

Bètadag

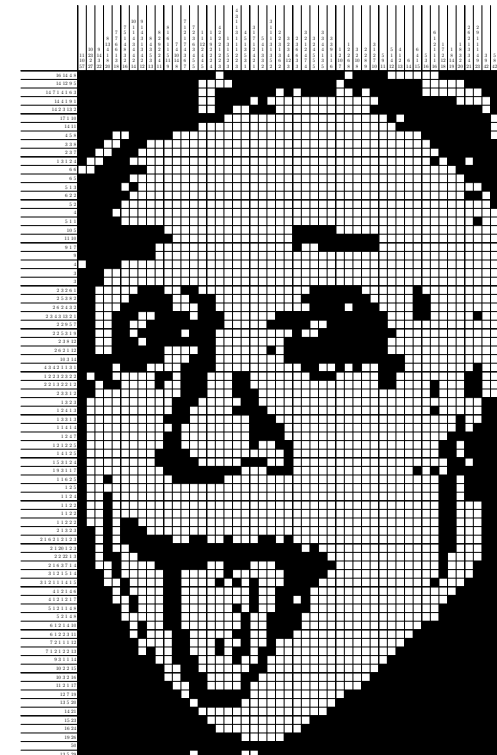
---

## Japanse puzzels

dr. Walter Kusters, Informatica (& Wiskunde)

vrijdag 23 januari 2009

[www.liacs.nl/home/kusters/nono/](http://www.liacs.nl/home/kusters/nono/)



Als je **Japanse puzzels** zegt, denkt iedereen aan **Sudoku**.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Als je **Japanse puzzels** zegt, denkt iedereen aan **Sudoku**.

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

bron: Wikipedia

Maar wij gaan het vandaag hebben over **Nonogrammen**.

Een **Nonogram** is een puzzel; een klein voorbeeld:

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

Naast iedere rij en boven iedere kolom staan in volgorde de lengtes van aaneengesloten series **rode** (of zwarte) vakjes.

Waar moeten die **rode** vakjes komen?

De oplossing ziet er zo uit:

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

	●		●	
●				●
	●	●	●	

Naast iedere rij en boven iedere kolom staan in volgorde de lengtes van aaneengesloten series **rode** (of zwarte) vakjes.

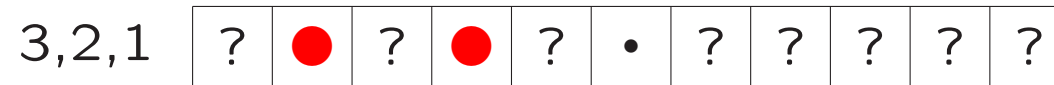
Hoe los je Nonogrammen op?

De meeste mensen gebruiken **logische regels**, en **heuristieken** = **vuistregels** zoals “redeneer eerst een keer via de rijen, en dan via de kolommen”.

Een voorbeeld van een logische regel is: “als het getal 3 naast een rij van breedte 5 staat, moet het middelste vakje wel rood zijn”. Je kijkt dan eigenlijk naar één rij of kolom.

Een heuristiek uit het dagelijks leven: als je een bedrag (zeg 80 cent) moet betalen, kun je het beste de grootste kleinere munt (50 cent?) die je hebt eerst geven.

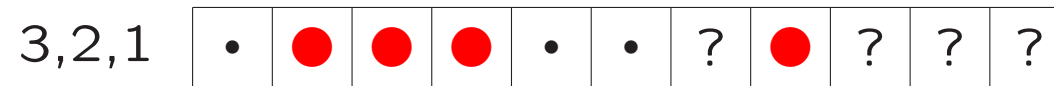
Stel dat je van een rij al weet:



Een • betekent een zeker leeg vakje, een ● staat voor een zeker gevuld vakje. De rest is nog onbekend.

Wat kun je hier nu concluderen?

We concluderen dan dat voor deze rij geldt:



Een • betekent een zeker leeg vakje, een ● staat voor een zeker gevuld vakje. De rest blijft nog onbekend.

Dus door naar een enkele rij of kolom te kijken kun je vooruitgang boeken.

Hoe kun je zulke conclusies trekken? Of beter nog: hoe kan een computer(programma) dat?

Een “flauwe” oplossing is: probeer alle mogelijkheden, en kijk of die iets gemeenschappelijk hebben. Dit heet wel **brute-force**. Een computer is daar erg goed in.

Dat kan voor een enkele rij of kolom van een Nonogram wel handiger = efficiënter, met **dynamisch programmeren**: wat is het volgende getal in 1, 1, 2, 3, 5, 8, 13, 21, 34, ...?

Hoe ver komen we als je alleen per rij/kolom kijkt? Een • betekent weer een zeker leeg vakje, een ● staat voor een zeker gevuld vakje.

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

•	•	•	•	•
•	●	•	●	•
•	•	•	•	•
?	?	•	?	?
?	?	●	?	?

Maar nu zitten we vast ... tenzij we rijen en kolommen *samen* bekijken.

Dit hadden we:

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

	•	•	•	•	•
	•	●	•	●	•
	•	•	•	•	•
<i>v</i>	<i>w</i>	•	?	?	
<i>u</i>	<i>x</i>	●	?	?	

Stel dat  $u = \bullet$ , dan (kolom) moet  $v$  leeg zijn, en dus (rij)  $w = \bullet$ , en dus (kolom) moet  $x$  leeg zijn. Tegenspraak (rij)! Dus  $u$  moet leeg zijn.

Dat was een lastige logische redenering, ook voor een computer. Maar de rest is nu eenvoudig.

Soms heeft een probleem meer mogelijke (goede en foute) oplossingen dan je kunt doorrekenen. Zo heeft een  $5 \times 5$  Nonogram al

$$2^{25} = 2^{10} \cdot 2^{10} \cdot 2^5 = 1024 \cdot 1024 \cdot 32 \approx 32 \text{ miljoen}$$

mogelijke invullingen! Er zijn immers  $5 \times 5 = 25$  vakjes die elk 2 mogelijkheden hebben.

De “ $80 \times 50$  Einstein” heeft  $2^{4000} \approx 10^{1200}$  mogelijkheden.

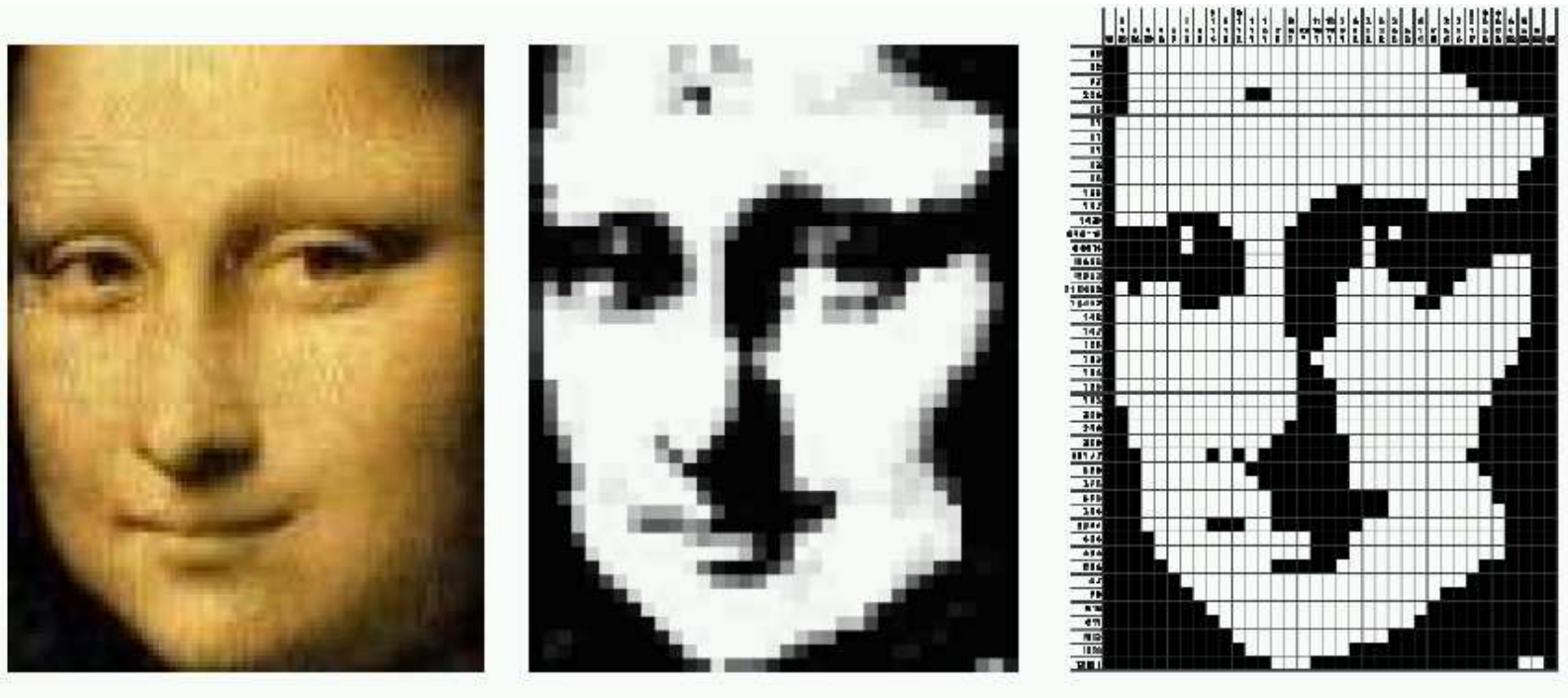
Dus brute-force, alles domweg proberen, lost een complete puzzel niet *snel* op ...

Het oplossen van een Nonogram is een **NP-volledig** probleem, net als het maken van een schoolrooster. Dat kan eigenlijk alleen goed met brute-force, maar dan duurt het veel te lang. Van een mogelijke oplossing kun je wel snel zien of hij goed of fout is.

Het grootste open probleem in de informatica  $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$  gaat hierover. Je kunt een miljoen dollar verdienen als je dit oplost: de **Clay Prize**.

Het gaat daar niet om het oplossen van één puzzel of maken van één schoolrooster, maar om het vinden van een efficiënte algemene methode — als die al bestaat.

Hoe maak = construeer = ontwerp je zelf een Nonogram?



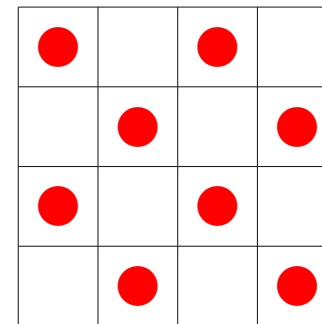
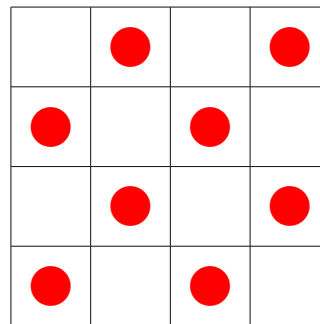
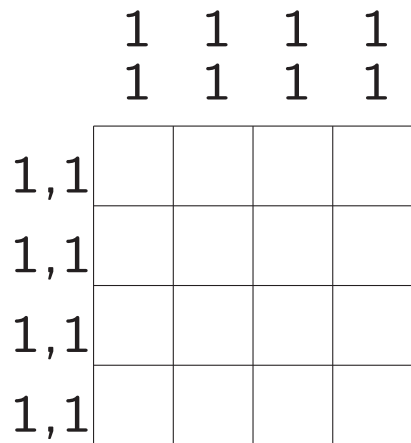
kleurenfoto

grijswaarden-plaatje

puzzel

Denk eraan dat een *goed* Nonogram een **unieke** oplossing moet hebben.

Ze hebben in het algemeen namelijk soms heel verschillende oplossingen:

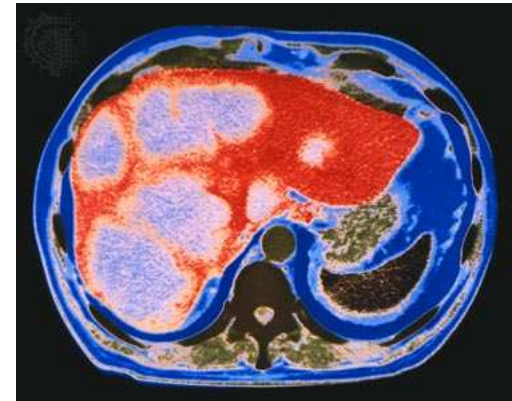


Waarom doen wetenschappers Nonogrammen?

**Tomografie** houdt zich bezig met het volgende probleem:  
Hoe reconstrueer je een object uit **projecties**?

Voorbeelden:

- Nonogrammen oplossen
- Hoe zien de organen eruit, gegeven CT-scans?
- Waar zitten de “gaten” in een diamant?

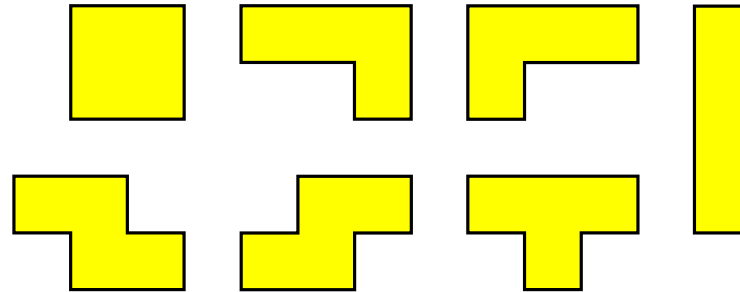


Ook aan een spel als **Tetris** kleven allerlei vragen:

- Hoe speel je het zo goed mogelijk?  
→ **AI** = Kunstmatige intelligentie
- Hoe moeilijk is het?
- Wat kan er allemaal gebeuren?  
Je kunt bijna alle “configuraties” bereiken, zie:

`www.liacs.nl/home/kosters/tetris/`

De 7 Tetris-stukken:



Stukken vallen random; volle regels worden verwijderd.

De vraag “Kun je met een gegeven serie (inclusief volgorde) van deze stukken een bord helemaal leeg spelen?” is heel erg moeilijk te beantwoorden. Een **brute-force** berekening lost het op ... maar duurt erg lang.

De vraag is **NP-volledig**, net als de vraag of een graaf een Hamilton-circuit heeft (“kun je rondwandelen?”), en net als de vraag of een Japanse puzzel (uniek) oplosbaar is.