

Nascholing

Algoritmisch Denken



Martin Bruggink
Renske Smetsers



Speerpunten bijeenkomst #2

- Praktische bijeenkomst
- Hele ontwikkelcyclus doorlopen
 - Algoritme ontwerpen
 - Flowchart
 - Code
 - Reflectie

Programma tweede bijeenkomst

- Terugkoppeling sorteren
- AD in nieuw examenprogramma / onderzoek (Erik Barendsen)
- Diner (+/- 17.45)
- Aan de slag met lesmateriaal
- Unplugged
- Afsluiting
- Einde (20.00)

Warming-up

- Denk aan een pareltje uit je eigen informatica klassen, bij voorkeur gerelateerd aan programmeren. Waar kijk je met plezier/trots op terug?
- Deel in tweetallen je pareltjes.



Terugkoppeling sorteren



Erik Barendsen

Rollen:

- hoogleraar vakdidactiek informatica (OU)
& universitair hoofddocent (RU)
- voorzitter van de vernieuwingscommissie
Informatica vwo-havo

Kerndoel van Alg. Denken

“... first and foremost, it should teach those young people to THINK.”

George Pólya (1887 – 1985)



The big 5:

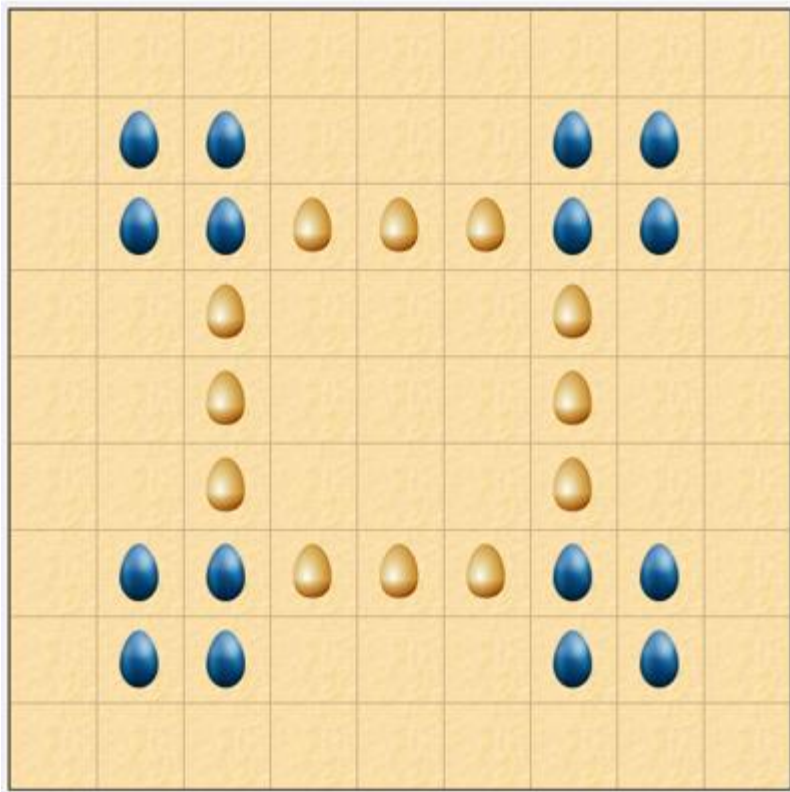
- Algoritmiek: denken in regels en opeenvolgingen (van deeltaken)
- Abstractie: hoog niveau / weglaten details
- Decompositie: opdelen in deelproblemen
- Generalisatie: patronen & generieke opl.
- Evaluatie: voldoet oplossing aan eisen?
had het beter/makkelijker gekund?

Lesmateriaal

<http://course.cs.ru.nl/greenfoot/docenten.html>

- Nascholing materiaal/bestanden
- Lesmateriaal voor leerlingen
- Lesmateriaal voor docenten

Omschrijf een algoritme



Bestaande methodes:

layBlueEgg

layGoldenEgg

move

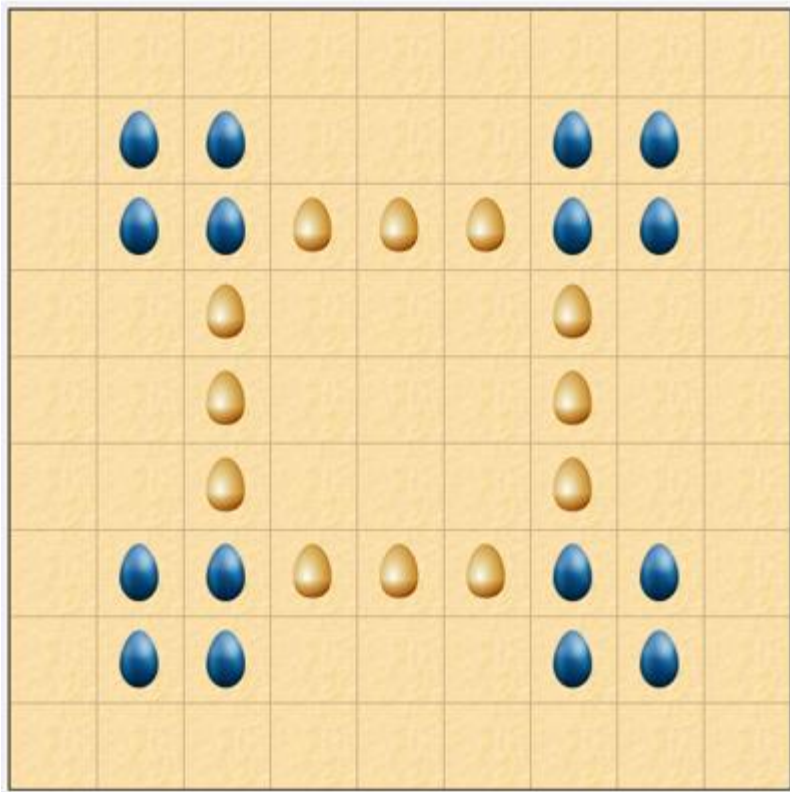
turnLeft

turnRight

Keuze:

Flowchart of code

Verzanden in details...



layBlueEgg

move

turnLeft

layBlueEgg

Move

turnLeft

...

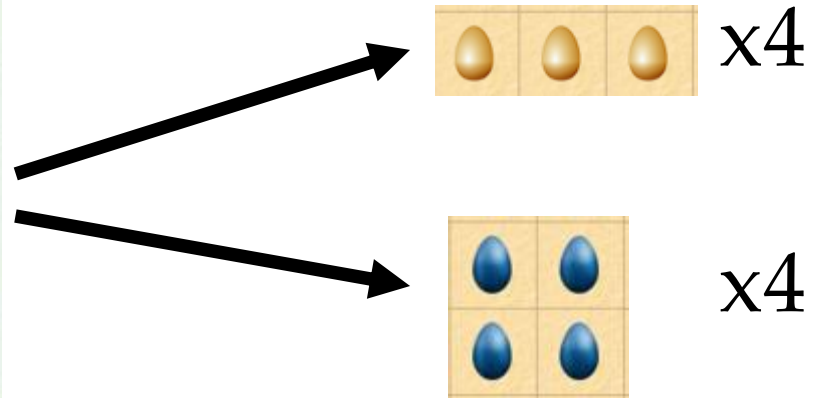
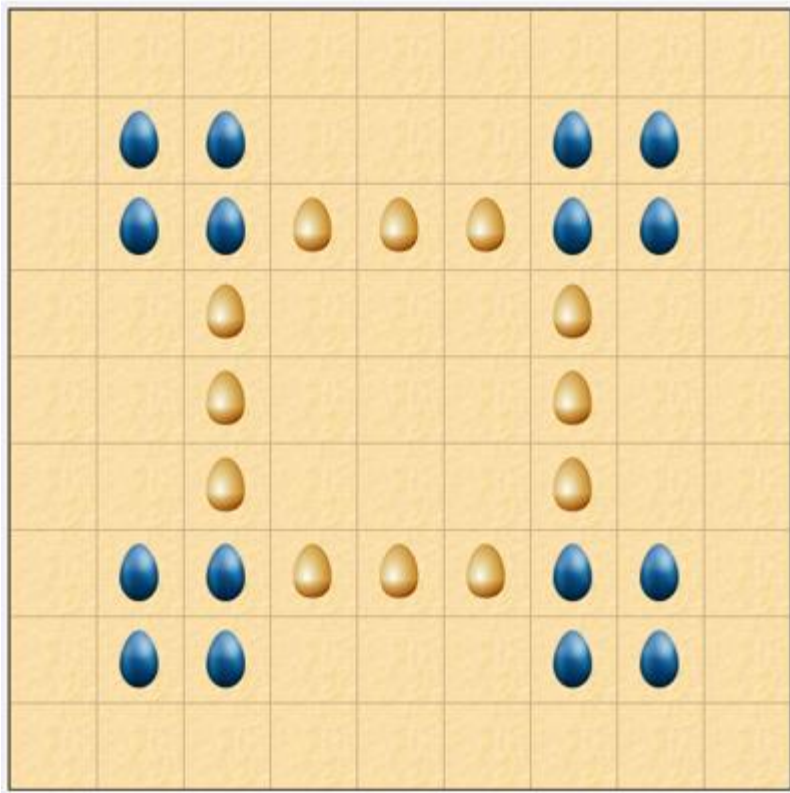
move

layGoldenEgg

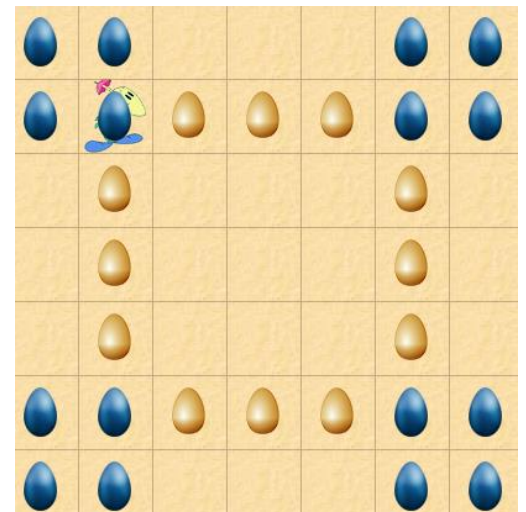
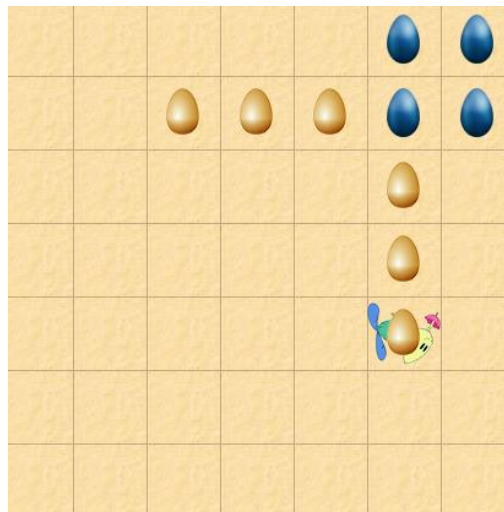
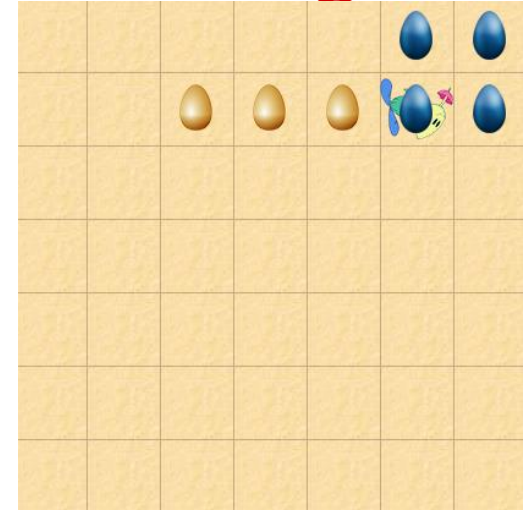
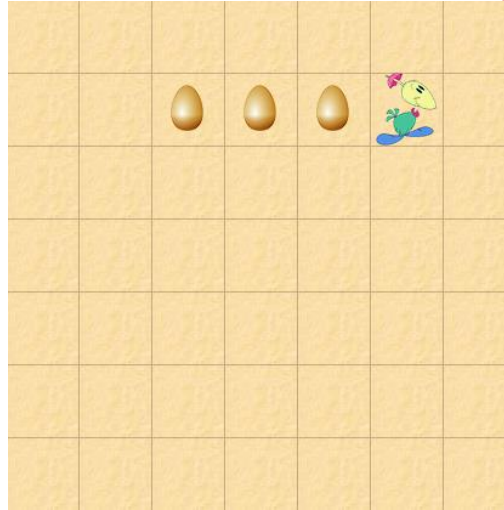
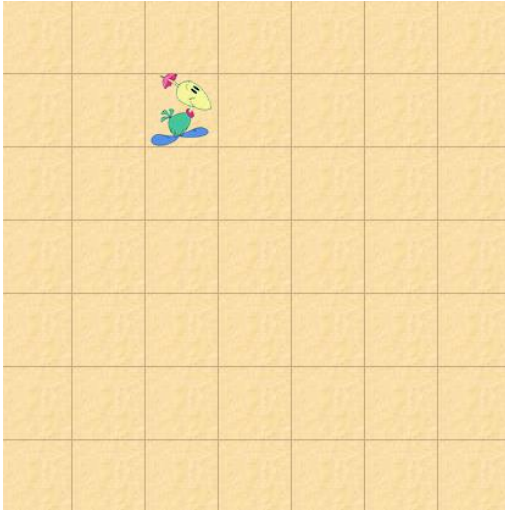
move

...

Decompositie



Stel: Gewenste oplossing



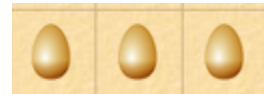
Submethodes schakelen

Submethodes:

- `layBlueSquareEggs`



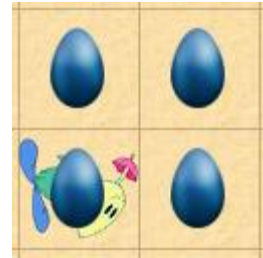
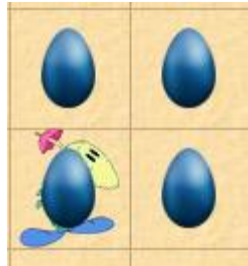
- `layGoldLineEggs`



Wat moet je weten om deze achter elkaar aan te kunnen roepen?

Submethodes schakelen

Benodigde kennis:



Situatie na aanroep `layBlueSquareEggs( )`

Klaar om verder te gaan?

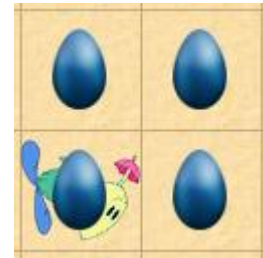
Of

Zelfde kijkrichting als oorspronkelijk?

Submethodes schakelen

Situatie na aanroep `layBlueSquareEggs( )`

Klaar om verder te gaan?



Of

Zelfde kijkrichting als oorspronkelijk?

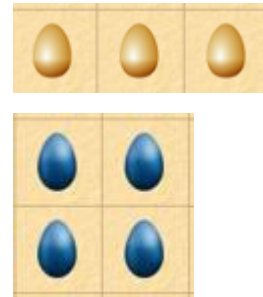


Welke maakt niet uit, maar maak een KEUZE en beschrijf deze!

Oplossing mbv decompositie

4x herhalen:

- layGoldLineEggs
- layBlueSquareEggs



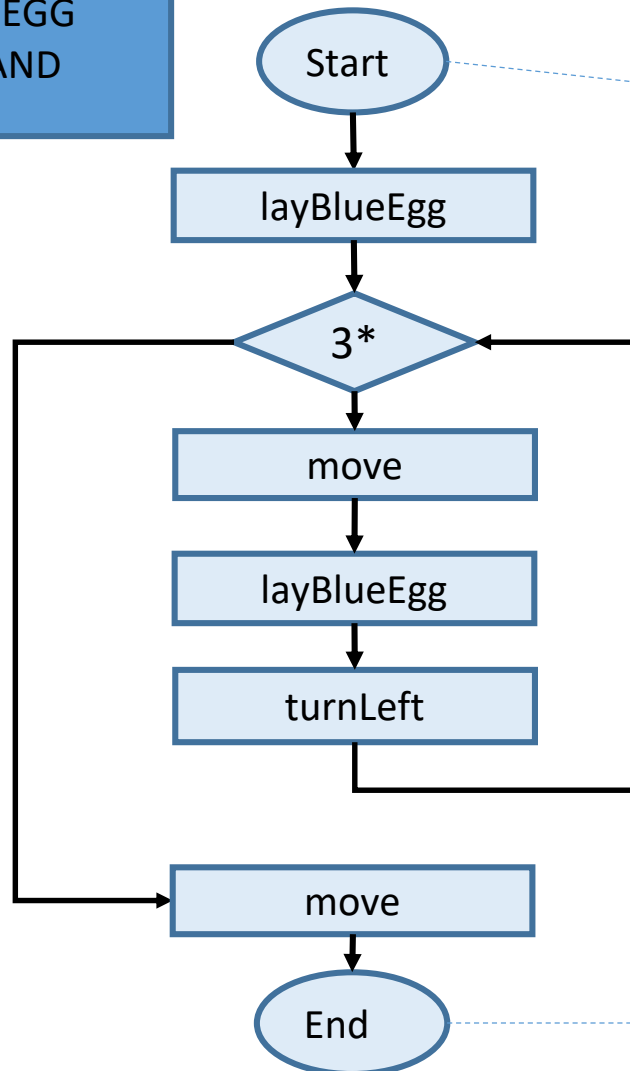
Opeenvolging van submethodes

- Je moet kunnen redeneren over gedrag van submethodes:
 - Zinnige naamgeving
 - Javadoc
 - Maar ook: afspraken over begin & eindtoestand
- Goede gewoontes: maken hergebruik en grotere/complexere programma's mogelijk, schelen uiteindelijk tijd en frustratie (debugging)

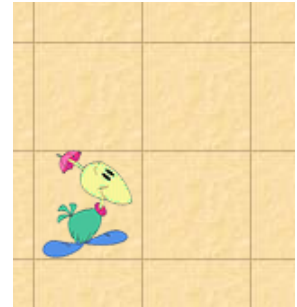


Mutator methods (behavior)

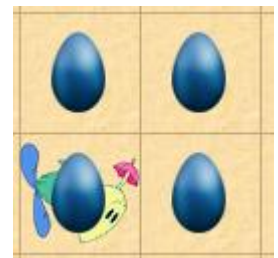
LAY BLUE EGG
SQUARE AND
TURN



Initial situation: Dodo in world, at least 2 sq away from border in front and on left



Final situation:
Dodo same location as
initial, turned right, laid
square of eggs



Aan de slag met lesmateriaal

- Zie blad: materiaal verkenning

Algorithmisch Denken



Uitdaging: Modderig dorp

Probleem:

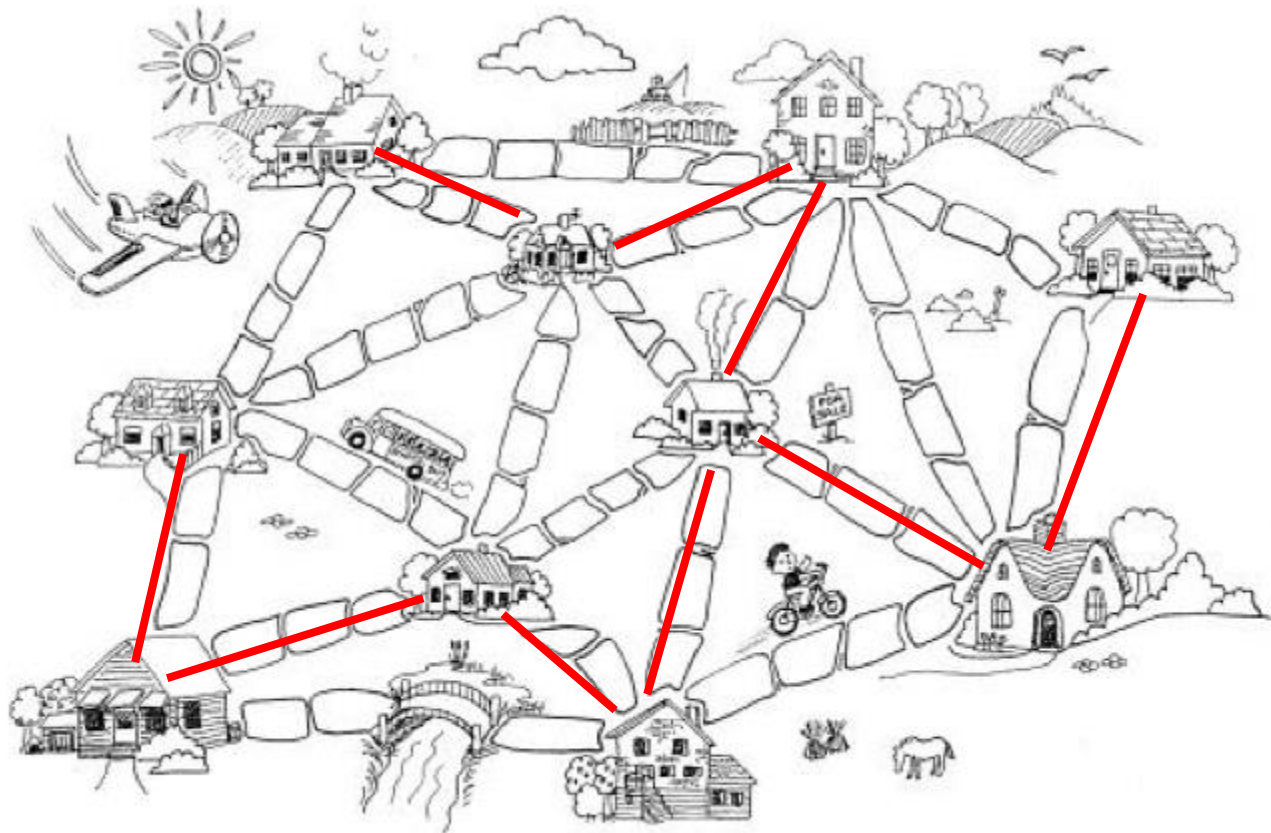
- Dorp heeft geen straten
- Regen? Modderige schoenen!
- Beperkte financiën: burgemeester wil ook zwembad bouwen

Uitdaging:

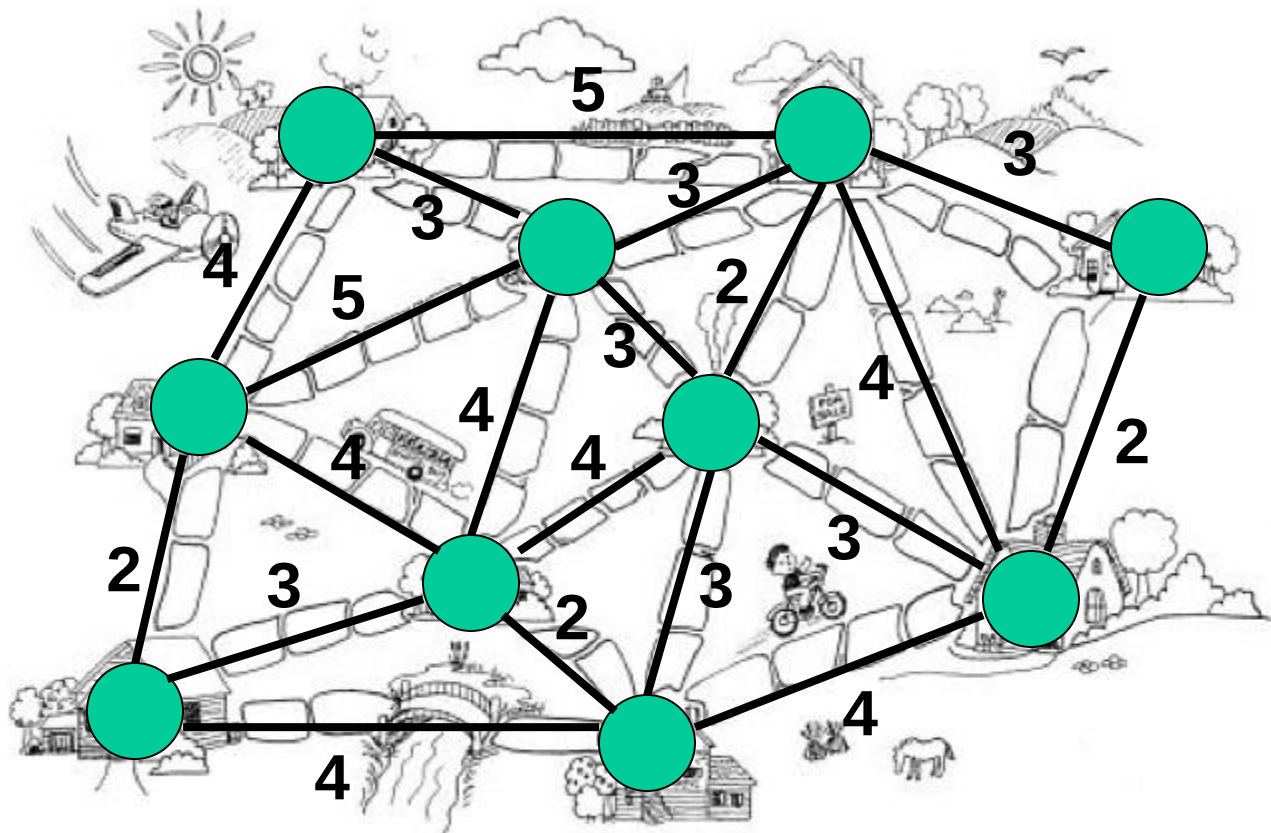
- Asfalteer sommige straten
- Net genoeg zodat iedereen overal kan komen
- Zo goedkoop mogelijk
 - Asfalteringskosten afhankelijk van lengte

Jouw oplossing

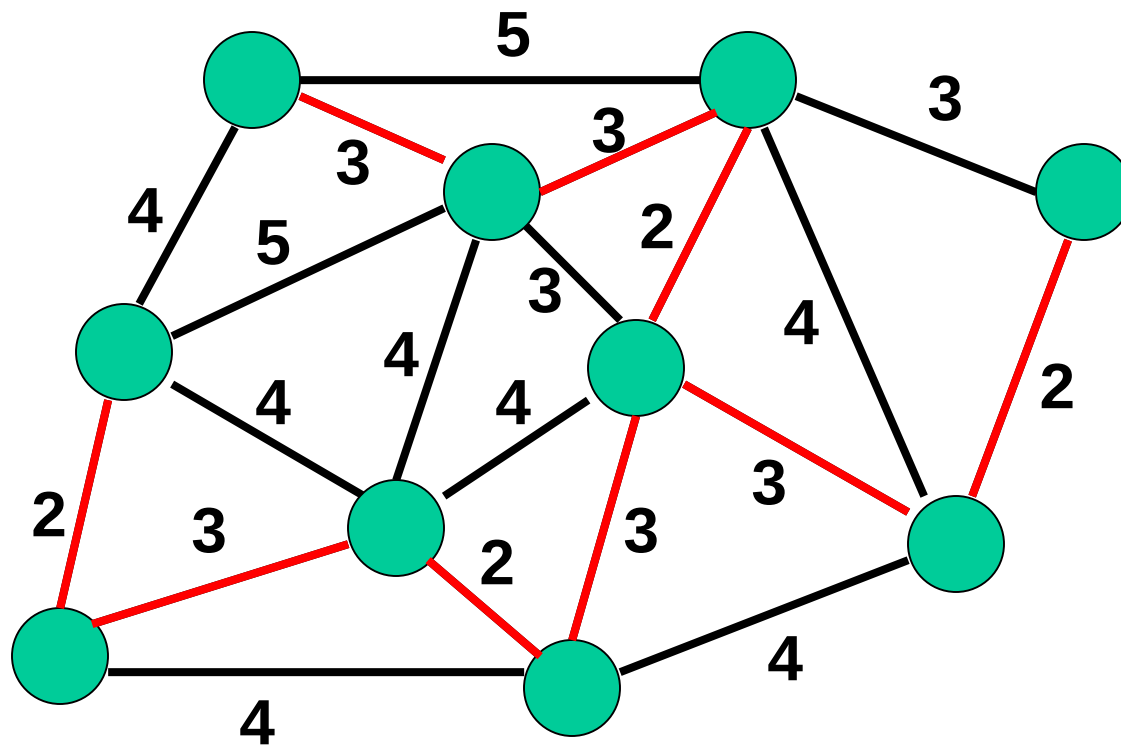
- Wat heb je bedacht?
- Hoe veel gaat het kosten?
- Wat was jouw strategie?



Graaf als abstractiemiddel



Een oplossing als graaf



Verskillende strategieën

Strategie 1: Dure straten elimineren

- Reverse-delete algorithm

Strategy 2: Alle mogelijkheden berekenen

- Brute force
- Heel erg veel werk

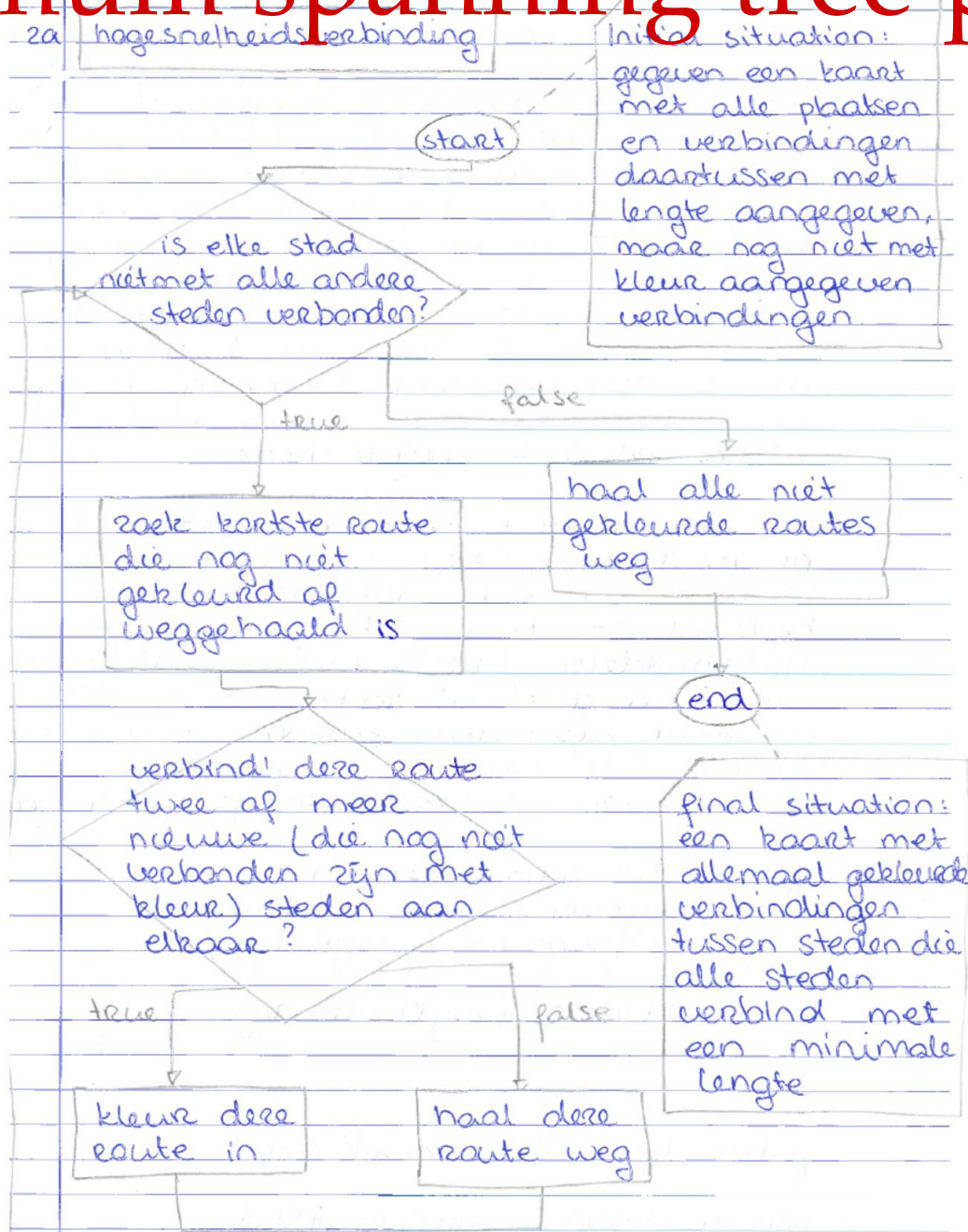
Strategy 3: Goedkoopste intekenen

- Kruskal (gretig)
- Efficiënte algoritme

Waar het om gaat

- **Minimal spanning tree** problem
 - Alle knopen verbinden
 - Minimale totale lengte
- Kaart/graaf maken: abstractie
- Efficiënte algoritmes bestaan
- ‘Echte wereld’ netwerken:
 - Elektriciteit en gas leidingen
 - Computer en telefoon

Minimum spanning tree problem



Masterclass voor leerlingen

- Cursus Computational Thinking
- Doelgroep: 4/5 Vwo/Havo (met wisB)
- Start: 6 april (8 x op donderdag middag)
- Kosten: gratis

- Aanmelden en info:

<http://www.ru.nl/pucofscience/scholieren/activiteiten/masterclasses-1617/>

Opdracht voor 23 maart:

- Sorteren in de klas
- Materiaal verkenning
 - zie lijst met opgaven
 - 1x reflectie formulier naar keuze

Vooruitblik naar 23 maart

- Terugkoppeling
 - Sorteren in de klas
 - Opdrachten (knelpunten/vragen/ervaringen)
- Verdieping in algoritmisch denken en lesmateriaal:
 - Verder met een Greenfoot module
 - AD oefenopdracht / Unplugged
- Neem een laptop mee

Vragen? Opmerkingen?



Tot de volgende keer!

- ma 23 maart
 - inloop vanaf 15.30 uur
 - start 16.00 uur tot 20.00 uur