

Programmeertheorie

Wouter Vrielink

Wat is Programmeertheorie?

A.K.A. Heuristieken

- Project van 4 weken
- Groepen van 3
- Dagelijks van 9-17 werken aan
- 6 “cases”; onoplosbare (?) problemen
- Assistentie
 - Mentoren
 - Dagelijks

Wat zijn heuristieken?



Locale strategien



Shortcuts



Indicatie van goede oplossingen



Geen garanties

Optimaal
Perfect
Rationeel

Heuristics in practice

Probleem:

- Reis van Limburg naar Groningen
- Geen bordjes met Groningen

Oplossingsvorm:

- Gebruik weg X om bij weg Y te komen
- Gebruik weg Y om bij weg Z te komen



Heuristics in practice

Probleem:

- Reis van Limburg naar Groningen
- Geen bordjes met Groningen

Oplossingsvorm:

- Gebruik weg X om bij weg Y te komen
- Gebruik weg Y om bij weg Z te komen

Heuristiek:

- Neem de weg die het meest naar het noorden gaat





Gebruik heuristieken om sneller te zoeken!

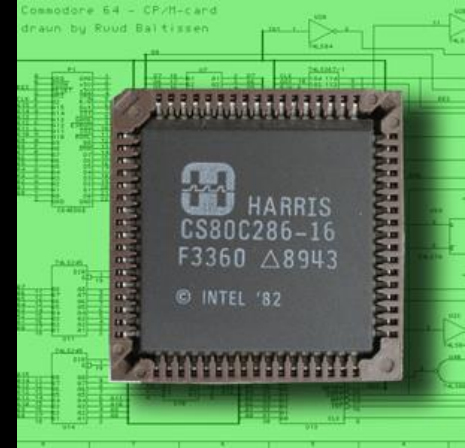
Wouter Vrielink

6 Cases

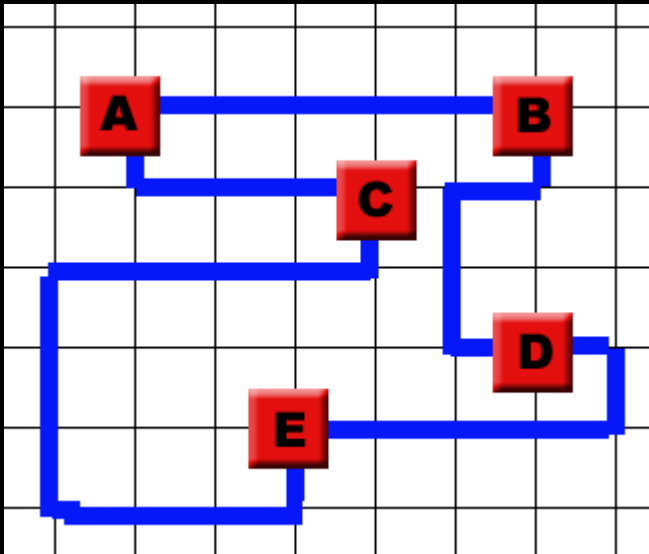
- Wat maakt een probleem 'onoplosbaar'?
 - Datastructuur/Representatie
 - Oplossingen/deel-oplossingen
 - De sleutel! Algoritmes
-
- | | |
|--|--|
| <ul style="list-style-type: none">• Lectures & lesroosters• Chips & Circuits• Protein Pow(d)er | <ul style="list-style-type: none">• RailNL• Rush Hour• SmartGrid |
|--|--|

6 Cases

1 – Chips & Circuits



Netlist #1
A - B
A - C
C - E
D - B
D - E

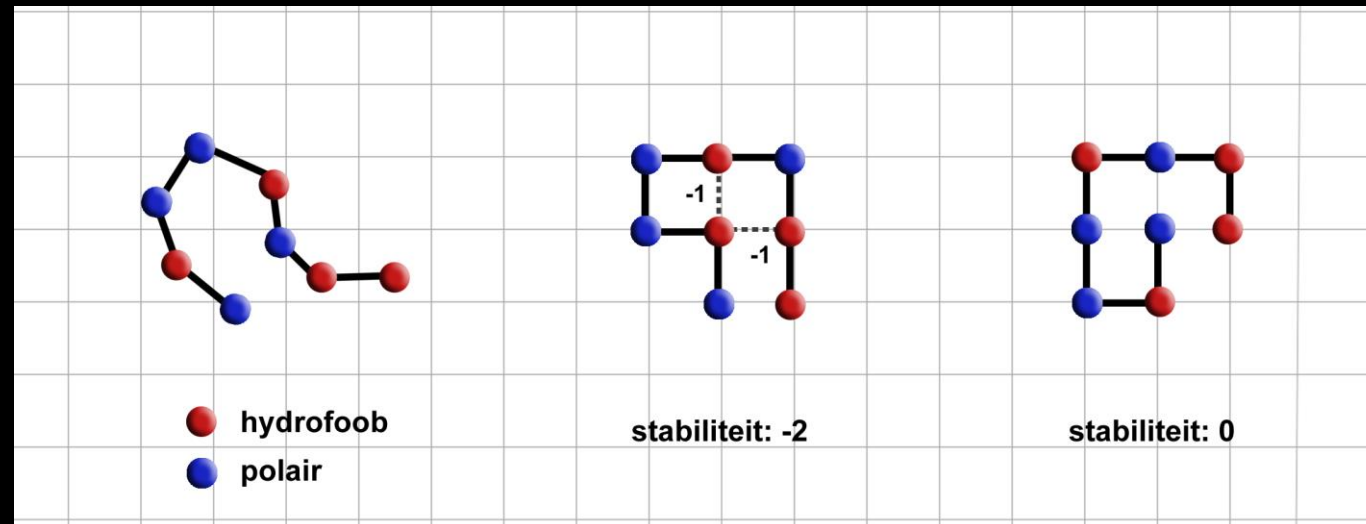


- Twee chips, met elk drie Netlists
- Gates (logische poorten) verspreid over de chips
- Specifiek paren van gates moeten met elkaar verbonden worden
- Kortsluiting voorkomen!
- Hoe korter de kabels, des te beter de oplossing

6 Cases

2 – Protein Pow(d)er

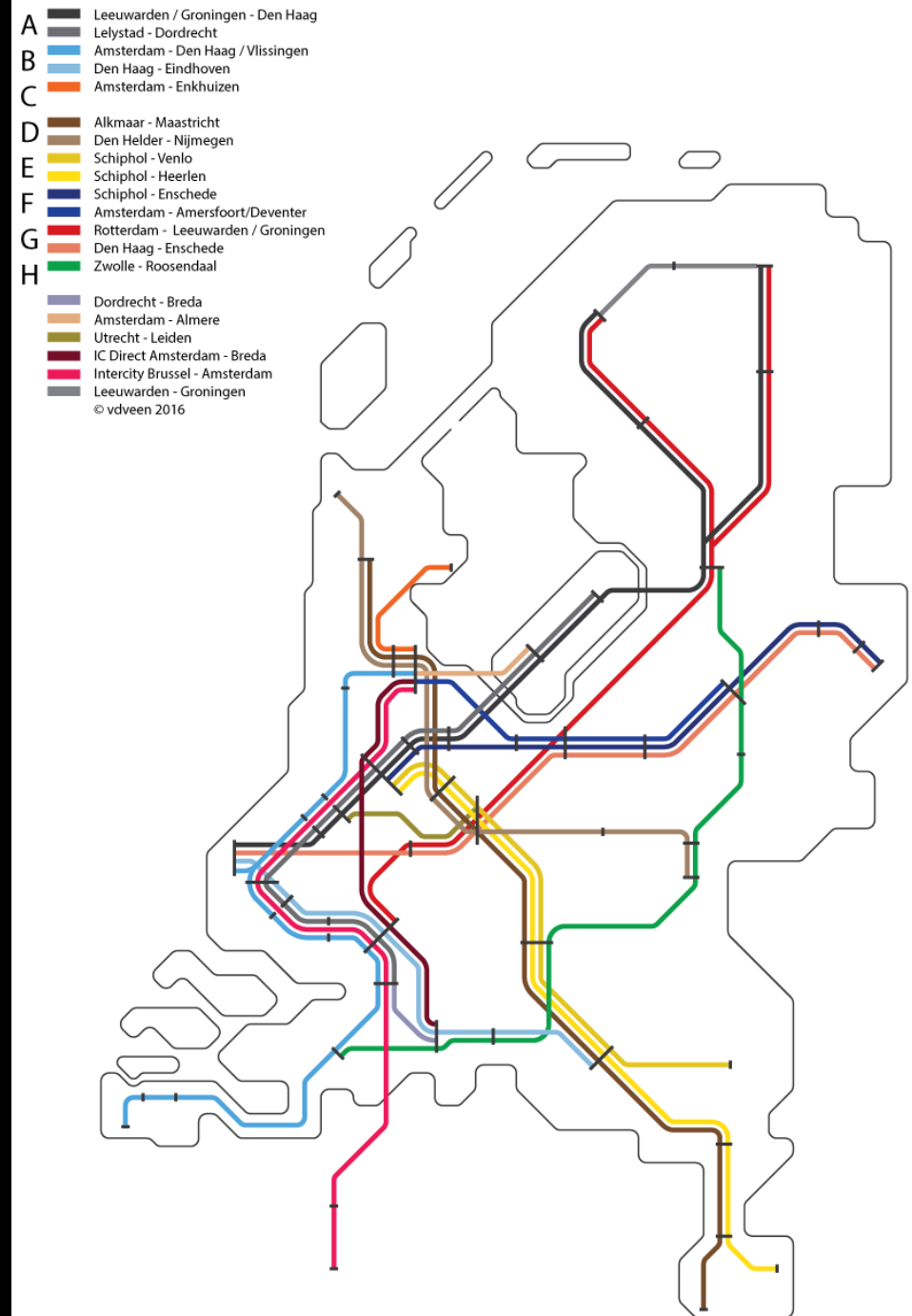
- Eiwitten vouwen: HHPHHPHPHHPH
- 2D HP-Model
- Hydrofobe en Hydrofiele aminozuren
- Stabiele vouwing wanneer hydrofobe aminozuren naast elkaar liggen



6 Cases

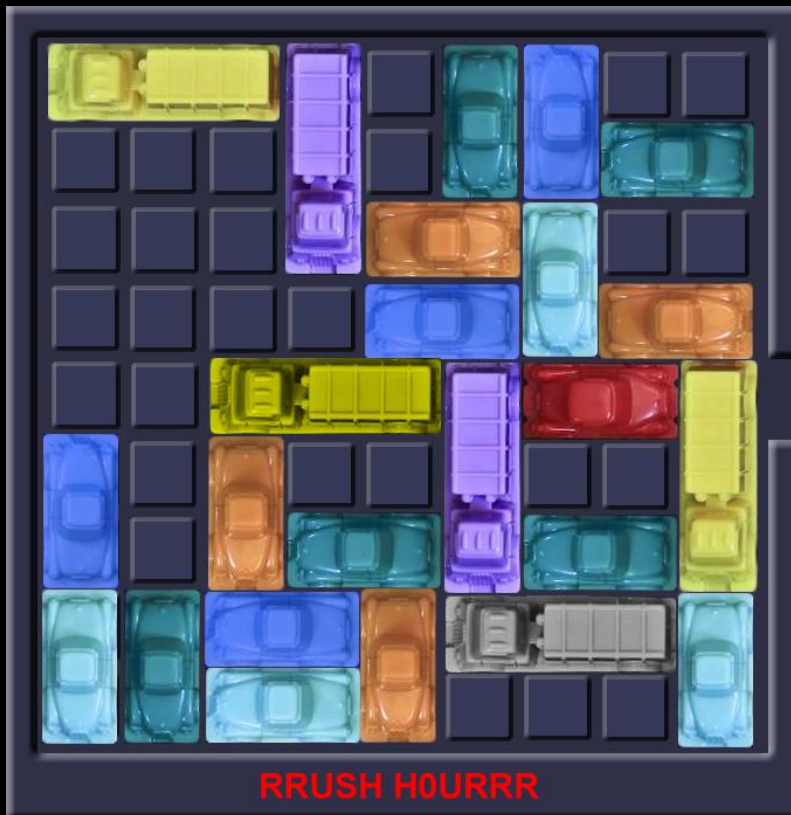
3 – RailNL

- De treinvoering in Noord+Zuid Holland en daarna heel Nederland
- Plan trajecten in voor treinen
- Zo veel mogelijk stations passeren, in zo min mogelijk minuten
- Houdt rekening met knelpunten zoals Utrecht en doodlopende sporen zoals Den Helder.



6 Cases

4 – Rush Hour



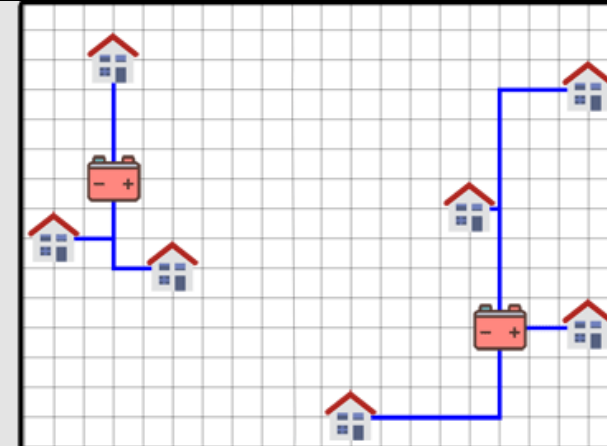
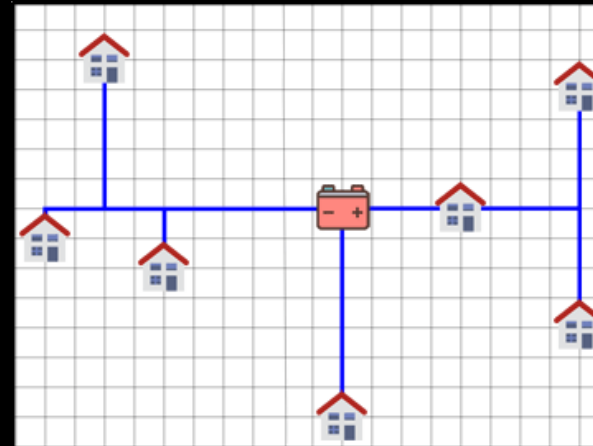
- Klassiek kinderspel
- Rij de rode auto naar buiten
- 7 verschillende spelboorden
- Voorkom botsingen



6 Cases

5 – SmartGrid

- Drie wijken; huizen en batterijen op vaste plaats
- Huizen wekken energie op
- Energie opslaan in batterijen
- Voorkom overcapaciteit
- Alle huizen op een batterij aansluiten
- Zo min mogelijk kabel gebruiken



6 Cases

6 – Lectures & lesroosters

- Vakken, zalen, activiteiten, studenten
- Geldig rooster maken
- Maluspunten



Ons team

Docenten

- Wouter Vrielink
- Bas Terwijn
- Jelle van Assema
- Anuj Pathania

Assistenten

- Joos Akkerman
 - Marijn Doeve
 - Mayla Kersten
 - Quinten van der Post
 - Alwan Rashid
 - Pamela Sneekes
-

Contacturen

- Dagelijks van 9-17
- Hoorcolleges
- Algemene techassistentie
- Met de assistent
 - Persoonlijke techassistentie
 - Presentatiesessie

Wat gebeurt er tijdens techassistentie?

- Éénmaal per week ~30 min
- Hulp bij (nadenken over) programmeren en je case
- Maar ook
 - Bugs bespreken
 - Uitleg
 - Afspraken maken
 - Discussie over o.a. infra-/data-structuren/algoritmen
 - Voortgang bespreken

Wat gebeurt er tijdens presentatiesessies?

- Presentaties (~10 min)
- Discussie (~5-10 min)

Beoordeling

- Milestones 3/10
- Uitleg van inzicht 50%
 - Presentatie
- Eindproduct 50%
 - Code
 - Git
- Op een schaal van 0-5

Beoordeling code

- Minimaal aanwezig
 - README (+ inhoud)
 - Duidelijke en inzichtelijke repo structuur
 - Geen crashes
 - [requirements.txt](#) of equivalent
 - Presentatie

Beoordeling code

- Reproducibility
 - README
 - Genereren van oplossingen/metriecken
 - Methode van presenteren oplossingen/metriecken
- Code
 - Modulariteit
 - Abstractie
 - Duplicate/redundant code
 - Style
 - Documentatie
 - etc

GitHub

“GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.”

GitHub != Dropbox

Git README.md

- Inleiding
 - Installatie -> requirements.txt
 - Waar kan ik wat vinden? Zet alles netjes in mappen.
 - Tests / hoe run ik dit programma
 - Auteurs
-
- Zie voor een voorbeeld:
 - <https://github.com/minprog/voorbeeld-repo>

Week	Onderdeel
1	Interpretatie van de case
	Dinsdag 9 uur: Openingscollege
	Woensdag 15 uur: College over problemen
	Vrijdag 15 uur: Live coding college
2	Een baseline zetten
	Dinsdag 9 uur: College over zoekalgoritmes
	Woensdag 9 uur: College over optimalisatie-algoritmes
	Vrijdag 15 uur: Live coding college
3	Algoritmes & resultaten
	Woensdag hele dag: Midterm presentaties
	Donderdag ? uur: Verdiepingscollege Refactoring & profiling
4	Vergelijken & presenteren
	Donderdag & Vrijdag: Eindpresentaties

Door de weken heen

- Week 1 - Interpretatie van de case
 - Wat is mijn case?
 - Welk (sub)probleem moet ik oplossen? Wat is een oplossing en hoe ziet die er uit?
 - Hoe kan ik die (digitaal) representeren? -> datastructuur
 - Hoe genereer ik random een oplossing?
- Week 2 - Een baseline zetten
 - Random is af. Kan resultaten laten zien.
 - Hoe genereer ik betere oplossingen dan random?
 - Waarom zou dit beter zijn?
- Week 3 - Algoritmes & resultaten
 - Maken nieuwe algoritmes
- Week 4 – Vergelijken & presenteren



Zometeen