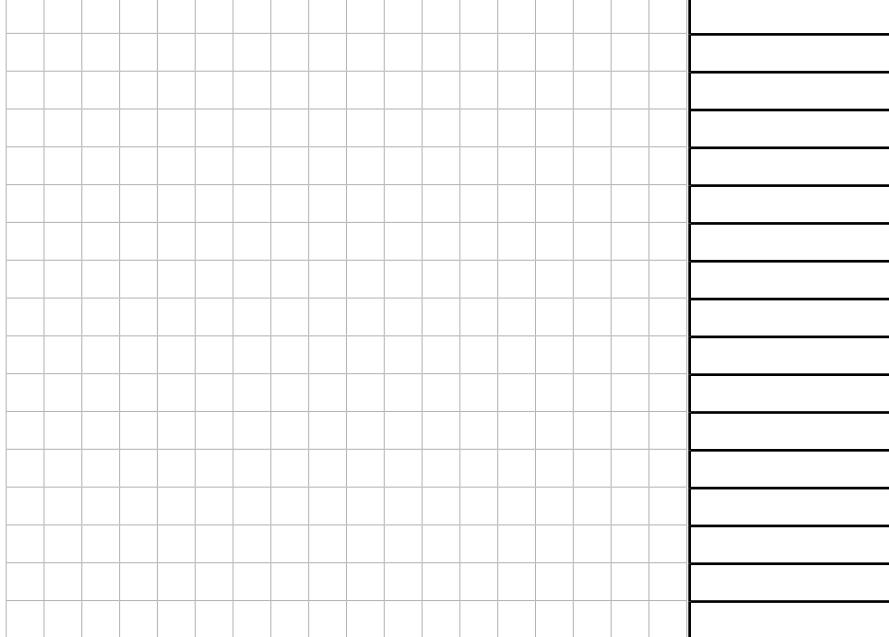
Nu je weet hoe getallen afbeeldingen kunnen weergeven, kan je ook voor een vriend een gecodeerd plaatje tekenen. Teken je afbeelding op het bovenste raster en wanneer je klaar bent, schrijf je de code in de zijlijn van het onderste raster. Knip het vel doormidden en geef de gecodeerde afbeelding aan je vriend om te kleuren. (Je hoeft niet het hele raster te gebruiken, je mag een aantal regels leeg laten als je deze niet nodig hebt.)



## **Werkblad: Maak je eigen afbeelding 2**

---

**Extra voor experts:** Als je een kleurenafbeelding wilt maken, kan je verschillende getallen verschillende kleuren laten voor stellen (bijvoorbeeld 0 is zwart, 1 is rood, 2 is groen enzovoort). Je gebruikt nu twee getallen om pixels weer te geven. De eerste om de lengte van het aantal gekleurde pixels weer te geven zoals de opdrachten hiervoor, en het tweede getal geeft de kleur weer. Probeer nu een kleurenafbeelding voor een vriend te maken. Maar vergeet niet er bij te vertellen welk getal welke kleur is.



# **Variaties en uitbreidingen**

---

1. Teken op een overtrekpapier bovenop het raster, nu kan het resultaat zonder raster gezien worden en wordt de afbeelding duidelijker.
2. In plaats van het raster in te kleuren kunnen kinderen ook post-its gebruiken, of zelfs objecten plaatsen op een groter raster. Of neem krijt en stoeptegels op het schoolplein.

## **Punt van discussie**

Het aantal pixels dat in een keer gecodeerd kan worden is beperkt omdat deze door een binair getal worden weergegeven. Hoe kan je een rij van twaalf zwarte pixels weergeven als je alleen maar nummers tot zeven kunt gebruiken? (Een goede manier is om zeven zwarte pixels weer te geven, gevolgd door nul witte en dan nog vijf zwarte pixels).

## **Waar gaat dit eigenlijk over?**

---

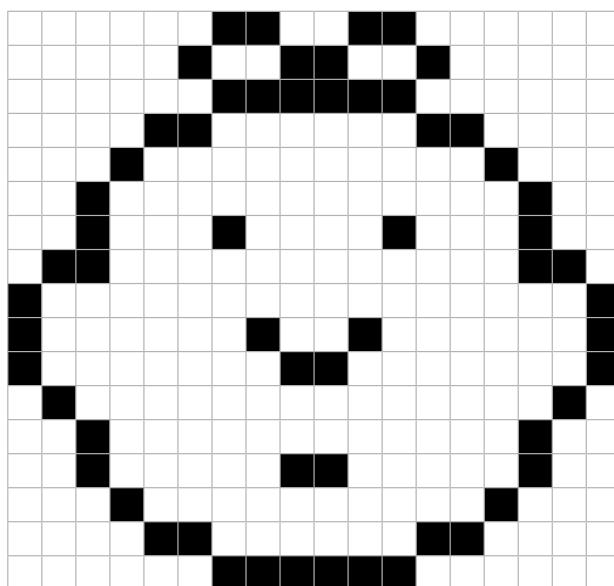
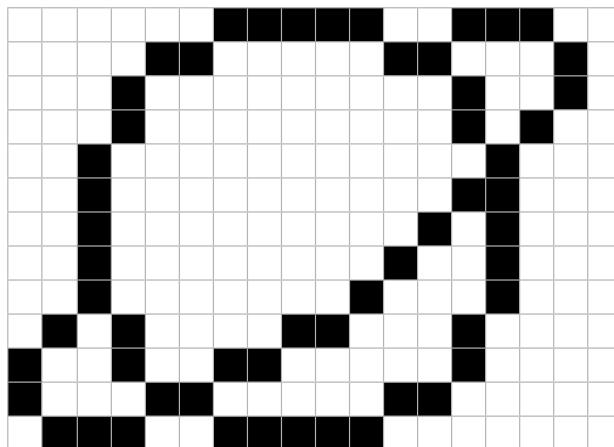
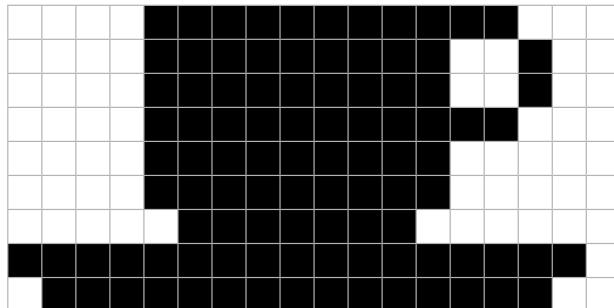
Afbeeldingen kunnen op je computer veel ruimte innemen en al snel te groot worden om te mailen. Gelukkig hebben veel foto's veel 'herhaling' in zich, een blauwe lucht of witte sneeuw met allemaal pixels met dezelfde kleur. Om de benodigde opslagruimte voor een foto te beperken kunnen programmeurs verschillende compressie technieken toepassen. De methode die we in deze opdracht gebruiken, heet 'run-length encoding' en is een efficiënte manier om afbeeldingen of filmpjes mee te verkleinen. Als we afbeeldingen of filmpjes niet zouden verkleinen zouden ze erg veel ruimte op je harde schijf innemen en zouden ze veel moeilijker via internet verspreid kunnen worden.



## **Oplossingen en hints**

---

### **Antwoorden voor het werkblad Pixels tekenen**



# Activiteit 3

---

## Zeg het opnieuw! —Tekst compressie

### Samenvatting

Omdat computers maar een beperkte ruimte hebben om informatie op te slaan, moeten we deze informatie zo efficient mogelijk beschrijven. Dit heet compressie. Door de data te coderen voor het opslaan en weer te decoderen wanneer het weer wordt opgehaald kan de computer meer data opslaan en het sneller over het internet versturen.

### Kerndoelen

- Taal: het herkennen van patronen in woorden en tekst.
- Techniek: Technische kennis en begrip. Hoe werken computers.

### Vaardigheden

- Kopieren van geschreven tekst

### Leeftijd

- 9 jaar en ouder

### Materialen

- *Voorbeeld voor in de klas: Zeg het opnieuw!*

Iedere leerling heeft nodig:

- *Werkblad: Zeg het opnieuw!*

Optionele uitbreidingsopdrachten:

- *Werkblad: Extra voor xperts*
- *Werkblad: Kort en krachtig*

## **Zeg het opnieuw!**

---

### **Introductie**

Computers moeten veel data opslaan en doorsturen en wel zo dat de data niet te veel ruimte innemen of dat het heel lang duurt om de informatie door te sturen. Ze verkleinen de tekst op ongeveer deze manier.

# **Spetter pieter**



**Spetter pieter**

### **Demonstratie en discussie**

Laat het hele refrein van *Spetter pieter pater* zien in de klas (zie ‘Voorbeeld voor in de klas, pagina 29). Zoek patronen in de letters van dit gedicht. Vind je groepen van 2 of meer letters die herhaald worden of misschien zelfs hele woorden of zinsdelen? (Vervang deze met rechthoeken net zoals hierboven.)

**Voorbeeld voor in de klas: Zeg het opnieuw!**

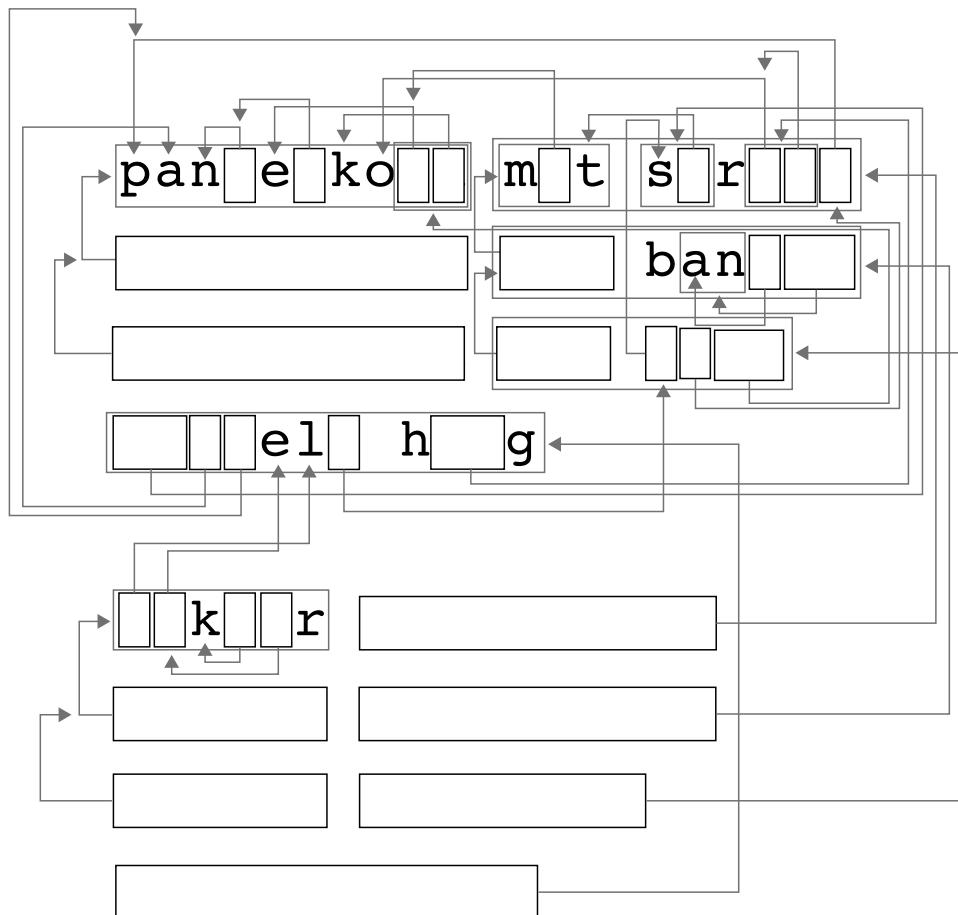
---

**Spetter pieter pater  
lekker in het water  
ga maar vast naar huis  
ik kom een druppel later**

## Werkblad: Zeg het opnieuw!

---

Veel woorden en letters missen in dit gedicht. Kan je de missende letters en woorden weer vinden? Je vindt deze in de rechthoeken waar met de pijlen naar verwezen wordt.



Kies nu een simpel gedicht of kinderversje en maak je eigen puzzel. Zorg ervoor dat de pijlen altijd naar een eerder stuk in de tekst verwijzen. Je puzzel moet ontcijferd kunnen worden door van links naar rechts te gaan en van boven naar beneden, net zoals we lezen.

**Uitdaging:** Probeer zo min mogelijk van de oorspronkelijke tekst te behouden!

Een aantal suggesties: *Slaap kindje slap*, *Lang zal ze leven*, *Drie kleine kleutertjes* — of probeer een stuk uit *Kat met de hoed*!

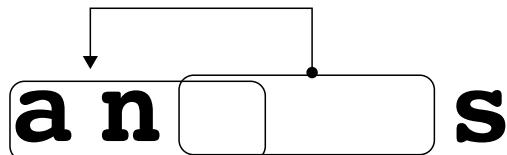
**Tip:** Om een wirwar van pijlen te vermijden: Schrijf ruim, laat veel ruimte over tussen de woorden en regels als je ze opschrijft, zodat je ruimte hebt voor rechthoeken in andere rechthoeken en pijlen die naar de rechthoeken verwijzen.

Het is ook makkelijker om je puzzel te maken door eerst het hele gedicht op te schrijven en daarna pas te beslissen waar de rechthoeken moeten komen.

## Werkblad: Extra voor Experts

---

Hoe zou je deze puzzel oplossen?



Missende tekstdelen kunnen ook naar zichzelf wijzen. In dit geval kan de het woord prima ontcijferd worden als de letters van links naar rechts worden gekopieerd. Iedere letter is bruikbaar voor de kopie waarvoor hij nodig is. Dit is bruikbaar voor computers als er veel bepaalde letters achter elkaar voor komen of sprake is van een patroon van letters.

Bij computers worden de rechthoeken en pijlen weer gegeven door getallen. Bijvoorbeeld, **ananas** kan geschreven worden als **an(2,3)s**. “2” betekent tel twee letters terug om het beginpunt voor het kopieren te vinden,

**an\_\_s**

en “3” betekent kopieer de 3 volgende letters:

**ana\_\_s**

**anan\_s**

**ananas**



Omdat er twee getallen zijn gebruikt om dit woord te coderen zijn normaal gesproken alleen groepen van twee of meer letters de moeite waard om te verkleinen, anders wordt er geen ruimte bespaard. Sterker nog: de grote van het bestand zou zelfs kunnen groeien als er twee getallen worden gebruikt om een letter te coderen.

Verzin nu wat woorden of een zin voor jezelf en codeer ze zoals een computer zou doen als ze verkleind zouden worden. Kan een klasgenoot ze ontcijferen?

## **Werkblad: Kort en krachtig**

---

Hoeveel woorden heb je hier echt nodig?

Stel je voor dat je een computer bent en zo veel mogelijk op je harde schijf wilt opslaan. Streep alle groepen van 2 of meer letters door die al een keer gebruikt zijn. Deze zijn niet meer nodig omdat ze vervangen kunnen worden door een verwijzing. Probeer zo veel mogelijk letters door te strepen.

*Ik ken een dame die een vogel doorslikte,  
je schrikt hé, een vogel doorslikte.*

*Ze slikte de vogel om de spin op te vangen  
die kriebelend friemelend in haar slokdarm bleef  
hangen.*

*Ze slikte de spin, om de vlieg weg te jagen  
En waarom de vlieg?*

*Ik hoor het je vragen  
Dat weet alleen zij, tot het einde der dagen.*

## Waar gaat dit eigenlijk over?

---

De opslagcapaciteit van computers neemt ongelooflijk snel toe, in de laatste 25 jaar is de opslagruimte van een ‘gewone’ computer ongeveer een miljoen keer groter geworden. Maar we vinden ook steeds meer dingen die we op onze computer op willen slaan. Hele boeken, zelfs hele bibliotheken, muziekcollecties en films, maar die kosten allemaal wel opslagruimte. En dat is niet het enige probleem; we willen het ook allemaal delen en grote bestanden zijn een probleem op het internet, dan duurt het te lang om te downloaden. En we maken computers ook steeds kleiner, zelfs een telefoon of een horloge moet tegenwoordig veel op kunnen slaan.

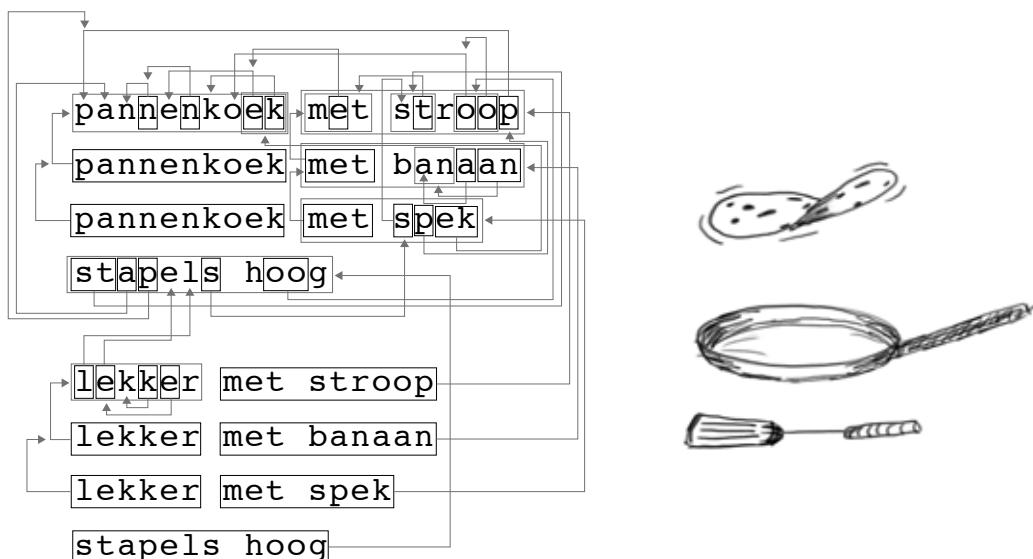
Gelukkig is er een oplossing voor dit probleem. In plaats van alleen maar meer opslagruimte bij te kopen of een snellere internetverbinding te nemen, kunnen we data verkleinen door een compressie toe te passen zodat de data minder ruimte inneemt. Dit proces van verkleinen en weer uitpakken van data gebeurt gewoonlijk automatisch door de computer. Wij merken alleen dat de harde schijf meer op kan slaan en websites sneller zichtbaar zijn, maar de computer doet al het harde werk.

Er zijn veel methodes van compressie uitgevonden. De methode in deze opdrachten, met het verwijzen naar eerder gebruikte stukken tekst, heet ‘Ziv-Lempel encoding’ of ‘LZ encoding’, uitgevonden door 2 Israeliëse professoren in de jaren zeventig. Het kan in iedere taal gebruikt worden en kan makkelijk de data tot de helft verkleinen. Je kent het waarschijnlijk wel onder de naam ‘zip’ bestanden en het wordt ook gebruikt voor ‘GIF’ afbeeldingen.

Andere methodes van tekstcompressie zijn gebaseerd op het idee dat letters die vaak gebruikt worden kortere codes moeten hebben dan andere, net als de Morsecode.

## Oplossingen en hints

---



# Activiteit 4

---

## Kaarten truc—Fout opsporen & herstellen

### Samenvatting

Wanneer data worden opgeslagen op een harde schijf of worden verzonden van de ene computer naar de andere, nemen we aan dat de data niet veranderen tijdens dit proces. Maar soms gaat het fout en worden de data per ongeluk gewijzigd. De activiteit die we nu gaan doen, gebruikt een magische truc om te laten zien hoe je erachter kan komen dat data fout (corrupt) zijn en hoe je die fout weer kan herstellen.

### Kerndoelen

- Rekenen: Getallen groep 5 en hoger. Rekenen en schattingen ontdekken.
- Algebra groep 5 en hoger. Patronen en relaties ontdekken.

### Vaardigheden

- Tellen
- Herkennen van even en oneven getallen

### Leeftijd

- 9 jaar en ouder

### Benodigd materiaal

- Een set van 36 “koelkastmagneten”, maar aan een kant gekleurd.
- Een metalen bord (een whiteboard werkt goed) voor de uitleg.
- Ieder tweetal heeft nodig: 36 identieke kaarten, aan een kant gekleurd.

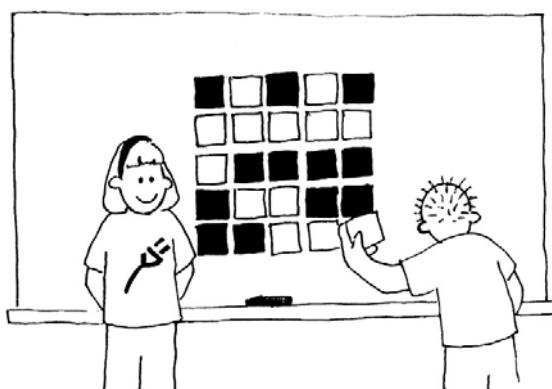
# De “Magische truc”

---

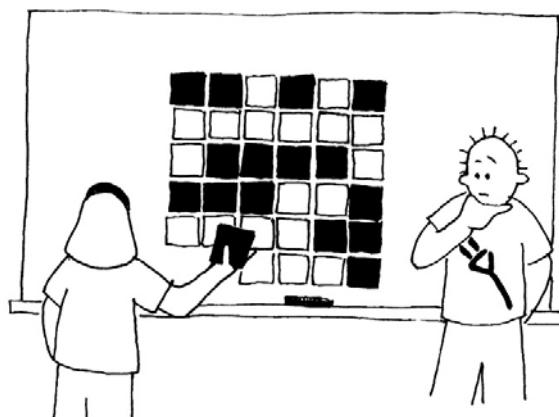
## Demonstratie

Hier is je kans om een goochelaar te worden! Je hebt een stapel identieke kaarten met twee verschillend gekleurde zijden. Voor de demonstratie is het het makkelijkst om platte magnetische kaarten te hebben die verschillend gekleurd zijn aan beide kanten, koelkastmagneten zijn ideaal.

1. Vraag een leerling om de kaarten in een raster van 5 bij 5 te plaatsen met een willekeurige kant boven.



2. Voeg heel ontspannen vervolgens nog een rij en kolom toe “om het iets moeilijker te maken”. Deze kaarten zijn de sleutel tot succes. Je moet de extra kaarten zo plaatsen dat je in iedere rij en in iedere kolom een even aantal gekleurde kaarten hebt.



3. Vraag de leerling om 1 kaart om te draaien terwijl jij niet kijkt. De rij en kolom die de gewijzigde kaart hebben zullen nu een oneven aantal gekleurde kaarten hebben en de gewijzigde kaart onthullen. Kunnen de leerlingen raden hoe de truc in elkaar steekt?

### **Leer de truc aan de leerlingen:**

1. Werk in tweetallen; de leerlingen leggen hun kaarten in een 5 bij 5 raster.
2. Hoeveel gekleurde kaarten zijn er in iedere rij en kolom? Is dit even of oneven? Denk eraan, 0 is ook een even getal.
3. Voeg nu een zesde kaart toe aan iedere rij en zorg er voor dat het aantal gekleurde kaarten even is. Deze extra kaart heet de “controlebit”.
4. Voeg nu een zesde rij toe aan de onderkant en zorg er voor dat het aantal gekleurde kaarten in iedere kolom ook even is.
5. Draai er nu een om. Wat merk je nu op aan deze rij en kolom? (Ze hebben een oneven aantal gekleurde kaarten). Controle kaarten worden gebruikt om uit te vinden of er een fout is gemaakt en waar deze zit.
6. Doe nu deze truc om de beurt.

### **Uitbreidingsactiviteiten:**

Probeer het eens met andere objecten. Alles wat zich in twee ‘staten’ kan bevinden is mogelijk. Gebruik bijvoorbeeld speelkaarten of munten.

Wat gebeurd er als er twee of meer kaarten worden gedraaid? (Het is niet altijd te zeggen welke kaarten zijn gedraaid, maar vaak wel mogelijk om te zien dat er iets veranderd is. Je kan de verandering meestal terugbrengen naar een of twee paar kaarten. Met 4 draaiingen is het mogelijk dat de controle kaarten toch goed lijken achteraf en blijft de fout onopgemerkt).

Een andere interessante oefening is de volgende: neem de onderste kaart aan de rechterkant. Als je deze goed legt voor de kolom erboven ligt deze dan ook goed voor de rij links? (Het antwoord is ja, altijd.)

In deze kaartenoefening hebben we even aantal controlekaarten en een even aantal gekleurde kaarten gebruikt. Kunnen we het ook doen met een oneven aantal controlekaarten? (Dit is mogelijk, maar de onderste kaart aan de rechterkant kan alleen goed geplaatst worden als het aantal rijen en kolommen zelf allebei even of allebei oneven zijn. Dus een 5 bij 9 raster werkt of een 4 bij 4, maar een 3 bij 4 raster zal niet gaan.)

## Een echt voorbeeld voor de experts!

---

Hetzelfde controlesysteem wordt gebruikt bij boekcodes. Meestal kun je achterop boeken een 10-cijferige code vinden. Het 10e cijfer is een controlecijfer, vergelijkbaar met de controlebit in de opdracht.

Als je een boek bestelt en daarbij het ISBN (Internationaal Standaard Boek Nummer) gebruikt, dan kan de uitgever controleren of je geen fout hebt gemaakt. Ze hoeven daarvoor alleen maar naar het controlecijfer te kijken. Zo zit je nooit op het verkeerde boek te wachten!

Je kunt het controlecijfer als volgt uitrekenen:

Vermenigvuldig het eerste cijfer met tien, het tweede met negen, het derde met acht, en zo verder, tot en met het negende cijfer vermenigvuldigd met twee. Tel daarna alle uitkomsten bij elkaar op. Het ISBN 0-13-911991-4 geeft bijvoorbeeld:

$$(0 \times 10) + (1 \times 9) + (3 \times 8) + (9 \times 7) + (1 \times 6) + (1 \times 5) + (9 \times 4) + (9 \times 3) + (1 \times 2) = 172$$

Daarna deel je de uitkomst door elf. Wat blijft er over?

$$172 \div 11 = 15 \text{ waarbij } 7 \text{ overblijft}$$

Als er 0 overblijft, dan is het controlecijfer ook 0. Als een ander getal overblijft, dan krijg je het controlecijfer als je het overgebleven getal aftrekt van 11.

$$11 - 7 = 4$$

Kijk maar terug. Is dit het laatste cijfer van het ISBN? **Ja!**

Als het laatste cijfer van het ISBN geen vier is, dan weten we dat er fout is gemaakt.

Het is mogelijk dat het controlecijfer 10 is. Dan zijn er twee cijfers nodig. In dit geval wordt de letter x gebruikt.



*Een streeptjescode (UPC) op een doos Weet-Bix™*

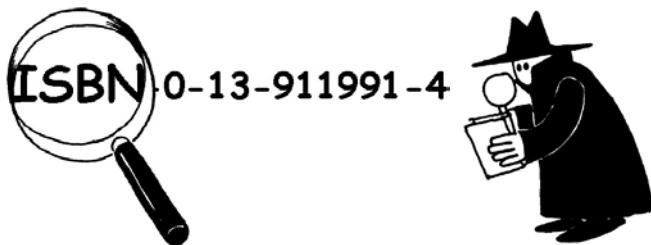
Een ander voorbeeld van een controlecijfer is de streeptjescode op boodschappen. Daarbij wordt een andere formule gebruikt. Als de streeptjescode verkeerd wordt gelezen, dan is het laatste cijfer anders dan het berekende getal. In dat geval maakt de scanner een geluid en kan de kassamedewerker de code opnieuw scannen.

# **Werkblad voor experts: Controleer het boek!**

---

## **Detective Blockbuster**

### **Boek Traceer Service, Inc.**



Wij, van het Bureau Blockbuster, vinden en controleren ISBN controlecijfers tegen een kleine vergoeding.

Help ons bureau - zoek in je klaslokaal of in de bibliotheek naar echte ISBN-nummers.  
Kloppen de controlecijfers?

Soms worden er fouten gemaakt. Veel gemaakte fouten zijn:

- de waarde van een cijfer is veranderd;
- twee cijfers naast elkaar zijn verwisseld;
- er staat een extra cijfer in de reeks
- een cijfer is verwijderd uit de reeks

Kun je een boek vinden met de letter X in plaats van controlecijfer 10? Dat is vast niet heel moeilijk - van elke elf boeken zou één boek dit moeten hebben.

Welk soort fouten wordt niet ontdekt? Kun je een cijfer veranderen en toch het juiste controlecijfer krijgen? Wat gebeurt er als twee cijfers omgekeerd zijn (een gebruikelijke typfout)?

## **Waar gaat dit eigenlijk over?**

---

Stel je voor: je stort € 10 op je bankrekening. De baliemedewerker typt het bedrag in en stuurt het naar een centrale computer. Maar wat als er iets misgaat en de code voor € 10 verandert in € 1.000? Dat is niet erg als je de klant bent, maar het is natuurlijk wel erg voor de bank!

Het is belangrijk om fouten op te sporen in verzonden data. Daarom moet een ontvangende computer de data die binnenkomt kunnen controleren op fouten door elektrische verstoring onderweg. Als er een fout ontdekt wordt, kunnen de data soms opnieuw verstuurd worden. In andere gevallen is dat niet mogelijk, bijvoorbeeld als een disk of band is beschadigd door blootstelling aan magnetische of elektrische straling, door warmte of door fysieke schade. Het wachten wordt wel heel vervelend lang als de data door een fout opnieuw verzonden moeten worden vanaf een ruimtesonde. (Het duurt meer dan een half uur om een radiosignaal te versturen van Jupiter naar de aarde.)

Het is dus belangrijk om fouten in data te kunnen herkennen (foutdetectie) en de originele data te kunnen herstellen (foutcorrectie).

Bij computers wordt dezelfde techniek als bij het spel “kaarten omkeren” gebruikt. Door de bits in denkbeeldige rijen en kolommen te zetten, en controlebits aan elke rij en kolom toe te voegen, kunnen we niet alleen ontdekken dat er een fout is opgetreden, maar ook waar. Het beschadigde deel is hersteld en daarmee is een foutcorrectie uitgevoerd.

Natuurlijk hebben computers vaak veel complexere foutdetectiesystemen die meerdere fouten kunnen ontdekken en herstellen. Een groot deel van de harde schijf in een computer is bedoeld voor het herstellen van fouten zodat de computer betrouwbaar werkt, ook als delen van de disk niet goed werken. De systemen die hiervoor gebruikt worden lijken heel sterk op het controleschema.

## **Oplossingen en hints**

---

Fouten die soms niet worden ontdekt zijn fouten waarbij één cijfer omhoog gaat en een ander cijfer omlaag gaat. De som kan dan nog steeds hetzelfde zijn.

# Activiteit 5

---

## Twintig keer raden — Informatie theorie

### Samenvatting

Hoeveel informatie zit er in een boek van 1000 pagina's? Zit er meer informatie in een telefoonboek van 1000 bladzijden, of in een stapel van 1000 lege vellen papier of in de boeken van Harry Potter? Als we dit kunnen meten dan kunnen we een inschatting maken van hoeveel ruimte nodig is om de informatie op te slaan. Kan je bijvoorbeeld de volgende zin lezen?

*n dz zn mssn d klnkrs.*

Waarschijnlijk wel, want er zit niet veel ‘informatie’ in de klinkers. Deze activiteit laat een manier zien om de hoeveelheid informatie te meten.

### Kerndoelen

- Rekenen: groep 5 en hoger. Getallen ontdekken, groter dan, kleiner dan en verspreiding.
- Algebra groep 5 en hoger. Patronen en reeksen.
- Taal

### Vaardigheden

- Getallen vergelijken en het werken met reeksen.
- Deductie
- Vragen stellen

### Leeftijd

- 10 jaar en ouder.

### Materialen

- Voor de eerste activiteit is geen materiaal nodig.
- Er is een uitbreidingsactiviteit waar de leerling het volgende voor nodig heeft: *Werkblad: Beslisbomen*

# Twintig keer raden

---

## Discussie

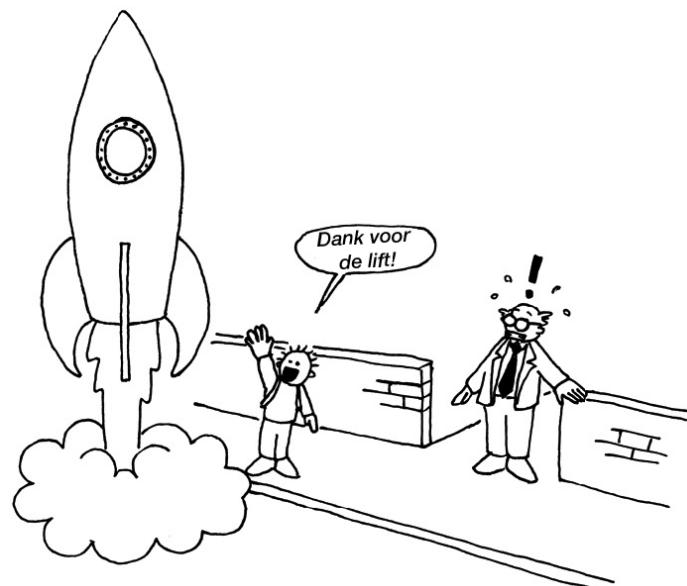
1. Bespreek met de leerlingen wat zij denken dat informatie is.
2. Hoe kunnen we meten hoeveel informatie er in een boek staat? Is het aantal pagina's of het aantal woorden belangrijk? Kan het ene boek meer informatie bevatten dan het andere? Wat als het een heel saai of stom boek is of juist een heel interessant of heel leuk boek? Zou een boek met bijvoorbeeld 400 pagina's vol met "bla, bla, bla, bla" meer of minder informatie bevatten dan bijvoorbeeld een telefoonboek?

Leg uit dat computerdeskundigen de informatie meten door te kijken hoe verrassend een bericht (of boek!) is. Als iemand je iets vertelt dat je al weet - bijvoorbeeld wanneer een vriend die altijd naar school loopt tegen je zegt dat hij "naar school is gelopen" - dan geeft hij je weinig informatie, omdat dat niet verrassend nieuws is. Als hij je zou vertellen dat hij vandaag met de helicopter naar school is gekomen dan is dat heel verrassend en bevat dat bericht dus veel informatie.

Hoe kan de verrassingswaarde van een bericht worden gemeten?

Een manier om de verrassingswaarde te meten is, is na te gaan hoe moeilijk het is om de informatie te raden. Als je vriend je vraagt: "Raad eens hoe ik naar school ben gekomen vandaag," en hij heeft gelopen, dan raad je dat waarschijnlijk in een keer. Het zal waarschijnlijk een paar keer raden kosten voor je bij een helicopter zou zijn, laat staan een ruimteschip.

De hoeveelheid informatie die een bericht bevat wordt dus gemeten door te kijken hoe makkelijk het is om het te raden. Het volgende spelletje geeft ons een idee over hoe het werkt.



## **Twintig vragen activiteit**

---

Dit is een aanpassing van het 20-vragenspel (zie [www.matsoft.nl/20vragen](http://www.matsoft.nl/20vragen)). Leerlingen mogen vragen stellen aan een ander en deze mag alleen met ja en nee antwoorden tot het antwoord is geraden. Iedere vraag mag gesteld worden, maar het antwoord kan dus alleen ja of nee zijn.

Bijvoorbeeld, de vraag: “Ik denk aan”:

- een getal tussen 1 en 100
- een getal tussen 1 en 1000
- een getal tussen 1 en 1.000.000.
- elk heel getal
- een reeks van 6 getallen in een patroon (aanpassen aan het niveau van de groep). Raad in de volgorde van de eerste tot de laatste (bijvoorbeeld 2,4,6,8,10).

Tel het aantal vragen dat gesteld wordt om het antwoord te achterhalen. Hiermee kan je de hoeveelheid “informatie” meten.

### **Nabespreking**

Welke strategie heb je gebruikt? Welke zijn het beste?

Leg uit dat het maar zeven keer raden kost om een getal tussen één en honderd te raden als je de verzameling mogelijke uitkomsten iedere keer halveert. Bijvoorbeeld:

1. Is het minder dan 50? Ja.
2. Is het minder dan 25? Nee.
3. Is het minder dan 37? Nee.
4. Is het minder dan 43? Ja.
5. Is het minder dan 40? Nee.
6. Is het minder dan 41? Nee.
7. Dan moet het 42 zijn! JA!

Bijzonder is dat als de verzameling mogelijke uitkomsten tot duizend, in plaats van honderd, is dit niet betekent dat je tien keer zo vaak moet raden, maar dat er maar drie vragen meer nodig zijn. Iedere keer als het domein van de getallen verdubbelt heb je maar één vraag meer nodig om het antwoord te vinden.

## **Uitbreiding: Hoe veel informatie zit er in een bericht?**

Computerdeskundigen raden niet (alleen) met getallen - ze kunnen ook raden welke letter waarschijnlijk de volgende zal zijn in een woord of een zin.

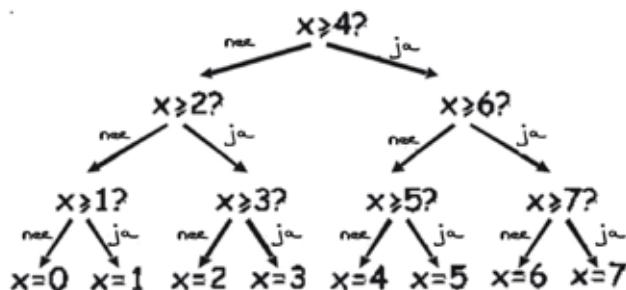
Probeer het raadspel te spelen met een korte zin van 4 tot 6 woorden. De letters moeten geraden worden in de juiste volgorde, van de eerste naar de laatste. Laat iemand alle letters noteren die geraden zijn en houd bij hoeveel keer raden het kost om iedere letter te vinden. Alle vragen die beantwoord kunnen worden met “ja” of “nee” kunnen gebruikt worden. Dus de vragen zouden kunnen zijn: “Is het een T?”, “Is het een klinker?”, “Komt het voor de M in het alfabet?” Een spatie telt ook als letter en moet dus ook geraden worden. Raad om de beurt en kijk welke delen van het bericht het makkelijkst te raden zijn.

# Werkblad: Beslisboom

---

Wanneer de strategie om zo snel mogelijk bij het antwoord te komen bekend is, is het zelfs niet nodig om de vragen te stellen. De persoon die het getal in zijn hoofd heeft, weet welke vragen er komen en hoeft alleen het antwoord te geven zonder de vraag te horen. Een beslisboom maakt dit duidelijk.

Hier is een ‘beslisboom’ om een getal te raden tussen 0 en 7:



Welk pad van ja/nee beslissingen volg je om het getal 5 te ‘raden’?

Hoe veel ja/nee beslissingen heb je nodig om tot elk willekeurig getal te komen?

Nu gebeurt er iets merkwaardigs en fascinerends. Onder de getallen (0,1,2,3...) in de laatste rij van de boom schrijf je het getal nog een keer maar nu in binaire code (zie Activiteit 1, ook bij binair staat het kleinste getal rechts).

Kijk nu nog eens goed naar de boom. Als nee = 0 en ja = 1, wat zie je dan?

In het getalraadspel hiervoor probeerde we de vragen zo te kiezen dat de volgorde van antwoorden precies hetzelfde was als in deze beslisboom.

Maak nu je eigen beslisboom om een getal tussen 0 en 15 te raden.

## Extra voor experts:

Wat voor soort boom zou je gebruiken om iemands leeftijd te raden? En wat voor boom om de volgende letter te raden in een zin?

## Waar gaat dit eigenlijk over?

---

Een gevierrede Amerikaanse wiskundige (daarnaast jongleur en eenwieler) genaamd Claude Shannon deed veel experimenten met dit spel. Hij mat de hoeveelheid informatie in bits, iedere ja/nee vraag is op te vatten als 1/0 bit. Hij vond dat de hoeveelheid informatie die in een bericht staat, afhangt van wat je al weet. Soms kunnen we zulke vragen stellen dat een heleboel andere vragen overbodig worden. In dit geval is de hoeveelheid informatie in een bericht laag. Bijvoorbeeld de hoeveelheid informatie bij het tossen van een munt is normaal gesproken maar één bit: kop of munt. Maar als de munt niet zuiver is en het negen van de tien keer kop wordt, is de hoeveelheid informatie niet één bit maar, geloof het of niet, minder dan één bit!

Hoe kom je erachter wat de uitkomst is van een toss met minder dan een simpele ja/nee vraag? Heel simpel - stel vragen als “zijn de volgende twee tossen beide kop?” Voor een reeks van tossen met de oneerlijke munt zal het antwoord in 80% van de gevallen “ja” zijn. In de 20% van de keren waarin het antwoord “nee” is hoef je maar twee extra vragen te stellen. Maar gemiddeld vraag je minder dan één vraag per toss!

Shannon noemde de informatiedichtheid van een bericht “entropie”. Entropie hangt niet alleen af van het aantal mogelijke uitkomsten - in het geval van een munt opgooien, 2 - maar ook om de kans dat het gebeurt. Bij ongewone gebeurtenissen, of verrassende informatie hebben we meer vragen nodig om achter het bericht te komen omdat ze ons informatie vertellen die we niet al wisten - denk aan het naar school gaan met een helicopter.

De entropie van een bericht is erg belangrijk voor computerdeskundigen. De ruimte om een bericht op te slaan kun je niet kleiner maken dan zijn entropie en de beste compressieprogramma's zijn een equivalent van het raadspel. Omdat een computerprogramma de ‘vragen’ stelt, kan de lijst met gestelde vragen altijd gereproduceerd worden, dus als de antwoorden (bits) maar opgeslagen worden, kunnen we de informatie altijd weer herconstructueren! De beste compressieprogramma's kunnen tekstbestanden tot een kwart van hun originele grootte terugbrengen en dat scheelt een hoop opslagruimte!

Deze raadmethode kan ook gebruikt worden om een computerprogramma te maken dat raadt wat de gebruiker wil gaan typen. Bijvoorbeeld de bekende zoeksuggesties van zoekprogramma's of autocorrect op telefoons.



## **Oplossingen en hints**

---

Het antwoord op een enkele ja/nee vraag komt exact overeen met 1 bit informatie - of het nou een simpele vraag als “Is het meer dan 50?” of een meer complexe als “Is het tussen de 20 en 60?”

Als de vragen op een bepaalde manier worden gesteld in het getalraadspel is de reeks van antwoorden precies de binaire representatie van dat getal. Drie is 011 in binair en dat wordt in de beslisboom weergegeven met “Nee, ja, ja” en dat is hetzelfde als we voor nee een nul opschrijven en een één voor ja.

Een boom voor iemands leeftijd kan rekening houden met het feit dat oudere leeftijden minder vaak voorkomen. Een vraag als “is de leeftijd lager dan 25” geeft waarschijnlijk meer informatie dan “is de leeftijd lager dan 50?”

De beslisboom voor letters in een zin hangt waarschijnlijk af van de voorafgaande letter.

# **Deel II**

---

## **Computers aan het werk**

**algoritmes**

---

## Computers aan het werk

---

Computers werken door een rij van opdrachten uit te voeren die voor ze zijn gemaakt. Deze serie opdrachten stellen hen in staat om informatie te sorteren, te vinden en te versturen. Om dit zo snel mogelijk te kunnen doen, moet je een computer zo goed mogelijk de opdracht geven om informatie te verzamelen uit grote hoeveelheden data.

Een *algoritme* is zo'n serie van opdrachten om een taak uit te voeren. Het idee van een algoritme is de basis van de computerwetenschap. Algoritmes zijn de manier waarop we computers problemen laten oplossen. Sommige algoritmes zijn sneller dan andere en veel algoritmes die zijn bedacht, maken het mogelijk om problemen op te lossen die vroeger heel veel tijd kosten of zelfs niet oplosbaar waren - bijvoorbeeld om Pi te berekenen tot een miljoen cijfers achter de komma, of om alle pagina's te vinden op het internet waar je naam op staat, of om de beste manier te vinden om pakketjes in te pakken in een zeecontainer of je gebruikt een algoritme om uit te zoeken of enorme getallen wel of geen priemgetal zijn.

Het woord ‘algoritme’ komt van de naam van Mohammed ibn Musa Al-Khowarizmi - de naam betekent Mohammed, zoon van Moses, uit Khowarizm - hij was verbonden aan het Huis der Wijsheid in Bagdad rond het jaar 800. Een academie voor onderwijs en onderzoek. Hij bracht de rekenkunsten uit India naar de Arabische wereld en vervolgens naar Europa, waar ze in het jaar 1120 werden vertaald in het Latijn. De eerste woorden uit het schrift waren “Dixit Algorismi” - “Zo zei Algorismi”.

# Activiteit 6

---

## Slagschepen—Zoekalgoritme

### Samenvatting

Computers zijn vaak nodig om informatie te vinden in grote hoeveelheden data. Ze moeten een snelle en efficiënte manier ontwikkelen om dit te doen. Deze activiteit behandelt 3 manieren om te zoeken: lineair zoeken, binair zoeken en groeperen.

### Kerndoelen

- Rekenen: Groep 5 en hoger. Getallen ontdekken: groter dan, kleiner dan of gelijk aan.
- Geometrie: Groep 5 en hoger. Vorm en ruimte, coördinaten.

### Vaardigheden

- Logisch redeneren

### Leeftijd

- 9 jaar en ouder

### Materiaal

- Ieder tweetal heeft nodig: Een kopie van de slagschipspelletjes:
  1. *1A, 1B* voor spel 1
  2. *2A, 2B* voor spel 2
  3. *3A, 3B* voor spel 3

Je zou ook een aantal kopietjes nodig kunnen hebben van de extra spel pagina's: *1A extra, 1B extra, 2A extra, 2B extra, 3A extra en 3B extra*.

# **Slagschepen**

---

## **Introductieactiviteit met de hele klas**

1. Vraag ongeveer vijftien leerlingen om in een rij voor de klas te gaan staan. Geef ieder kind een kaart met een getal erop (in willekeurige volgorde). Houd de getallen bedekt voor de rest van de klas.
2. Geef een andere leerling vier of vijf snoepjes of zoiets. Hun taak is het vinden van een bepaald getal. Ze kunnen ‘betalen’ om een kaart te zien. Als ze het juiste getal hebben gevonden voor ze al hun snoepjes hebben gebruikt, mogen ze de rest houden.
3. Herhaal dit zo vaak als je wilt.
4. Schud nu de kaarten en deel ze opnieuw uit. Maar deze keer moeten de leerlingen voor de klas in oplopende volgorde van getallen gaan staan. Het zoekproces wordt nu herhaald.

Als de getallen nu gesorteerd zijn, is een slimme strategie om een ‘betaling’ te gebruiken om de helft van de rij af te laten vallen door de middelste leerling te vragen zijn of haar kaart te laten zien. Door deze strategie te herhalen moet het juiste getal gevonden kunnen worden door maar 3 snoepjes te gebruiken. Het zal duidelijk zijn dat dit veel efficiënter is dan de ‘willekeurige’ zoekmethode hiervoor.

## **Activiteit**

De leerlingen leren de verschillende manieren van zoeken door de verschillende slagschipspelletjes te spelen. Als ze het spel spelen, laat je ze nadenken over de strategieën die ze gebruiken om het juiste schip te vinden.

# **Slagschepen 1 — Een lineair zoekspel**

---

## **Lees de volgende instructie voor aan de leerlingen**

1. Doe dit spel in tweetallen. Een van jullie neemt blad 1A, de ander 1B. Laat je blaadje niet aan de ander zien!
2. Omcirkel allebei een slagschip in de bovenste rij op je blad en vertel je tegenstander het getal dat in het schip staat geschreven.
3. Probeer nu om de beurt te raden waar het schip van je tegenstander ligt. (Je zegt een letter van een schip en je tegenstander vertelt vervolgens je het getal van dat schip).
4. Hoeveel schoten heb je nodig om het schip van je tegenstander te vinden? Dit is je score.

(De bladen 1A extra en 2B extra zijn extra voor leerlingen die het nog een keer willen spelen of als leerlingen per ongeluk al het blad van de tegenpartij zien. De bladen 2A en B en 3A en B zijn voor volgende spellen).

## **Nabespreking**

Wat zijn de scores?

Wat zijn de minimum en maximum scores die behaald kunnen worden? (Dat zijn 1 en 26, als leerlingen niet twee keer op hetzelfde schip schieten).

Deze methode heet de lineaire zoekstrategie (of zoekgoritme), omdat deze strategie langs alle posities gaat, één voor één.

## **Slagschepen 2 — Een binair zoekspel**

---

### **Instructies**

De instructies bij deze versie van het spel zijn hetzelfde als die bij het vorige spel, maar de getallen op het schip staan nu in **oplopende volgorde**. Leg dit uit aan de leerlingen voordat ze beginnen.

1. Doe dit in tweetallen. Een van jullie heeft blad 2A, de ander blad 2B. Laat je blaadje niet aan de ander zien!
2. Jullie kiezen nu beiden een schip op de bovenste rij. Omcirkel dit schip en vertel je tegenstander het getal dat in het schip staat.
3. Raad nu om de beurt de plek van het schip van je tegenstander. (Je zegt een letter van een schip en je tegenstander vertelt je welk getal daar staat).
4. Hoeveel schoten heb je nodig om het schip van je tegenstander te vinden? Dit is je score.

### **Nabespreking**

Wat zijn de scores?

Welke strategie hebben de leerlingen die laag hebben gescoord gebruikt?

Welk schip heb je als eerste uitgekozen om te beschieten? (Het schip in het midden vertelt je in welke helft je schip moet zitten).

Welke lokatie koos je hierna? (Opnieuw is de beste strategie weer het middelste schip te kiezen uit de overgebleven helft).

Als je deze strategie volgt hoeveel schoten heb je dan maximaal nodig? (vijf)

Deze methode heet de ‘binaire zoekstrategie’, omdat het probleem in twee delen gesplitst wordt, hoger of lager.

# **Slagschepen 3 — Een zoekspel door middel van groeperen**

---

## **Instructies**

1. Doe dit in tweetallen. Een van jullie heeft blad 3A, de ander blad 3B. Laat je blaadje niet aan de ander zien!
2. Jullie kiezen nu beiden een schip op de bovenste helft. Omcirkel dit schip en vertel je tegenstander het getal dat in het schip staat.
3. Nu je het getal van het schip van je tegenstander weet, weet je meteen in welke kolom (0 tot 9) deze staat. Dat werkt zo: je telt de cijfers van het getal van het schip bij elkaar op. En het laatste getal van het antwoord is het nummer van de kolom waar het schip zich in bevindt. Bijvoorbeeld, om het schip met nummer 2345 te vinden, doe je  $2+3+4+5$  is 14. Het laatste cijfer van 14 is 4, dus het schip moet in kolom 4 zitten. Nu je weet in welke kolom het schip zit, kan je beginnen met raden. Deze zoekstrategie heet “groeperen” (“hashing” in het engels), omdat de cijfers worden gegroepeerd.
4. Speel nu het spel met deze zoekstrategie. Je kan dit spel vaker spelen met hetzelfde blad - kies gewoon schepen uit verschillende kolommen.

(Let op, anders dan bij de vorige zoekspellen moeten de extra bladen 3Aextra en 3Bextra echt samen gebruikt worden, omdat het patroon van schepen in de kolommen moet overeenkomen).

## **Nabespreking**

Wat zijn de scores?

Welke schepen zijn snel te vinden? (Degene die alleen in een kolom staan).

Welke zijn moeilijker te vinden? (Degene wiens kolom veel schepen bevat).

Welke van de 3 zoekmethode is het snelst en waarom?

Wat zijn de voordelen van de 3 verschillende zoekstrategieën? (De tweede strategie is sneller dan de eerste, maar daarbij hoeven de schepen niet in volgorde te liggen. De derde strategie is vaak sneller dan de eerste twee, maar het is mogelijk dat deze door toeval juist weer heel traag is. In het slechtste geval, als alle schepen in dezelfde kolom zitten doordat de som van cijfers van het getal van de schepen toevallig allemaal hetzelfde eindcijfer hebben, is deze methode net zo langzaam als de eerste strategie).

## **Uitbreidingsactiviteiten**

---

Laat de leerlingen hun eigen spelletjes maken met deze drie zoekstrategieën. Voor het tweede spel moeten ze de getallen in oplopende volgorde zetten. Vraag hen hoe ze het derde, groepeerzoekspel, heel moeilijk kunnen maken (het moeilijkst is als alle schepen in dezelfde kolom zitten). En hoe kunnen ze het juist heel makkelijk maken? (probeer hetzelfde aantal schepen in iedere kolom te krijgen, door de juiste getallen te kiezen). Wat zou er gebeuren als iemand je voor de gek houdt en een getal zegt waar geen schip van bestaat? (In het Lineaire spel duurt het 26 schoten om hierachter te komen. In het Binaire spel 5 schoten en bij het Groepeerspel hangt het af van het aantal schepen dat in de betreffende kolom zit).

Als je gebruik maakt van de binaire zoekstrategie, hoeveel schoten heb je dan nodig als er honderd schepen zouden zijn (rond de zes), en bij duizend schepen (rond de negen), en bij een miljoen (rond de negentien)? (Valt het op dat het aantal schoten dat nodig is heel langzaam op loopt vergeleken met het aantal schepen? Er is een extra schot nodig voor iedere verdubbeling van schepen).

## Mijn Scheepen

totaal aantal schoten:

A	B	C	D	E	F	G	H	I	J	K	L	M
9058	7169	3214	5891	4917	2767	4715	674	8088	1790	8949	13	3014
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

## JOUW Scheepen

totaal aantal schoten:

A	B	C	D	E	F	G	H	I	J	K	L	M
8311	7621	3542	9264	450	8562	4191	4932	9462	8423	5063	6221	2244
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1A

## Mijn schepen

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1630	9263	4127	405	4429	7113	3176	4015	7976	88	3465	1571	8625

totaal aantal schoten:

## JOUW schepen

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1630	9263	4127	405	4429	7113	3176	4015	7976	88	3465	1571	8625

totaal aantal schoten:

1 B

## Mijn schepen

totaal aantal schoten:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
163	445	622	1410	1704	2169	2680	2713	2734	3972	4208	4871	5031	5283	5704	6025	6801	7440	7542	7956	8094	8672	9137	9224	9508	9663

Jouw schepen

totaal aantal schoten:

A B C D E F G H I J K L M  
N O P Q R S T U V W X Y Z

2A

## Mijn schepen

totaal aantal schoten:

A	B	C	D	E	F	G	H	I	J	K	L	M
33	183	730	911	1927	1943	2200	2215	3451	3519	4055	5548	5655
5785	5897	5905	6118	6296	6625	6771	6831	7151	7806	8077	9024	9328
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

## JOUW schepen

totaal aantal schoten:

A	B	C	D	E	F	G	H	I	J	K	L	M
33	183	730	911	1927	1943	2200	2215	3451	3519	4055	5548	5655
5785	5897	5905	6118	6296	6625	6771	6831	7151	7806	8077	9024	9328
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

2B

# Mijn schepen

totaal aantal schoten:

0	1	2	3	4	5	6	7	8	9
A  9047	C  3080	E  5125	H  8051	I  1481	L  7116	O  6000	R  9891	S  1989	W  1062
B  1829	D  9994	F  1480	G  8212	J  4712	K  6422	M  8944	P  7432	T  2050	X  2106
					N  4128	Q  4110	U  8199	V  4392	Y  5842
								Z  7057	

# Jouw schepen

totaal aantal schoten:

0	1	2	3	4	5	6	7	8	9
A	E	H	I	L	M	N	R	S	W
B	F	G		K			T	U	X
C				J			V	Y	Z
D									

# 3A

# Mijn schepen

totaal aantal schoten:

0	A E B C D	1	E F G H I	2	H I J K L	3	I J K L M	4	J K L M N	5	K L M N O	6	L M N O P	7	M N O P R	8	N O P R S	9	O P R S T	Y W X U	Z V X U
	A 9308 B 1478 C 8417 D 9434		E 6519 F 2469 G 5105		H 1524 I 8112 J 2000		K 4135 L 9050 M 1265 N 5711		O 4200 P 7153 Q 6028		R 3121 S 9503 T 1114 U 7019		V 2385 W 5832 X 1917 U		Y 1990 Z 2502						

Jouw schepen

totaal aantal schoten:

Jouw scheepen		totaal aantal schoten:									
0	A	1	B	2	C	3	D	4	E	5	F
1											
2											
3											
4	H	I	J	K	L	M	N	O	P	Q	R
5											
6	O	P	Q	U	S	T	V				
7	R										
8											
9	W	X	Y	Z							

۳۳

## Mijn schepen

totaal aantal schoten:

A	B	C	D	E	F	G	H	I	J	K	L	M
6123	1519	9024	5164	2038	2142	7156	9974	9375	7104	1004	1023	5108
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1884	3541	5251	4840	3654	3289	2480	5602	8965	4053	2405	2304	1959

## JOUW schepen

totaal aantal schoten:

A	B	C	D	E	F	G	H	I	J	K	L	M
6123	1519	9024	5164	2038	2142	7156	9974	9375	7104	1004	1023	5108
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1884	3541	5251	4840	3654	3289	2480	5602	8965	4053	2405	2304	1959

# 1 A extra

## Mijn schepen

totaal aantal schoten:

2387	9003	3951	5695	1284	4761	7118	1196	1741	3791	3405	3132	6682
A	B	C	D	E	F	G	H	I	J	K	L	M
9493	9864	7359	1250	7036	2916	7562	9299	8910	6713	5173	8617	4222
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

## JOUW schepen

totaal aantal schoten:

1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6
A	B	C	D	E	F	G	H	I	J	K	L	M
1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1 B extra

## Mijn schepen

A	B	C	D	E	F	G	H	I	J	K	L	M
28	326	943	1321	1896	2346	2430	2929	3106	3417	4128	4717	4915
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

totaal aantal schoten:

## JOUW schepen

A	B	C	D	E	F	G	H	I	J	K	L	M
5123	5615	6100	7015	7120	7695	7812	8103	8719	9020	9608	9713	9911
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

totaal aantal schoten:

2A extra

## Mijn scheepen

totaal aantal schoten:

56	194	306	1024	1510	1807	2500	2812	3011	3902	4178	5902	5915
A	B	C	D	E	F	G	H	I	J	K	L	M
6102	6526	6818	7020	7155	7913	8016	8230	8599	8902	9090	9526	9812
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

## JOUW scheepen

totaal aantal schoten:

56	194	306	1024	1510	1807	2500	2812	3011	3902	4178	5902	5915
A	B	C	D	E	F	G	H	I	J	K	L	M
56	194	306	1024	1510	1807	2500	2812	3011	3902	4178	5902	5915
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

2B extra

## Mijn schepen

totaal aantal schoten:

0	1	2	3	4	5	6	7	8	9
A \ 1982	C \ 6113	E \ 9121	H \ 5009	I \ 2651	L \ 1248	O \ 2004	R \ 9369	S \ 1321	W \ 9172
B \ 7841	D \ 1055	F \ 1011	G \ 2984	J \ 1751	M \ 1716	P \ 5173	T \ 3004	U \ 7190	X \ 2052
				K \ 4848	N \ 2148	Q \ 2806	V \ 3285	Y \ 6012	Z \ 7525

## Jouw schepen

totaal aantal schoten:

0	1	2	3	4	5	6	7	8	9
A	E	H	L	M	R	S	T	V	Y
B	F	I	J	K	O	P	Q	W	Z
C	G							X	
D									

# 3A extra

## Mijn schepen

totaal aantal schoten:

0	1	2	3	4	5	6	7	8	9
A  8615	E  1361	H  7726	K  3000	L  1814	O  9656	P  4002	R  6993	V  8208	Y  2917
B  7003	F  7644	I  9003	J  5557	M  2002	N  8844	S  3121	T  4300	W  9423	Z  4122
C  1991	G  5600					U  1907			
D  6211									

## Jouw schepen

totaal aantal schoten:

0	1	2	3	4	5	6	7	8	9
A	C	E	H	I	L	O	R	S	W
B	D	F	G	J	K	M	N	P	X
						V		U	Y
								Z	

# 3B extra

## Waar gaat dit eigenlijk over?

---

Computers slaan een hoop informatie op, maar ze moeten hier ook weer snel doorheen kunnen om het juiste te vinden. Een van de grootste zoekproblemen in de wereld wordt aan internetzoekmachines gevraagd, zij moeten miljarden webpagina's doorzoeken in een fractie van een seconde. De data die aan de computer zijn gevraagd om te gaan zoeken, zoals een woord, een barcode of de naam van een auteur, heet een "zoeksleutel".

Computers kunnen heel snel door informatie heen en je zou misschien denken dat ze gewoon aan het begin van de harde schijf of het internet beginnen en dan blijven zoeken tot ze het juiste gevonden hebben.

Dit is wat we deden in het lineaire zoekspel. Maar deze methode heel traag, zelfs voor computers.

Neem bijvoorbeeld een supermarkt die 10.000 verschillende producten in de schappen heeft staan. Wanneer er een barcode wordt gescand bij de kassa moet de computer door 10.000 nummers om het product en de bijbehorende prijs te vinden.

Zelfs als het maar een duizendste van een seconde duurt om iedere code te checken, kost het toch tien seconden om door de hele lijst te gaan. Hoe lang zou je dan met een kar vol bij de kassa staan?



is

Een betere strategie voor het oplossen van het probleem is de binaire zoekmethode. In deze methode worden de getallen op volgorde gezet en door naar het middelste getal te kijken, weet je meteen in welke helft je verder moet zoeken. Dit proces wordt herhaald tot het product gevonden is. Terug naar de supermarkt, de 10.000 producten kunnen nu doorzocht worden met 14 pogingen, wat misschien tweehonderdste van een seconde duurt, daar merk je ook met een volle kar niks van.

De derde strategie om data te vinden heet "groeperen" of "hashing". De informatie wordt gegroepeerd opgeslagen volgens van tevoren bepaalde methode. Bij het zoeken wordt de zoeksleutel aangepast om de groep (dus de locatie) te vinden. Bijvoorbeeld, als de zoeksleutel een telefoonnummer is, kan je alle cijfers van het nummer optellen de uitkomst vervolgens door elf delen en dan neem je het restgetal; de computer hoeft nu nog alleen maar te zoeken op de locatie van dit laatste getal om daar de juiste gegevens te vinden.

Op dit punt is de groepeersleutel een beetje als de controlegetallen uit activiteit 4 - de magische kaartentruc - een klein stukje informatie wordt toegevoegd (een groepsnummer) die bepaalt waar de data opgeslagen worden. Gewoonlijk vindt de computer dan vrijwel direct de informatie waar hij naar zoekt. Er is maar een hele kleine kans dat verschillende sleutels de computer naar dezelfde plek leiden en de computer dan alsnog door een hele hoop informatie heen moet om de juiste informatie te vinden.

Computerdeskundigen gebruiken vaak een of andere variant van de groeperingsstrategie bij het zoeken; alleen als het heel belangrijk is dat de data op volgorde blijven is dit niet handig. Het is ook niet slim om deze methode te gebruiken als een zeldzaam langzame reactie van de computer (in het geval dat er zich heleboel items in de groep bevinden) uit den boze is. Dit kan bij deze methode af en toe gebeuren.

# Activiteit 7

---

## Lichtste en zwaarste — Sorteer algoritmes

### Samenvatting

Computers worden vaak gebruikt om lijsten in een bepaalde volgorde te zetten, bijvoorbeeld namen in alfabetische volgorde, e-mails of afspraken in de volgorde van tijd of items in numerieke volgorde. Lijsten sorteren helpt ons dingen snel te vinden en maakt het makkelijker om extreme waardes te vinden. Als je de cijfers van een schooltoets op volgorde zet, zie je snel het hoogste en laagste behaalde cijfer.

Als je de verkeerde methode gebruikt kan het lang duren om een lange lijst op een volgorde te krijgen, zelfs met een snelle computer. Gelukkig zijn er verschillende snelle methodes uitgevonden voor het sorteren. Met deze activiteit ontdekken leerlingen verschillende sorteestrategieën, en zien ze hoe een slimme strategie de taak veel sneller kan uitvoeren dan een domme strategie.

### Kerndoelen

- Rekenen: Meten groep 4 en hoger. Het leren van simpele weegopdrachten.

### Vaardigheden

- Gebruik van een balans
- Sorteren
- Vergelijken

### Leeftijd

- 8 jaar en ouder

### Materialen

Elk groepje leerlingen heeft nodig:

- Een set van 8 elementen van dezelfde grootte, (maar niet doorzichtig) verschillend van gewicht (bijv. melkpakken of doosjes gevuld met zand of zoets).
- Een balans
- Werkblad: gewichten sorteren*
- Werkblad: Verdeel en heers*

# Lichtste en zwaarste

---

## Discussie

Computers moeten vaak lijsten van dingen op volgorde zetten. Bedenk met elkaar waarbij het ‘op volgorde’ staan belangrijk is. Wat zou er gebeuren als het niet op volgorde zou staan?

Computers kunnen normaal gesproken maar twee waardes tegelijk vergelijken. De activiteit op de volgende pagina gebruikt deze beperking om de leerlingen een idee te geven over hoe dat gaat.

## Activiteit

1. Verdeel de leerlingen in groepjes.
2. Iedere groep heeft zijn eigen gewichten, een balans en een kopie van het werkblad *Gewichten sorteren* nodig.
3. Laat de leerlingen zelf de activiteit doen en bespreek daarna het resultaat.

# Werkblad: Gewichten sorteren

---

**Doel:** Vind de beste manier om groepen gewichten op volgorde te zetten zonder dat je hun gewicht kent.

**Je hebt nodig:** Zand of water, 8 dezelfde doosjes en een balans

## Wat moet je doen:

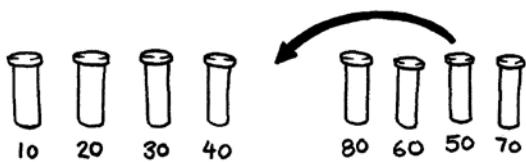
1. Vul ieder doosje met een verschillende hoeveelheid zand of water en doe ze goed dicht.
2. Zet de doosjes zo neer dat ze niet meer op volgorde van gewicht staan en hussel ze nog even door elkaar.
3. Nu moeten we het lichtste doosje vinden. Hoe doe je dat het makkelijkst?

**Let op:** Je mag de balans alleen gebruiken om te zien welke van de twee doosjes het zwaarst is en alleen twee gewichten mogen per keer met elkaar worden vergeleken.

4. Kies drie willekeurige gewichten en sorteert in volgorde van licht naar zwaarst door gebruik te maken van de balans. Hoe doe je dat? Wat is het minimaal aantal benodigde vergelijkingen die je moet maken? En waarom?
5. Sorteer nu alle doosjes van licht naar zwaar.
6. Wanneer je denkt klaar te zijn, controleer je de volgorde door ieder doosje opnieuw te wegen, is de volgende zwaarder dan die daarvoor?

## Selectie sortering

Een methode die de computer zou kunnen gebruiken is selectie sortering. Dit is hoe het werkt: vind eerst het lichtste gewicht in de rij en zet deze aan de kant. Vervolgens zoek je uit de overgebleven doosjes weer het lichtste en zet deze op volgorde naast de vorige



lichtste. En herhaal dit tot alle gewichten zijn verplaatst naar het andere rijtje.

Tel hoeveel vergelijkingen je moet maken.

**Extra voor experts:** Laat zien hoe je kan berekenen hoeveel vergelijkingen je nodig hebt met acht doosjes? En met negen? Of twintig?

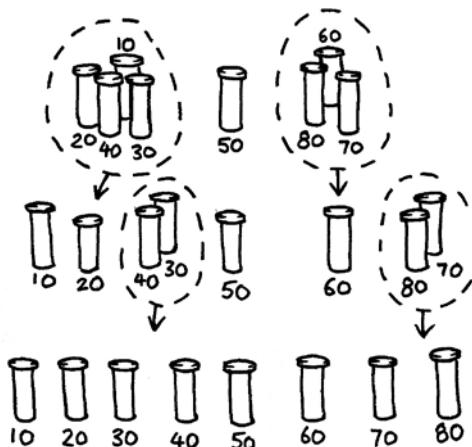
# Werkblad: Verdeel en heers

---

## Snelsorteren

Snelsorteren is een stuk sneller dan selectie sorteren, zeker bij langere lijsten. Sterker nog, dit is een van de beste manieren die we tot nu toe kennen. Zo werkt het:

1. Kies een willekeurig doosje en zet het aan een kant van de balans.
2. Vergelijk nu ieder overgebleven doosje met dit doosje. Als ze lichter zijn zet je ze links, zijn ze zwaarder zet je ze rechts en het gekozen doosje zet je in het midden (het kan dus gebeuren dat je met veel meer doosjes aan de ene kant eindigt dan aan de andere).
3. Kies nu een van de groepen doosjes (rechts of links) en herhaal de vorige opdracht. Doe ook hetzelfde voor de andere groep. Denk er aan dat je het eerste doosje in het midden laat staan.
4. Blijf dit herhalen op de overgebleven groepen tot alle groepen uit maar 1 doosje bestaat. Als alle groepen verdeeld zijn tot de grootte van maar 1 doosje staan de doosjes gesorteerd van licht naar zwaar!



Hoeveel vergelijkingen waren nu nodig?

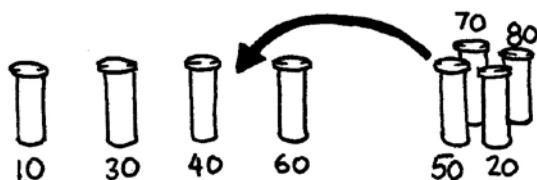
Je ervaring zou nu moeten zijn dat de snelsorteermethode veel sneller is dan de selectie sorteermethode of je moet toevallig de pech hebben gehad dat je begon met de lichtste of zwaarste. Als je het geluk had precies met het middelste gewicht te beginnen zou je het in 14 vergelijkingen gehaald moeten hebben. En bij de selectie sorteermethode had dit 28 vergelijkingen gekost. In elk geval is de snelle sorteermethode even snel als de selectie sorteermethode en in vrijwel alle gevallen een stuk sneller.

**Extra voor experts:** Als de snelle sorteermethode toevallig altijd de lichtste uit zou kiezen, hoeveel vergelijkingen heb je dan nodig?

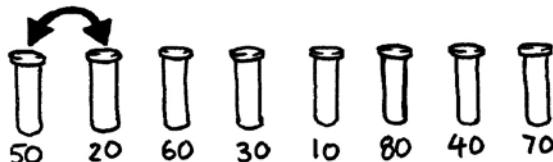
## Variaties en uitbreidingen bij het sorteren

Er zijn veel verschillende methodes bedacht om dingen te sorteren. Je kunt je gewichten proberen te sorteren volgens deze methodes:

**Invoegsorteren** werkt door iedere keer een doosje uit een ongesorteerde groep te pakken en deze in te voegen op de juiste plek in een steeds groter wordende rij (zie afbeelding hieronder). Met iedere keer invoegen wordt de groep ongesorteerde doosjes kleiner en de rij gesorteerde doosjes langer totdat de hele rij is gesorteerd. Bij kaartspelletjes gebruik je vaak deze methode om de kaarten in je hand op volgorde te zetten..

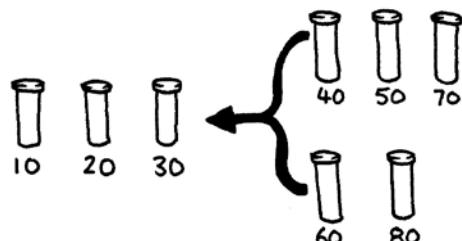


**Bubbel sorteren** (in het Engels “bubble sort”) houdt in dat je iedere keer door de hele rij gaat en de doosjes per tweetal omdraait als deze niet op volgorde staan. De rij is gesorteerd als er geen doosjes meer van plek ruilen. Het zwaarste doosje gaat door dit systeem iedere keer stapsgewijs mee naar achteren (en komt boven drijven als een luchtbol, vandaar de naam van deze methode). Deze methode is niet heel snel, maar hij is makkelijk te begrijpen.



**Samenvoegend sorteren** is een andere methode die ‘verdeelt en heerst’ om een rij te sorteren. Eerst wordt de rij in (ongeveer) twee gelijke rijen verdeeld. Eerst worden beide (halve) rijen gesorteerd en daarna worden ze samengevoegd. Samenvoegen gaat heel makkelijk - je verplaatst de hele tijd de lichtste van de twee rijen naar de een nieuwe rij. In de afbeelding hieronder zijn de 40 en de 60 aan de voorkant van de rij, dus de 40 wordt als eerste in de nieuwe rij geplaatst.

Hoe sorteert je de korte rijen? Gewoon, gebruik de samenvoegend sorteermethode! Als je dit blijft doorvoeren, eindigt je uiteindelijk met rijen van 1 doosje en dan weet je dat je kan stoppen.



## **Waar gaat dit eigenlijk over?**

---

Informatie is veel makkelijker vindbaar in een gesorteerde lijst. Telefoonboeken, woordenboeken, de index van een boek, ze gebruiken allemaal de alfabetische volgorde.

Het leven zou veel ingewikkelder worden als ze dat niet zouden doen. Als een lijst van getallen (bijvoorbeeld uitgaven) in volgorde wordt gezet, zijn de extremen zo veel makkelijker te vinden, omdat ze dan aan het begin of eind van een lijst terecht komen. Ook is dan makkelijk te zien of waardes twee of meer keer voorkomen omdat ze nu naast elkaar staan.

Computers zijn veel tijd kwijt met het sorteren van dingen, dus hebben computerdeskundigen snelle methodes bedacht om dit te doen. Sommige langzamere methodes, zoals invoegsorteren, selectie sorteren of bubbelsorteren kunnen handig zijn in speciale situaties, maar de snelste methodes, zoals snelsorteren worden in vrijwel alle gevallen gebruikt.

Snelsorteren gebruikt een concept dat “recursie” heet. Dat betekent dat je een lijst blijft delen in kleinere stukken en bij ieder deel dezelfde sorteertechniek gebruikt. Deze specifieke benadering heet “verdeel en heers”. De lijst wordt maar in stukken ‘verdeeld’ totdat deze ‘beheerst’ kan worden. Bij het snelsorteren worden de lijsten verdeeld tot ze maar een item bevatten. En het is niet moeilijk om 1 item in volgorde te plaatsen. Dit klinkt misschien erg onpraktisch, maar in de praktijk is dit veel en veel sneller dan alle andere methodes.

## **Oplossingen en hints**

---

De beste manier om het lichtste gewicht te vinden is ieder doosje om de beurt te wegen en steeds bij te houden wat de lichtste is. En dat kan door bij elke beurt twee doosjes te vergelijken en de lichtste te houden. Deze weer met een andere vergelijken en weer de lichtste te houden. En dit herhalen tot je alle doosjes gewogen hebt.

Vergelijk de gewichten op de balans. Dit kan makkelijk gedaan worden met drie vergelijkingen, en soms zullen twee vergelijkingen al genoeg zijn, als de leerlingen begrijpen dat de vergelijkingen transitief zijn (dat betekent, als A lichter is dan B en B is lichter dan C dan moet A dus lichter zijn dan C).

### **Voor de experts:**

Hier is een verkorte uitleg voor de som van het aantal vergelijkingen dat selectie sorteren maakt.

Om de lichtste van twee objecten te vinden heb je één vergelijking nodig, drie objecten hebben twee vergelijkingen nodig, vier, drie, enzovoorts. Om acht objecten te sorteren met selectie sorteren hebben we zeven vergelijkingen nodig om de lichtste te vinden, zes vergelijkingen om de één na lichtste te vinden, vijf voor de volgende enzovoort. Dat leidt tot de volgende som:

$$7 + 6 + 5 + 4 + 3 + 2 + 1 = 28 \text{ vergelijkingen.}$$

n objecten hebben dus  $1 + 2 + 3 + 4 + \dots + n - 1$  vergelijkingen nodig om te sorteren.

Al deze getallen bij elkaar optellen gaat veel makkelijker door er groepen van te maken. Bijvoorbeeld, om de getallen 1 tot en met 20 bij elkaar op te tellen ( $1 + 2 + 3 + \dots + 20$ ), kan je er makkelijk groepjes van maken waar allemaal hetzelfde uitkomt (21 in dit geval) en dat levert 10 groepjes op:

$$\begin{aligned} & (1 + 20) + (2 + 19) + (3 + 18) + (4 + 17) + (5 + 16) + \\ & (6 + 15) + (7 + 14) + (8 + 13) + (9 + 12) + (10 + 11) \\ & = 10 \text{ groepjes van } 21 = 10 \times 21 \\ & = 210 \end{aligned}$$

**Algemener geldt:  $1 + 2 + 3 + 4 \dots + n = n \times (n + 1)/2$**

# Activiteit 8

---

## Snel, sneller, snelst —Sorteer Netwerken

### Samenvatting

Computers zijn supersnel, maar er zijn grenzen aan hoe snel ze sommige problemen kunnen oplossen. Een manier om een probleem sneller op te lossen is om er meerdere computers tegelijk aan te laten werken. Het is dan wel de kunst om de computers zo in te zetten dat ze niet op elkaar hoeven te wachten en allemaal een eigen deel van het probleem oplossen. Bij deze activiteit gebruiken we sorteert netwerken waarmee we door gelijktijdig vergelijken een lijst een stuk sneller kunnen sorteren.

### Kerndoelen

- Rekenen: getallen en ordening. Getalbegrip: groter dan, kleiner dan

### Vaardigheden

- Vergelijken
- Sorteren
- Ontwikkelen van strategie
- Samenwerken

### Leeftijd

- 7 jaar en ouder

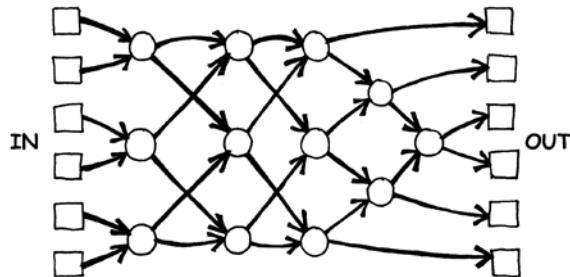
### Materialen

- Deze opdracht doe je buiten op het schoolplein.
- Krijt
- Twee sets met zes kaarten, een set met de getallen 1 t/m 6, een set met willekeurige getallen tussen 100 en 1000. Kopieer het werkblad, plak op karton en knip uit.
- Stopwatch

# Sorteer netwerken

---

Voorafgaand aan de activiteit tekenen we het volgende netwerk op het schoolplein.

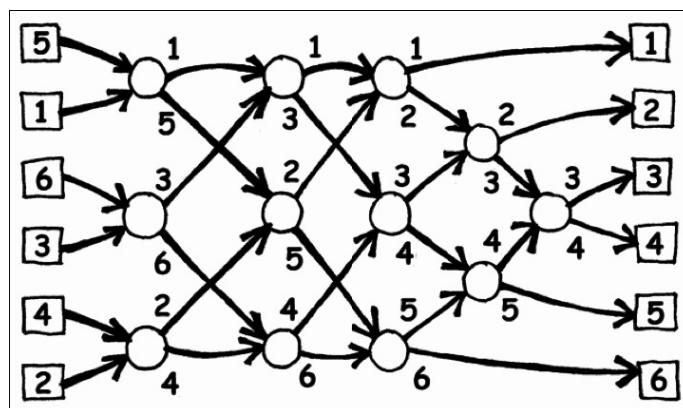


## Instructies voor de leerlingen

Deze activiteit laat je zien hoe computers willekeurige getallen op volgorde zetten door gebruik te maken van een zogenaamd sorteernetwerk.

1. Verdeel de klas in groepjes van zes. Per keer kan maar een groepje het sorteernetwerk gebruiken.
2. Verdeel de zes kaarten binnen je groepje.
3. Ieder groepslid gaat in een vierkant staan met uitgaande pijlen. Jullie getallen zijn nu nog niet op volgorde.
4. Neem een stap volgens de pijl. Kom je in een cirkel dan moet je wachten totdat er nog iemand komt. Kom je in een vierkant dan wacht je daar.
5. Sta je met zijn tweeën in de cirkel dan vergelijk je de getallen op jullie kaart. Degene met het kleinste getal gaat verder over de pijl naar links, degene met het grootste getal neemt de rechter uitgang.
6. Staan jullie op de goede volgorde als jullie in de vierkanten aan de overkant zijn?

Als een groepje niet op volgorde eindigt, is er iets fout gegaan en moeten ze alle zes weer naar een startvierkant. Kijk of de opdracht voor in de cirkels goed begrepen wordt (kleinste naar links, grootste naar rechts), bijvoorbeeld:



## Kopieerblad: Sorteer netwerken

---

1

2

3

4

5

6

---

156

221

289

314

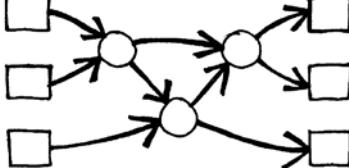
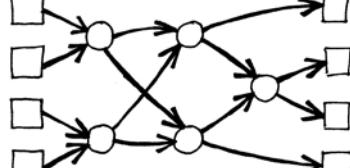
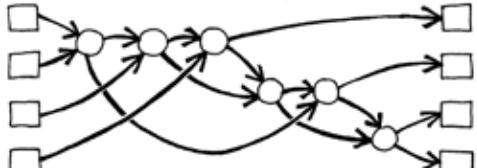
422

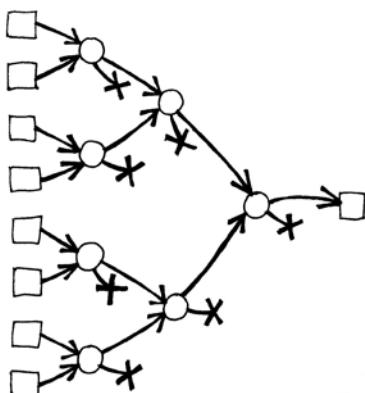
499

## Variaties

1. Als de kinderen snappen hoe het werkt kun je er met de stopwatch een wedstrijd van maken.
2. Gebruik kaarten met grotere getallen (bijvoorbeeld de getallen tussen 100 en 1000 van het kopieerblad).
3. Gebruik kaarten met nog grotere getallen, bijvoorbeeld je mobiele nummer, of gebruik woorden en sorteert die volgens alfabet, of sorteert op lengte of op leeftijd.

## Uitbreidingsactiviteit

1. Wat gebeurt er als de regel veranderd wordt in “kleinste naar rechts, grootste naar links”? (De getallen zullen in omgekeerde volgorde gesorteerd worden.) Werkt het ook als je het netwerk tegen de stroom in gebruikt? (Soms, wel, soms niet, de kinderen moeten in staat zijn een startsituatie te bedenken die niet tot een goed antwoord leidt bij gebruik in de verkeerde richting.)
2. Probeer netwerken te ontwerpen voor grotere of kleinere groepen. Bijvoorbeeld hier is een netwerk voor een groepje van drie. Kunnen de kinderen dat zelf bedenken?
3. Hieronder staan twee netwerken voor een groepje van vier. Welke is sneller? (Dat is het tweede netwerk. In het eerste netwerk moeten alle vergelijkingen na elkaar plaatsvinden, terwijl in het tweede netwerk sommige vergelijkingen onafhankelijk van elkaar gelijktijdig kunnen worden uitgevoerd. In computertermen wordt het eerste netwerk serieel genoemd, het tweede is den dele parallel uitgevoerd.)

4. Probeer een netwerk te maken voor een grotere groep.
5. Je kunt ook netwerken maken die het grootste of kleinste getal bepalen. Hiernaast bijvoorbeeld een netwerk om het minimum (of maximum) te bepalen voor 8 getallen. Na iedere vergelijking in de cirkel gaat de kleinste door en is de ander af.
6. Welke activiteiten uit het dagelijks leven kunnen wel of niet versneld worden door delen gelijktijdig te doen? Maakt het bijvoorbeeld uit of het gasfornuis een of vier pitten heeft bij de bereiding van een maaltijd?

## **Waar gaat dit eigenlijk over?**

---

Omdat we computers steeds meer gebruiken voor steeds meer informatie willen we graag dat zij de informatie zo snel als mogelijk verwerken.

Een manier om de computer het probleem zo snel mogelijk te laten oplossen is het gebruiken van zo min mogelijk stappen (zoals in de activiteiten 6 en 7).

Een andere manier om sneller tot resultaat te komen, bereik je door meerdere computers gelijktijdig in te zetten die tegelijkertijd aan verschillende delen van het probleem werken. Zo had het sorteernetwerk voor zes getallen twaalf cirkels waarin getallen met elkaar werden vergeleken. Omdat er soms wel drie van die vergelijkingen gelijktijdig plaats vonden had het netwerk slechts de tijd van vijf vergelijkingsacties nodig. Hiermee is dit netwerk meer dan tweemaal zo snel als een systeem dat maar één vergelijking tegelijkertijd kan doen.

Niet alle taken kunnen sneller gemaakt worden door het parallel inzetten van computers. Het graven van een sloot van 10 meter lang gaat makkelijker (en sneller) met meer mensen, terwijl het graven van een kuil van 10 meter diep veel minder kan profiteren van vele handen: je kunt niet alvast aan de tweede meter beginnen terwijl de eerste meter nog niet klaar is. Computerdeskundigen onderzoeken tot op de dag van vandaag manieren om problemen handig op te delen zodat er geprofiteerd kan worden van gelijktijdig werkende (parallel) computers.

# Activiteit 9

---

## Modderstad—Minimaal Ospannende Bomen

### Samenvatting

Onze maatschappij is verbonden middels heel veel netwerken: telefoonnet, elektriciteitsnet, de riolering, computernetwerk, en het wegennet. Gewoonlijk is er bij ieder netwerk een keuze waar de wegen, kabels of radioverbindingen worden aangelegd. Het zou fijn zijn als we een methode hebben om zaken efficiënt te verbinden.

### Kerndoelen

- Wiskunde: meetkunde. Ontdekken van vorm en ruimte: vinden van kortste paden op een kaart

### Leeftijd

- 9 en ouder

### Vaardigheden

- Probleem oplossen

### Materialen

Iedere leerling heeft nodig:

- *Werkblad: Modderstad*
- Fiches of stukjes karton (ongeveer 40 per kind)

# Modderstad

---

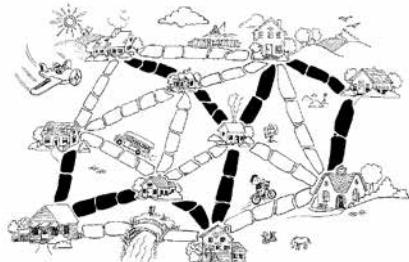
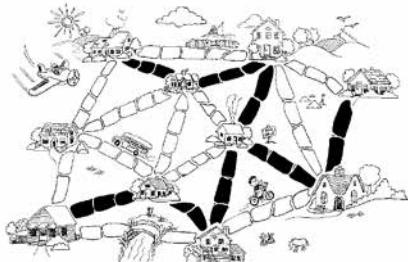
## Introductie

Bij deze activiteit leer je hoe computers worden gebruikt om oplossingen te vinden voor praktische problemen, bijvoorbeeld hoe huizen zo goed mogelijk aan te sluiten op het elektriciteitsnet. Laat de kinderen het werkblad “Modderstad” gebruiken waar het probleem wordt uitgelegd.

## Nabesprekking

Laat de kinderen vertellen over de oplossingen die ze hebben gevonden. Welke strategieën hebben ze gebruikt?

Een goede manier om de beste oplossing te vinden begint met de lege kaart. Vervolgens bedek je de wegen met fiches totdat alle huizen verbonden zijn. Bij de keuze van de weg die je gaat bedekken neem je (een van) de kortste weg(en) waarbij je wegen tussen al verbonden huizen overslaat. Soms zijn er verschillende mogelijkheden en het eindresultaat kan verschillen door de mogelijkheid die je hebt gekozen. Hieronder zie je twee mogelijke oplossingen.



Een andere strategie is om alle paden te bedekken en dan overbodige paden te verwijderen. Dit kan wel een stuk langer duren.

Waar zijn nog meer netwerken in de wereld om je heen?

Het op deze manier weergeven van netwerken met knopen (de huizen in het voorbeeld) en kanten (de wegen) wordt een graaf genoemd. Netwerken die je zo kunt weergeven zijn bijvoorbeeld de wegen tussen steden of het routenetwerk van een luchtvaartmaatschappij.

Naast het algoritme van de “minimaal opspannende boom” (“Minimum spanning tree” in het Engels) hebben computerdeskundigen nog vele andere algoritmes die goed werken indien het probleem als een graaf kan worden beschreven. Voorbeelden zijn het kortste pad tussen twee punten (denk aan routeplanner) of de kortste reis die alle steden aandoet.

## Werkblad: Modderstad

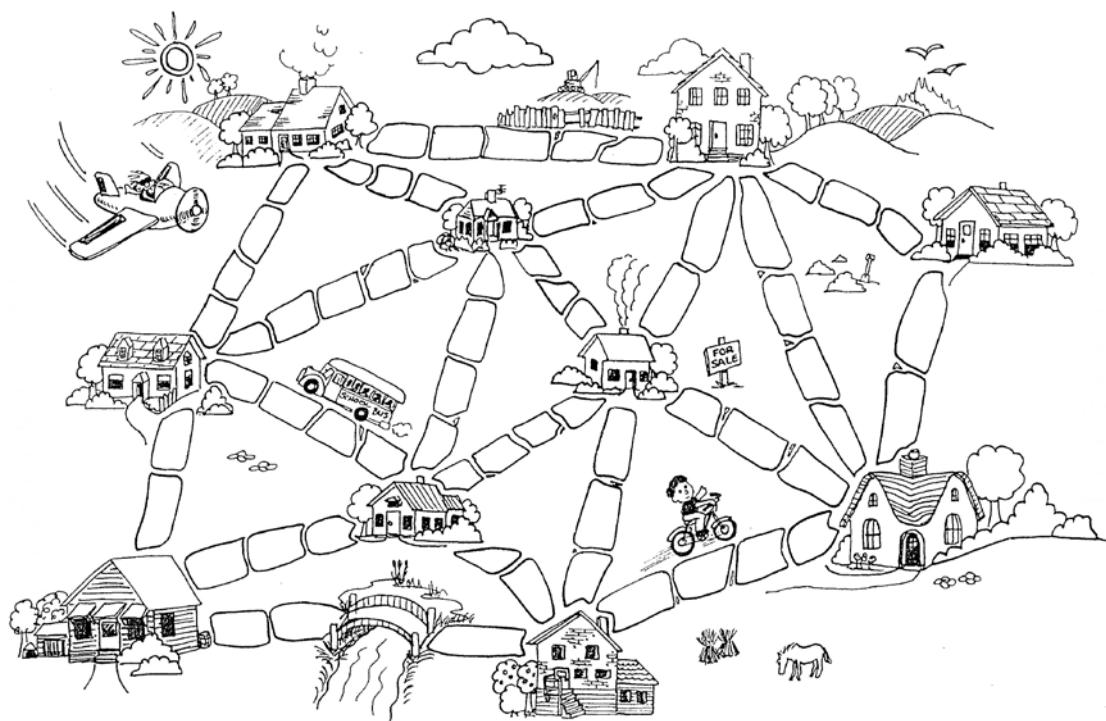
---

Er was eens een stad waar geen wegen waren. Vooral na zware regenstormen was het voor de inwoners heel lastig om bij elkaar op bezoek te gaan omdat de aarde dan in een grote modderpoel veranderde. Auto's bleven vast zitten en de mensen werden er behoorlijk smerig van. De burgemeester besloot dat sommige van de paden verhard moesten worden. Omdat ze geld wilden overhouden voor het aanleggen van een zwembad wilden ze niet teveel geld kwijt zijn. De burgemeester stelde voor dat het plan aan de volgende twee voorwaarden moest:

1. Er moesten voldoende paden verhard worden om alle bewoners de mogelijk te geven elkaar via verharde wegen te bereiken, en
2. Het verharden moest zo min mogelijk kosten.

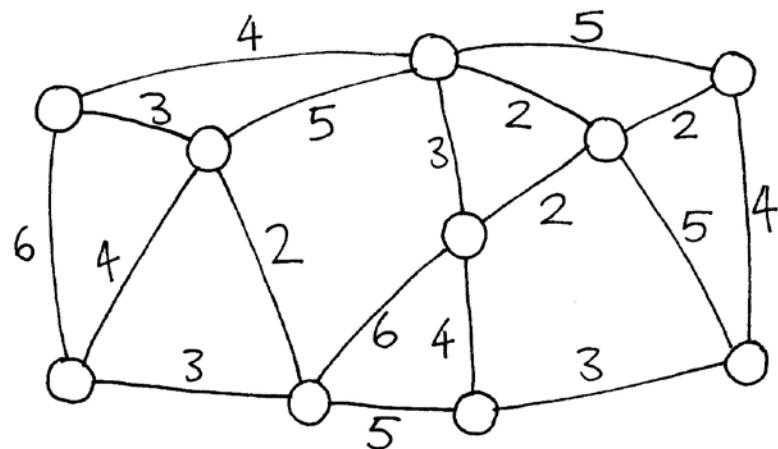
Hier zie je de plattegrond van het dorp. Het aantal tegels tussen de huizen geeft aan hoe duur het is om dat pad te verharden. Om een pad te verharden leg je fiches op de tegels. Maak met zo min mogelijk fiches een netwerk dat alle huizen verbindt.

Welke strategieën heb je gebruikt om dit probleem op te lossen?



## Variaties en uitbreidingen

Hier is een andere manier om de huizen en de paden weer te geven:



De huizen zijn weergegeven met cirkels (knopen), de modderige paden met lijnen (kanten), en de lengte van de paden wordt weergegeven door het getal dat erbij staat.

Informatici en wiskundigen gebruiken dit soort diagrammen vaak om een probleem te beschrijven. Ze noemen het een graaf. Niet te verwarren met die graaf die in Den Haag woont (“en zijn zoon heet Jantje”). Een graaf hoef je niet op schaal te tekenen, het gaat erom welke knopen verbonden zijn en welk gewicht bij zo’n verbinding hoort.

Kun je zelf wat andere modderdorpen tekenen en de goedkoopste manier vinden om verharde wegen aan te leggen? Kijk ook of de oplossing van anderen overeenkomt met die van jou.

Kan je een regel bedenken voor het aantal wegen dat nodig is voor de beste oplossing?

Hangt het af van het aantal huizen dat er is?

## **Waar gaat dit eigenlijk over?**

---

Stel dat je voor een nieuwe stad moet bedenken hoe het water, de elektriciteit en het gas naar alle huizen moet worden gebracht. Ieder huis apart verbinden met de elektriciteitscentrale is nogal omslachtig, veel beter is het om een netwerk aan te leggen dat voor alle huizen een verbinding met de centrale waarborgt. Het maakt daarbij niet echt uit dat sommige huizen een veel langere route naar de elektriciteitscentrale krijgen, zolang ze verbonden zijn is er stroom!

De opdracht om een netwerk te maken waarbij de totale lengte zo min mogelijk is wordt wel het probleem van een minimaal opspannende boom genoemd.

Minimaal opspannende bomen zijn niet alleen nuttig voor netwerken van gas en energie; ze helpen ook bij het oplossen van problemen in computernetwerken, telefoonnetwerken, oliepijpleidingen en routenetwerken van luchtvaartmaatschappijen. Hoewel je bij dat laatste voorbeeld, waarbij je de beste routes ontwerpt voor reizigers, je ook rekening moet houden met de kosten van de individuele reiziger. Niemand wil uren omreizen omdat het iets goedkoper is. Het vinden van de oplossing met het minimale totaal van wegen, zoals we deden voor Modderstad, is in het geval van vliegroutes niet voldoende omdat het geen rekening houdt met de wens bepaalde wegen sowieso op te nemen.

Je kunt minimaal opspannende bomen ook gebruiken bij andere problemen die als graaf kunnen worden gerepresenteerd. Bijvoorbeeld bij het handelsreizigersprobleem waarbij je zoekt naar een rondreis langs alle steden in het netwerk.

Er zijn efficiënte algoritmes (methodes) om in een willekeurige graaf een minimaal opspannende boom te vinden. Een eenvoudige methode is die waarbij je begint met geen enkele weg en vervolgens steeds de kortste nog niet gebruikte verbinding toevoegt die het netwerk laat groeien met een punt dat nog niet verbonden was. Deze methode heet Kruskals algoritme omdat J.B. Kruskal in 1956 er als eerste over publiceerde.

Er zijn nog vele problemen met grafen, waaronder het handelsreizigersprobleem, waar informatici nog altijd op zoek zijn naar snellere methodes die de best mogelijke oplossing vinden.

## **Oplossingen en hints**

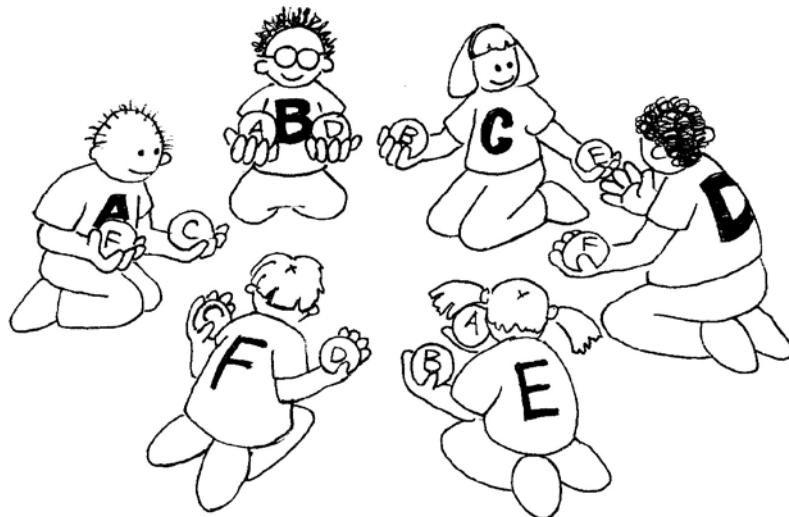
---

Hoeveel wegen zijn er nodig in een stad met  $n$  huizen? In de optimale oplossing zijn dat er altijd  $n-1$ . Omdat je met  $n-1$  wegen  $n$  huizen kunt verbinden. Een extra weg is overbodig omdat het voor een tweetal huizen een dubbele verbinding maakt.

# Activiteit 10

---

## Het sinaasappelspel — Routebepaling en vastlopen in netwerken



### Samenvatting

Als een heleboel mensen tegelijk één ding gebruiken (zoals auto's op de weg, of berichten via het internet), kan het vastlopen. Een goede samenwerking is nodig om dit te voorkomen.

### Kerndoelen

- Rekenen: Ontwikkelen van logica en redeneren

### Vaardigheden

- Samen problemen oplossen
- Logisch redeneren

### Leeftijd

- 9 jaar en ouder

### Materialen

Per leerling heb je nodig:

- Twee sinaasappels of twee tennisballen
- Naamplaatje of sticker

# Het sinaasappelspel

---

## Inleiding

Dit is een spel, waarbij je samen een probleem oplost. Het doel is dat elke persoon eindigt met de sinaasappels met de eigen letter in zijn handen.

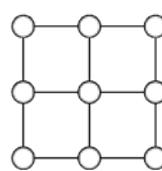
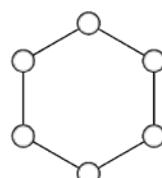
1. Groepen van vijf of meer leerlingen zitten in een kring.
2. De leerlingen krijgen een kaartje of sticker met een letter van het alfabet. Op de sinaasappels schrijf je de letters van elke leerling, per twee sinaasappels staat één letter. Dus er zijn twee sinaasappels met een “a”, twee met een “b”, enzovoorts.
3. Deel de sinaasappels willekeurig uit, maar één sinaasappel hou je bij je. Zo is er tijdens het spel altijd één leerling met één lege hand.(Geen enkele leerling mag in het begin een sinaasappel met zijn letter hebben.)
4. De leerling geven de sinaasappels door totdat elke leerling de twee sinaasappels met de eigen letter in zijn handen heeft. Daarbij gelden twee regels:
  - Je mag maar één sinaasappel per hand hebben, dus een leerling heeft maximaal twee sinaasappels in zijn bezit.
  - Een sinaasappel mag alleen worden doorgegeven naar een lege hand van een directe buurman in de cirkel. (Een leerling mag zelf kiezen welke van de twee sinaasappels hij doorgeeft aan zijn buurman.)

Leerlingen zullen snel merken dat, als ze “egoïstisch” zijn (hun eigen sinaasappels vasthouden zodra ze die krijgen), de groep misschien niet in staat is om het doel te bereiken. Het kan nodig zijn om te benadrukken, dat er in dit spel geen aparte winnaars zijn, maar dat iedereen tegelijk wint als iedereen zijn sinaasappels heeft en de puzzel dus is opgelost .

## Nabespreking

Welke strategieën hebben de leerlingen gebruikt om het probleem op te lossen?

Waar in het echte leven heb je meegemaakt dat iets vastliep? (Enkele voorbeelden kunnen zijn: een file, Koningsdag in Amsterdam, of proberen om een heleboel mensen in een keer door een deuropening te krijgen.)



## Uitbreidingsactiviteiten

Probeer de activiteit met een kleinere of grotere kring.

- Laat de leerlingen nieuwe regels bedenken..
- Doe de activiteit zonder er bij te praten.
- Probeer verschillende opstellingen zoals zitten op een lijn, of met meer dan twee buren voor sommige kinderen. Enkele suggesties staan hiernaast.

## **Waar gaat dit eigenlijk over?**

---

Routebepaling en vastlopen zijn problemen in vele netwerken, zoals het wegennet, telefoon en computer systemen. Ingenieurs besteden veel tijd aan het uitzoeken hoe deze problemen op te lossen en hoe netwerken te ontwerpen die deze problemen makkelijker oplossen.

Routebepaling, congestie en vastlopen kunnen frustrerende problemen in veel verschillende netwerken veroorzaken. Denk maar aan uw favoriete spitsuur! Het is meerdere keren gebeurd in New York dat het verkeer in de straten, zo overbelast raakte dat alles vastliep; niemand kon zijn auto nog verplaatsen! Soms als de computers "down" zijn in bedrijven (zoals banken) wordt het probleem veroorzaakt door een vastlopend communicatienetwerk. Het ontwerpen van netwerken, waardoor de routebepaling eenvoudig en efficiënt wordt en congestie wordt geminimaliseerd, is een moeilijk probleem voor veel soorten ingenieurs.

Soms werken meerdere personen tegelijk met dezelfde gegevens . Als een deel van de gegevens (zoals banksaldo van een klant) wordt bijgewerkt, is het belangrijk om de gegevens "op slot" te zetten tijdens de update. Als deze niet worden vergrendeld, kan iemand anders op hetzelfde moment aan de gegevens werken en dan zou het saldo verkeerd kunnen worden geregistreerd. Indien dit vergrendelingssysteem wordt gestoord door het vergrendelen van een ander, kan de boel vastlopen.

Een van de meest opwindende ontwikkelingen in het ontwerpen van computers is de komst van de parallelle berekening, waar honderden of duizenden PC-achtige processors worden gecombineerd (in een netwerk) om een enkele krachtige computer vormen. Veel problemen zoals het Orange Game moeten continu (maar veel sneller!) worden afgespeeld op deze netwerken, om deze parallelle computers te laten werken.

# **Deel III**

---

## **Computers vertellen wat te doen**

---

weergeven van een actieplan

---

## Computers vertellen wat ze moeten doen

---

Computers volgen opdrachten op, iedere seconde miljoenen opdrachten. Om een computer voor je aan het werk te zetten hoef je alleen maar de juiste opdrachten door te geven. Maar dat is minder makkelijk dan het klinkt!

Als wij een opdracht krijgen gebruiken we gezond verstand om te bedenken wat er bedoeld is. Als iemand bijvoorbeeld tegen je zegt “ga door die deur” bedoelen ze niet dat je dwars door de deur moet, maar dat je door de deuropening moet, na het openen van de deur als deze nog dichtzit. Computers zijn anders, ze kunnen opdrachten alleen letterlijk opvolgen. Wanneer je de computer in een robot bestuurt, moet je goed oppassen of de opdrachten die je geeft niet in een ongewenste toestand kunnen eindigen. Want de robot neemt alles letterlijk en zal proberen dwars door de deur te gaan als je dat opdraagt. Het is even wennen om opdrachten heel letterlijk te nemen.

De twee activiteiten in dit deel geven een idee van hoe het is om te communiceren met een machine die alles letterlijk neemt, en bovendien overweg kan met een beperkte verzameling van opdrachten.

De eerste opdracht leert ons over een machine die computers gebruiken om woorden, getallen en tekenreeksen te herkennen. Die machine heet ook wel een eindigetoestandsautomaat (In het engels: finite-state machine).

De tweede activiteit gaat over hoe we met computers kunnen communiceren. Een goede programmeur kan een computer vertellen wat te doen door een geschikte rij opdrachten te geven. Zo'n lijst opdrachten heet een programma en de computer zal de opdrachten letterlijk proberen uit te voeren. Er zijn heel veel verschillende programmeertalen, ieder met een eigen verzameling van opdrachten die gebruikt kunnen worden. Wij gaan met een vereenvoudigde taal aan de slag zodat we het principe ook zonder computer kunnen begrijpen.



# Activiteit 11

---

## Schatzoeken—Eindige automaat

### Samenvatting

Computerprogramma's moeten vaak rijen tekens verwerken, bijvoorbeeld de letters of woorden in een document, of zelfs een tekst die weer door een ander programma is aangeleverd. Computerdeskundigen gebruiken een zogenaamde "eindigetoestandsautomaat" (ETA) om dat te doen. Zo'n ETA werkt volgens een verzameling instructies die de computer vertelt of een rij tekens goed of fout is. Wij gaan met een zeer bijzondere ETA aan de slag: een schatkaart, die ons gaat vertellen welke routes naar de schat leiden (goed) en welke niet (fout)

### Kerndoelen

- Wiskundig denken, logica en redeneren
- Het gebruik van woorden en symbolen om patronen te beschrijven en patronen te herhalen
- Taal

### Vaardigheden

- Kaart lezen
- Patroonherkenning
- Logica
- Opdrachten uitvoeren

### Leeftijd

- 9 en ouder

### Materialen

- Benodigd zijn: De verzameling eilandkaarten (de instructies op de achterkant moeten geheim blijven voor de leerlingen die de schat gaan zoeken): kopieer de kaarten, knip uit, vouw dubbel zodat de eilandnaam op de voorkant en de instructies op achterkant staan en plastificeer eventueel.

Voor iedere leerling:

- *Werkblad: Vind je weg naar Schateiland*
- Pen of potlood

Er zijn twee extra activiteiten waarvoor de leerling nodig heeft:

- *Werkblad: Schateilanden*
- *Werkblad: De mysterieuze munt*

# Schateiland

---

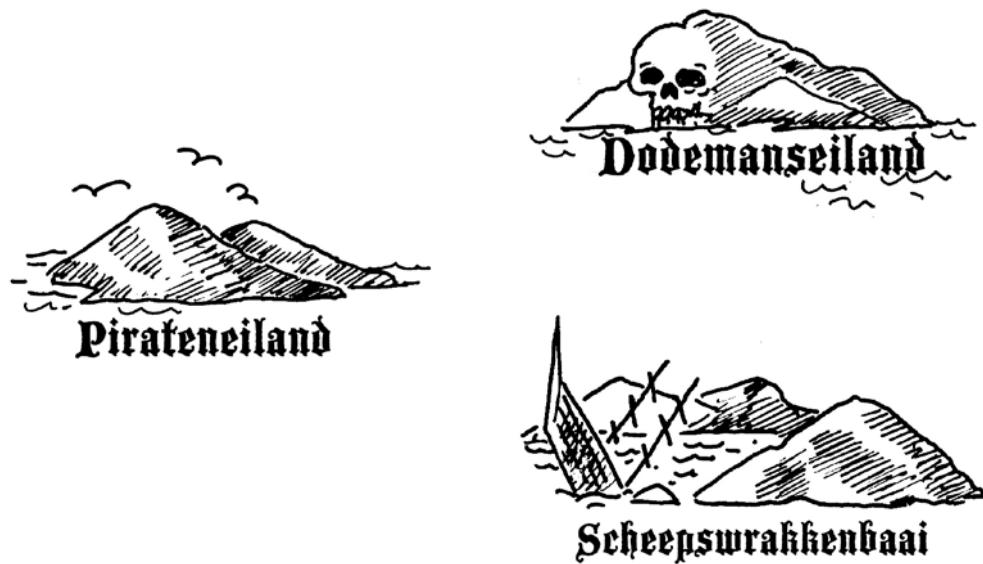
## Introductie

Het doel is een weg te vinden naar Schateiland. In dit deel van de wereld zijn er vriendelijke priatenschepen die vaste routes tussen eilanden varen. Vanaf ieder eiland vertrekken er twee schepen, A en B, en jij mag kiezen welke je neemt. Wat is de beste combinatie van schepen om zo snel mogelijk van Pirateneiland op Schateiland te komen? Op ieder eiland waar je aankomt kun je weer kiezen tussen schip A of B (niet allebei). De eilandbeheerder vertelt je naar welk eiland het door jou gekozen schip je brengt. Er is geen kaart die alle routes in beeld brengt, dus die zul je zelf moeten maken. Gebruik het werkblad om bij te houden hoe de eilanden verbonden zijn.

## Een voorbeeld vooraf

(Let op: De kaart in dit voorbeeld is een andere dan waar de kinderen mee aan de slag gaan.)

Schets op een bord drie eilanden in de zee zoals hieronder. Dit is de kaart zonder scheepsverbindingen:

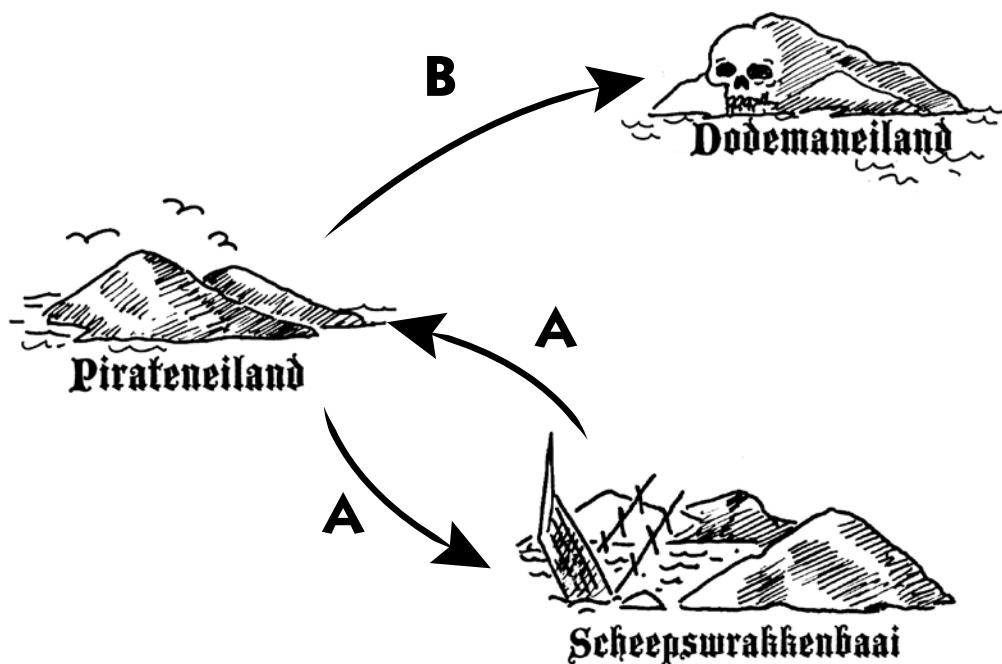


Kopieer de drie kaarten op de pagina “*Kaarten voor de uitleg van de activiteit*”, en vraag drie leerlingen om ieder één kaart vast te houden. De kaarten die hier gebruikt worden, horen bij dit voorbeeld en kunnen niet gebruikt worden voor de echte activiteit.

Je staat op Pirateneiland en vraagt om schip A. De leerling met de kaart Pirateneiland zegt je nu dat schip A je naar Scheepswrakkenbaai brengt. Teken een route A van Pirateneiland

naar Scheepswrakkenbaai. Op het eiland Scheepswrakkenbaai vraag je weer om schip A. Je zult teruggestuurd worden naar Pirateneiland. Teken ook deze route op de kaart. Terug op Pirateneiland vraag je nu om schip B. Dit schip brengt je naar Dodemaneiland, en daarvandaan vertrekken helaas geen schepen!

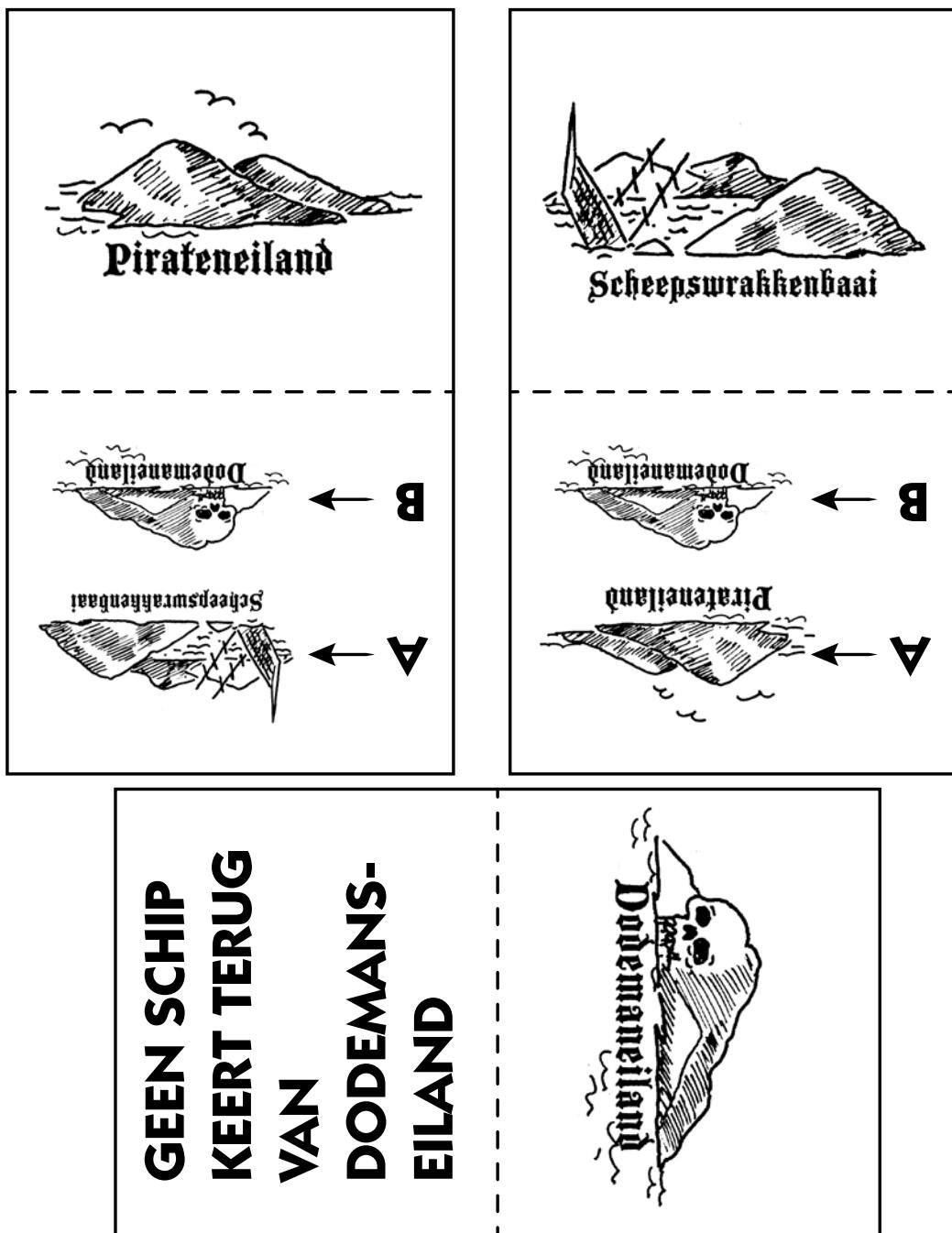
De kaart op het bord ziet er nu ongeveer zo uit:



## Kaarten voor de uitleg van de activiteit

Kaarten bij "Een voorbeeld vooraf".

Uitknippen, dubbelvouwen en dichtplakken met de achterkant. Zo dat de klas kan zien welk eiland de leerling is en de leerling kan zien welke boodschap hij of zij door moet geven.

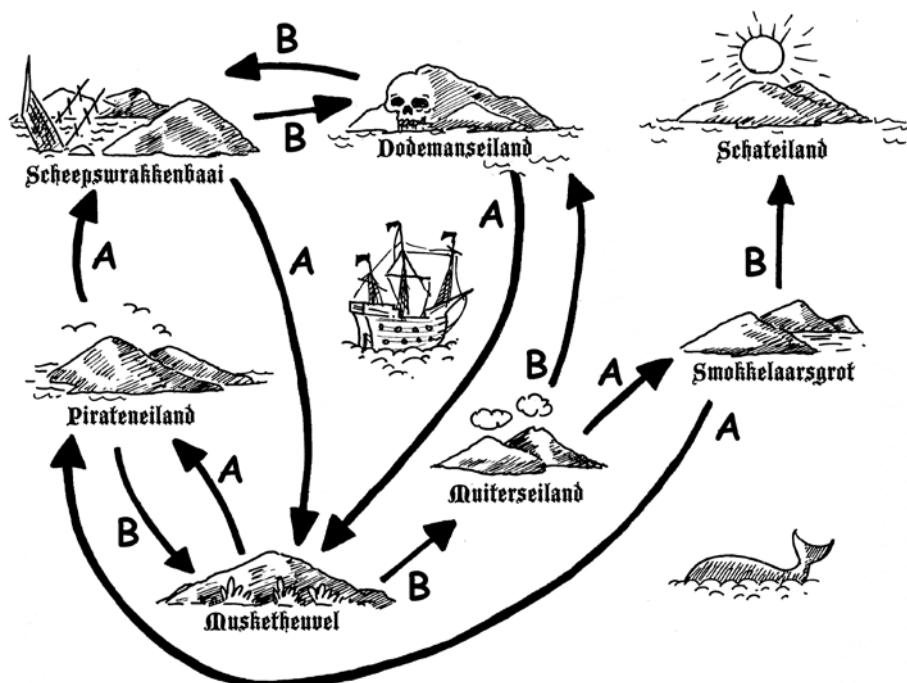


## Activiteit

Kies zeven leerlingen die eiland zullen zijn. Ieder van hen krijgt een kaart met aan de voorkant de naam en tekening van het eiland en de geheime instructies op de achterkant. De rest van de leerlingen krijgen het werkblad *Vind je weg naar Schateiland* en moeten de weg vinden van Pirateneiland naar Schateiland, die ze zorgvuldig intekenen op hun kaart. (Het is een goed idee om de leerlingen één voor één op pad te sturen zodat ze niet kunnen horen wat de andere kinderen ingefluisterd krijgen.)

Snelle zoekers: Probeer een andere route te vinden.

De volledig ingevulde kaart ziet er zo uit (laat deze dus niet aan de leerlingen zien!):

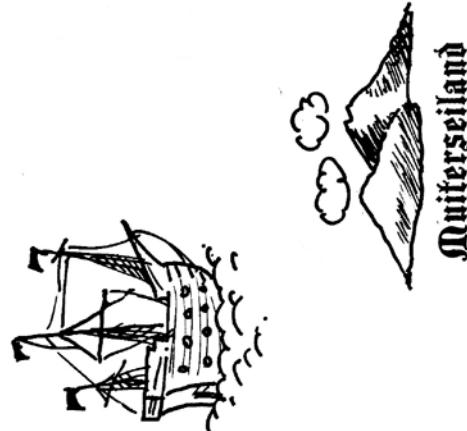


## Nabespreking

Wat is de snelste route? Kun je ook lange routes aangeven? Sommige routes bevatten een lus, je komt weer terug op een eiland waar je al was. Kan je hier een voorbeeld van vinden? (Bijvoorbeeld, BBBABAB en BBBABBABAB gaan allebei naar Schateiland.)

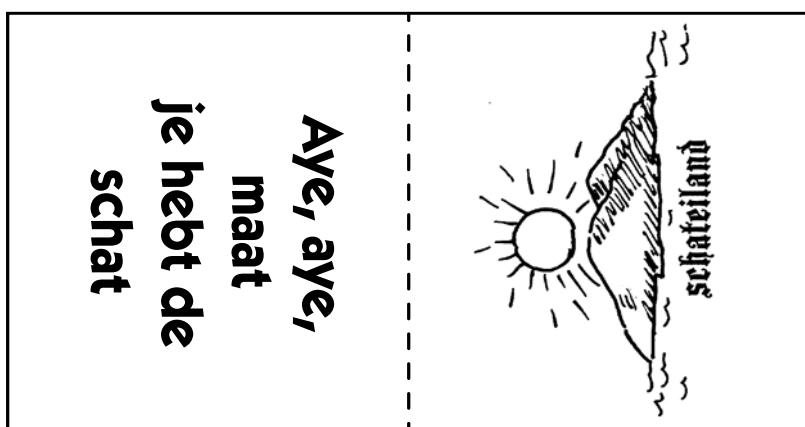
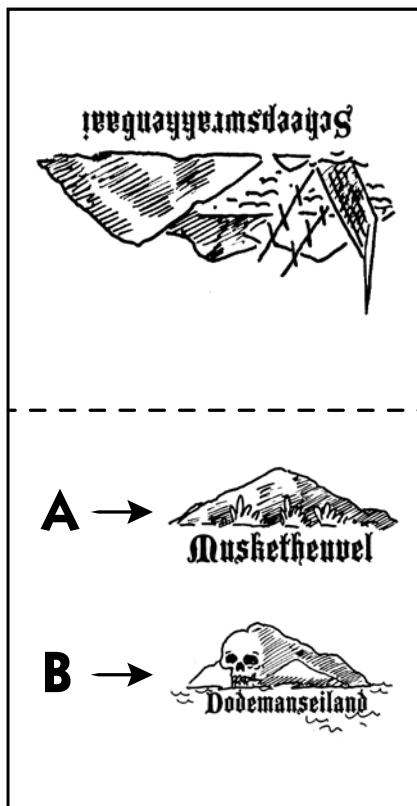
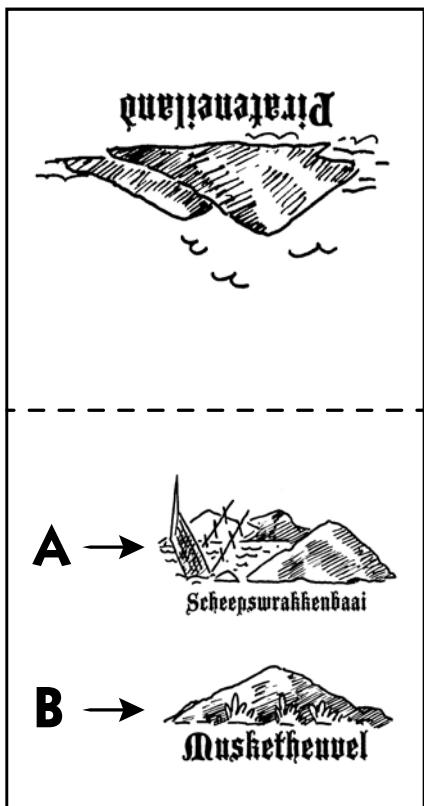
## Werkblad: Vind je weg naar Schateiland!

---

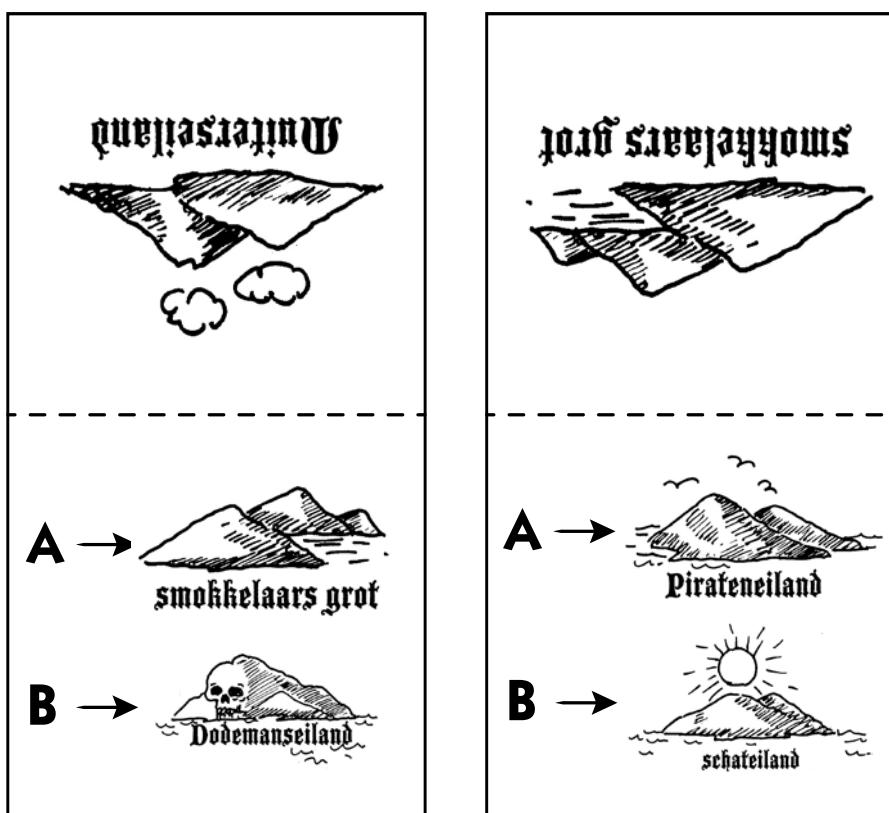
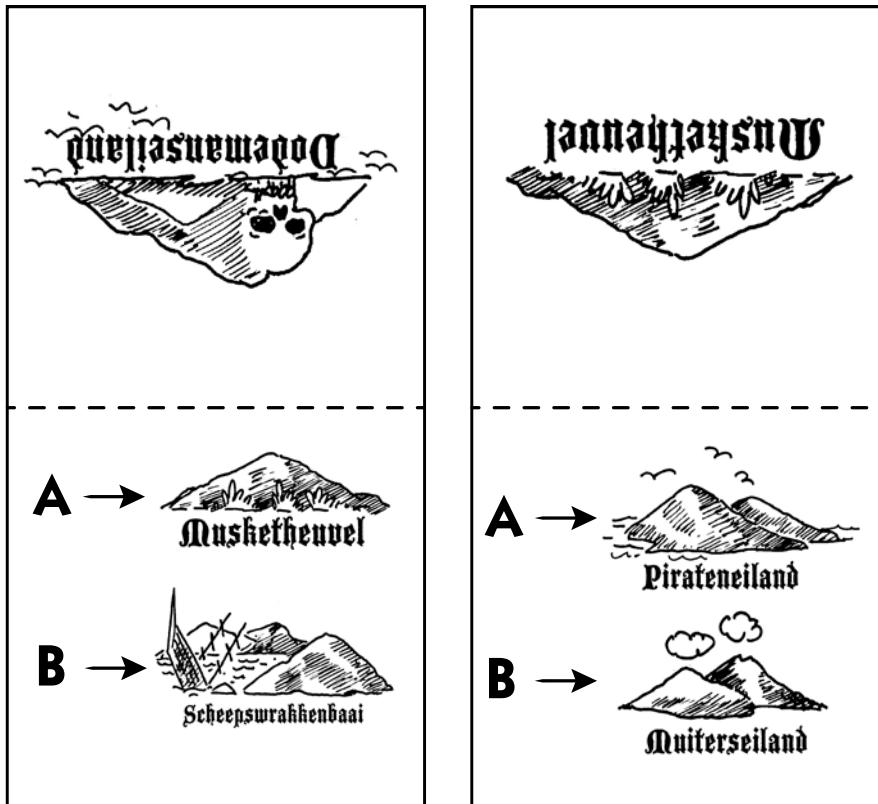


## Om te kopieren: Eilandkaarten 1/2

Kopieer de kaarten, knip ze uit en vouw ze dubbel. Laat de zeven eilandleerlingen ze als een dakje voor ze neerzetten met het eiland voor naar de klas gericht en de antwoorden naar zichzelf

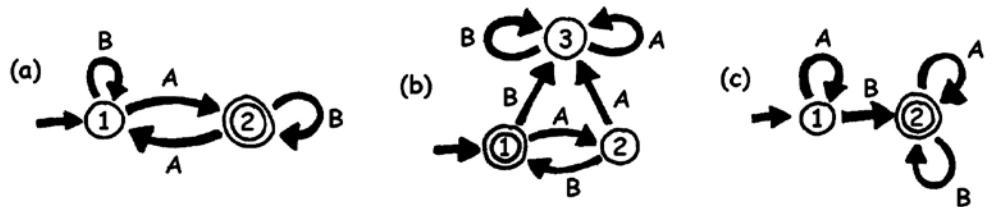


## Om te kopieren: Eilandkaarten 2/2



## Eindigetoestandsautomaat

Een andere manier om een kaart te tekenen is als volgt:



De eilanden zijn weergegeven door een genummerde cirkels, het doel-eiland, met de schat, is dubbelomcirkeld. Wat zijn de mogelijke routes om bij het doel te komen?

Merk op: Kaart (a) eindigt in de dubbele cirkel (eiland 2) uitsluitend als de route een oneven aantal A's heeft (bijvoorbeeld, AB, BABAA, or AAABABA).

Op kaart (b) kom je alleen in de dubbele cirkel met een route waarin A gevolgd door B nul of meer keer voorkomen (AB, ABAB, ABABAB, ...).

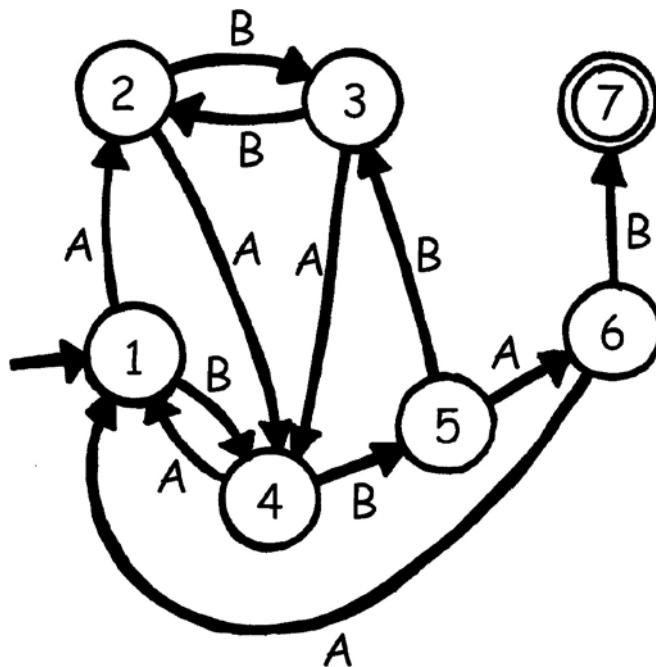
Bij kaart (c) is de eis dat er tenminste eenmaal een B in de route zit (de enige foute routes zijn die waarin alleen maar A's voorkomen: A, AA, AAA, AAAA, ...).

## Werkblad: Schateilanden

---

Kan jij je begraven schat goed verborgen houden? Hoe moeilijk kun jij het maken om de schat te vinden? Hoogste tijd om je eigen kaart te maken!

Hier is een schematische weergave van de kaart uit de opdracht. Computerdeskundigen gebruiken deze manier om snel en makkelijk uit te zoeken welke routes binnen het gewenste patroon, eindigen bij de schat op zeven, passen.

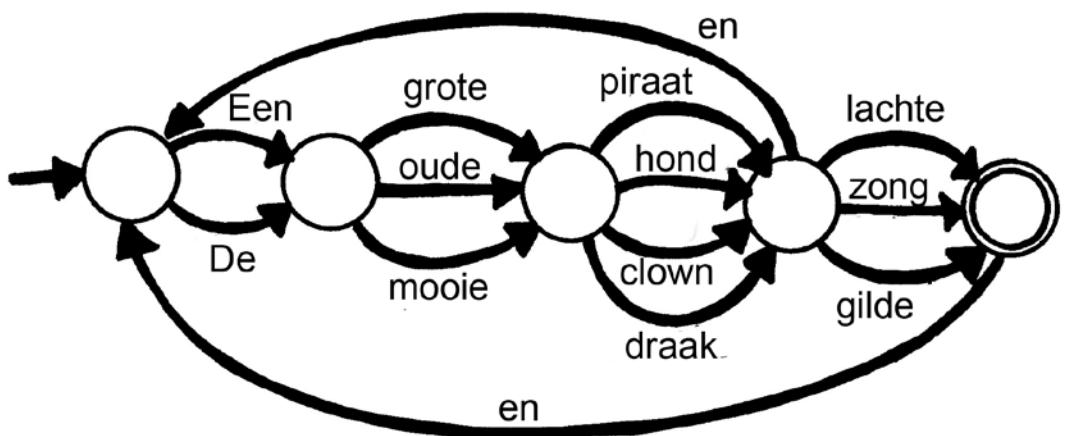


Teken op dezelfde manier je eigen schema waarop je goed kan zien welke routes door de piratenboten verzorgd worden. Teken vervolgens een lege kaart en maak de eilandkaarten met de geheime instructies op de achterkant. Wat is op jouw kaart de snelste manier om van Pirateneiland op Schateiland te komen?

Snappen jouw vrienden het door jou gemaakte schema? Geef ze een rij van A's en B's, en kijk of ze op het goede eiland uit komen.

Kun je nog meer opdrachten of spellen verzinnen met het schema? Zo'n schema is de weergave van een eindigetoestandsautomaat. Eindig omdat er maar eindig veel toestanden (eilanden) zijn, een automaat omdat precies wordt weergegeven hoe je van de ene toestand in de andere terecht komt.

Met het volgende schema kun je zinnen maken door willekeurige paden te volgen van de kaart en de woorden waar je langs komt op te schrijven.



Maak nu een eigen zinnenfabriek, misschien kun je er wel een heel verhaal mee vertellen!

## **Werkblad: De Mysterieuze Munt**

---

Vrienden hebben op het internet een spelletje gevonden waar een robot een munt opgooit en de speler moet proberen te raden of het kop of munt gaat worden. In eerste instantie leek het heel eenvoudig, de kans om te winnen was 50/50—althans dat dachten ze. Na een tijdje begonnen ze argwanend te worden. Er leek wel een patroon te zitten in de worpen van de munt.

Was het spel eerlijk? Helemaal niet! Ze besloten het te onderzoeken. Marie hield vanaf nu de resultaten bij en dit is wat ze vond: (k =kop, m = munt)

*k k m k k m k k m m k k k m m m k m k k k k  
k k m k k m m m k k m m m k k k k k m m k  
m m m m m k m m k m m m k k k m m k k k m k k  
k k k k k m m k k m m m m k k k k k k m m m m  
m m m*

Kun jij een voorspelbaar patroon vinden?

Er is een eenvoudige ‘kaart’ die de uitkomsten van de opeenvolgende worpen beschrijft.

Kun je die vinden? (Hint: er zijn maar 4 ‘eilanden’)

## Waar gaat dit eigenlijk over?

---

Een eindigetoestandsautomaat (of korter: eindige automaat; Engels: finite-state machine) wordt gebruikt om een computer een rij tekens of gebeurtenissen te laten verwerken.

Een eenvoudig voorbeeld is wanneer je een telefoonnummer belt en te horen krijgt “Toets 1 voor ..., toets 2 voor ..., toets 3 wanneer u een medewerker wilt spreken.”. Jouw toetsindrukken zijn de invoer voor de eindige automaat aan de andere kant van de lijn. Soms is de dialoog eenvoudig, maar hij kan ook heel complex worden. Het komt soms voor dat je met bepaalde toetsindrukken weer terugkomt waar je al was, er zit dan een lus (loop) in de eindige automaat. Als dat gebeurt, is er een fout gemaakt in het ontwerp van het systeem; als beller kun je er behoorlijk tureluurs van worden!

Een ander voorbeeld is wanneer je geld opneemt bij een PIN automaat. Het programma in de PIN-automaat leidt je langs een reeks gebeurtenissen. Intern weet de automaat welke rijen gebeurtenissen mogelijk zijn via een eindige automaat. Iedere toets die je indrukt brengt de automaat in een andere toestand. Sommige toestanden hebben opdrachten voor de PIN-automaat, bijvoorbeeld “geef € 50 biljet”, “druk een bewijsje af” of “werp de bankpas uit”.

Er zijn computerprogramma's die echt met de Nederlandse taal werken met kaarten zoals de zinnenfabriek op pagina 101. Zulke programma's kunnen zelf zinnen maken of iets doen met zinnen die door een gebruiker worden ingevoerd. In de zestiger jaren van de vorige eeuw schreef een computerdeskundige een beroemd geworden programma Eliza, genoemd naar Elize Dolittle, dat gesprekken kon voeren met mensen. Het programma deed zich voor als psychotherapeut en had openingszinnen als “Vertel me wat over je familie” en “Vertel me wat meer”. Hoewel het programma geen snars begreep van de zinnen die over en weer gingen, waren de zinnen zo goed dat sommige mensen het gevoel kregen met een echte psychotherapeut te spreken.

Computers blijven veel moeite houden met het begrijpen van natuurlijke taal maar met kunstmatige talen kunnen ze uitstekend overweg. Programmeertalen zijn belangrijke voorbeelden van kunstmatige taal. Computers lezen een programma geschreven in programmeertaal teken voor teken in en gebruiken dan een eindige automaat om het te vertalen naar hele eenvoudige taken die de computer kan uitvoeren.

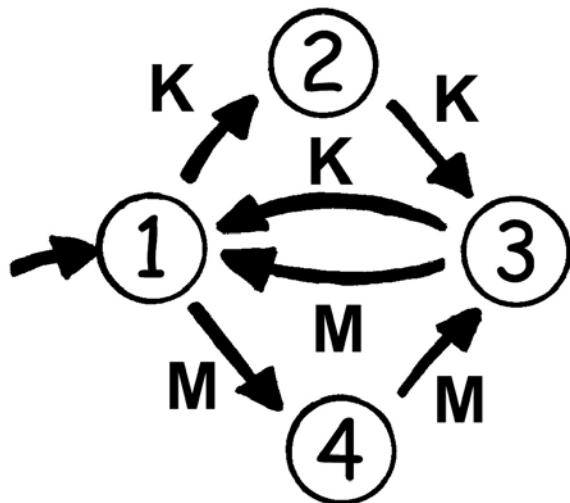


## Oplossingen en tips

---

### De mysterieuze munt

De robot in het spel van de mysterieuze munt gebruikte de volgende kaart om de uitkomst te bepalen:



Als je de mogelijkheden volgt, zie je dat je vanuit toestand 1 altijd na drie worpen weer terug bent. Van die drie worpen zijn de eerste en de derde eerlijk, de tweede worp is echter altijd een herhaling van de eerste.

$k\ k\ m$	$k\ k\ m$	$k\ k\ k$	$m\ m\ k$	$k\ k\ k$	$m\ m\ k$	$m\ m\ m$	$k\ k\ k$
$k\ k\ m$	$k\ k\ k$	$m\ m\ m$	$k\ k\ k$	$m\ m\ m$	$k\ k\ k$	$k\ k\ k$	$m\ m\ k$
$m\ m\ m$	$m\ m\ k$	$m\ m\ k$	$m\ m\ m$	$k\ k\ k$	$m\ m\ k$	$k\ k\ m$	$k\ k\ k$
$k\ k\ k$	$k\ k\ k$	$m\ m\ k$	$k\ k\ m$	$m\ m\ m$	$k\ k\ k$	$k\ k\ m$	$m\ m\ m$
$m\ m\ m$							

# Activiteit 12

---

## Marsorders—Programmeertalen

### Samenvatting

Computers worden meestal geprogrammeerd met behulp van een “taal” met een beperkte hoeveelheid van instructies die kunnen worden opgevolgd. Een van de meest frustrerende dingen van programmeren is, dat computers de instructies altijd letterlijk gehoorzamen, zelfs als het een gek resultaat oplevert. Deze activiteit geeft leerlingen enige ervaring met dit aspect van het programmeren.

### Kerndoel

- Taal: Informatie verwerven uit gesproken taal, Groep 5 of hoger.

### Vaardigheden

- Instructies geven en opvolgen.

### Leeftijd

- 7 jaar en ouder

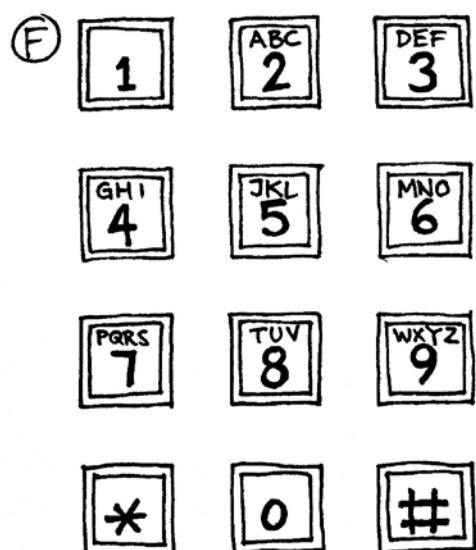
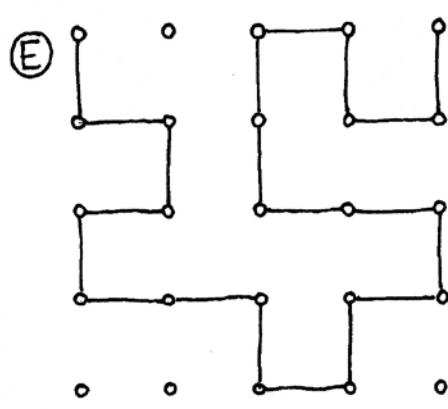
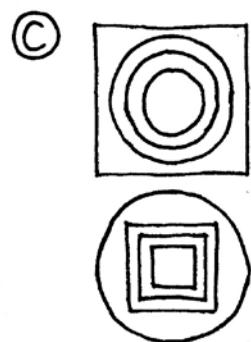
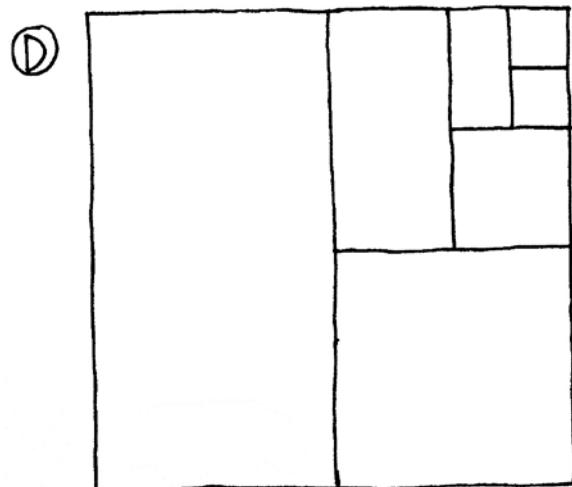
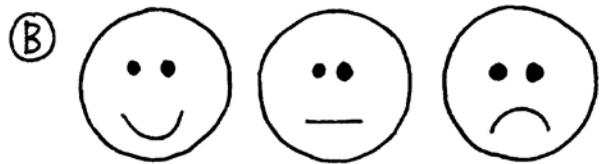
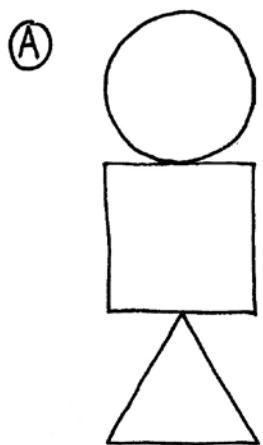
### Materialen

Je hebt nodig:

- Kaarten met afbeeldingen zoals op de volgende bladzijde.

Elke leerling heeft nodig:

- Potlood papier en liniaal



# Marsorders

---

## Inleiding

Bespreek of het goed zou zijn als mensen instructies altijd precies uitvoeren. Bijvoorbeeld, wat zou er gebeuren als je naar een gesloten deur wijst en zegt: "Ga door die deur"?

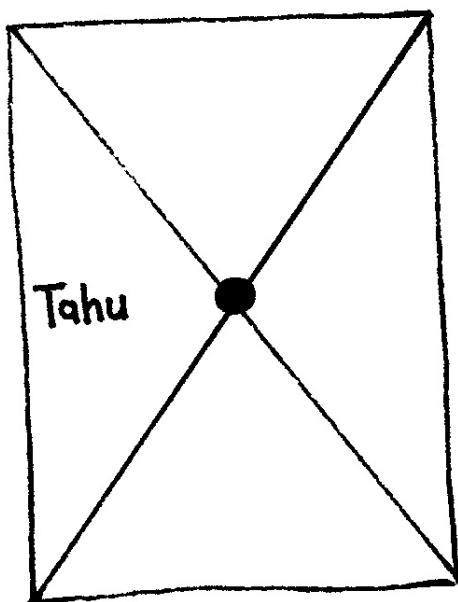
Computers werken door lijsten met instructies op te volgen, en ze doen precies wat de instructies zeggen, zelfs als ze geen zin hebben!

## Demonstratie voorbeeld

Kijk of de leerlingen een tekening kunnen maken met behulp van deze instructies:

1. Teken een stip in het midden van je blad.
2. Begin in de linkerbovenhoek van de pagina en trek een rechte lijn door de stip tot aan de rechterbenedenhoek.
3. Begin in de linkerbenedenhoek van de pagina en trek een rechte lijn door de stip tot aan de rechterbovenhoek.
4. Schrijf je naam in de driehoek in het midden van de linkerkant van de pagina.

Het resultaat moet er ongeveer zo uit zien::



## **Activiteiten**

Kies een leerling en geef hem een afbeelding (zoals de voorbeelden op de pagina). Het kind beschrijft de foto voor de klas en de anderen tekenen. De leerlingen kunnen vragen stellen ter verduidelijking van de instructies. Het doel is om te zien hoe snel en nauwkeurig de oefening kan worden voltooid.

Herhaal de oefening, maar deze keer mogen de leerlingen geen vragen te stellen. Het beste is om een eenvoudigere afbeelding te gebruiken voor deze oefening, omdat de leerlingen heel snel kunnen verdwalen.

Probeer nu de oefening met de opdrachtgevende leerling verborgen achter een scherm, zodat de enige communicatie is in de vorm van de gesproken instructies.

Wijs erop dat deze vorm van communicatie het meest lijkt op wat computerprogrammeurs ervaren bij het schrijven van programma's. Ze geven een reeks instructies aan de computer, en ze zien het effect van de instructies pas achteraf.

Laat de leerlingen een tekening maken en hun eigen instructies schrijven. Test de instructies uit in tweetallen of met de hele klas.

## **Variaties**

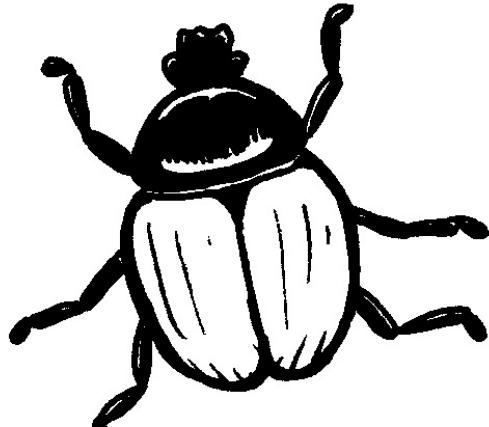
- Schrijf instructies om een papieren vliegtuigje te maken
- Schrijf instructies die leiden naar een geheimzinnige plek bij de school en gebruik daarbij instructies zoals “Ga vooruit x meter”, “ga links” (90 graden), and “ga rechts” (90 graden).
- Leerlingen moeten hun instructies testen en verfijnen, totdat ze het gewenste effect hebben.
- Blindenmannetje. Blinddoek een leerling en de andere leerlingen sturen hem de klas rond.

## **Waar gaat het eigenlijk over?**

---

Computers werken door het volgen van een lijst van instructies, beter bekend als een programma of script, dat is geschreven om een bepaalde taak uit te voeren. Programma's zijn geschreven in talen met een beperkte set van instructies die speciaal zijn ontworpen om computers te vertellen wat te doen. Sommige talen zijn meer geschikt voor bepaalde doeleinden dan andere.

Ongeacht welke taal ze gebruiken, moeten programmeurs bedreven zijn in het precies specificeren wat ze willen dat de computer doet. In tegenstelling tot de mens, zal een computer instructies naar de letter uitvoeren, zelfs als ze uitgesproken belachelijk zijn.



Het is belangrijk dat programma's goed geschreven zijn. Een kleine fout kan een hoop problemen veroorzaken. Stel je de gevolgen voor van een fout in het programma van een computer tijdens de lancering van een space shuttle, of in een kerncentrale, of bij de seinen op een treinspoor! Fouten worden gewoonlijk "bugs" genoemd naar aanleiding van (zo wordt gezegd) een nachtvlinder die ooit werd verwijderd ("debuggen") van een elektrisch relais (schakelaar) in een elektronische rekenmachine in 1940.

Hoe complexer het programma, hoe meer fouten er waarschijnlijk in zitten. Dit werd een groot probleem toen de VS werkte aan het Strategic Defence Initiative-programma ("Star Wars"), een computergestuurd systeem dat bedoeld was om een ondoordringbare verdediging tegen nucleaire aanvallen te vormen. Sommige computer wetenschappers beweerden dat het nooit zou kunnen werken als gevolg van de complexiteit en de inherente onbetrouwbaarheid van de benodigde software. Software moet zorgvuldig worden getest om zoveel mogelijk bugs te vinden en het zou niet haalbaar zijn om dit systeem te testen, omdat men dan raketten zou moeten afvuren op de Verenigde Staten om zeker te zijn dat het werkte!