

# Die drei ???<sup>®</sup> und die Kammer der Rätsel

190



EUROPA

<https://www.cd-hoerspiele.de/drei-fragezeichen-190.html>

Christian Köppe, Jacqueline Nijenhuis, Jacco Gnodde, Renske Smetsers-Weeda

# KEUZEMODULE ALGORITMIEK

# Die drei ???<sup>®</sup> und die Kammer der Rätsel

190



# Keuzethema Algoritmiek, berekenbaarheid en logica

## ***Complexiteit van algoritmen***

De kandidaat kan

**(havo:)** van gegeven algoritmen de complexiteit vergelijken, en kan klassieke ‘moeilijke’ problemen herkennen en benoemen.

**(vwo:)** het verschil tussen exponentiële en polynomiale complexiteit uitleggen, kan algoritmen op basis hiervan onderscheiden, en kan klassieke ‘moeilijke’ problemen herkennen en benoemen.

## ***Berekenbaarheid***

De kandidaat kan berekeningen op verschillende abstractieniveaus karakteriseren en relateren, en kan klassieke *onberekenbare* problemen herkennen en benoemen.

## ***Logica***

De kandidaat kan eigenschappen van digitale artefacten uitdrukken in logische formules.

# Keuzethema Algoritmiek, berekenbaarheid en logica

## **Complexiteit van algoritmen**

De kandidaat kan

(havo:) van gegeven algoritmen de complexiteit vergelijken, en kan klassieke 'moeilijke' problemen herkennen en benoemen.

(vwo:) het verschil tussen exponentiële en polynomiale complexiteit uitleggen, kan algoritmen op basis hiervan onderscheiden, en kan klassieke 'moeilijke' problemen herkennen en benoemen.

## **Berekenbaarheid**

De kandidaat kan berekeningen op verschillende abstractieniveaus karakteriseren en relateren, en kan klassieke *onberekenbare* problemen herkennen en benoemen.

## **Logica**

De kandidaat kan eigenschappen van digitale artefacten uitdrukken in logische formules.

*Voorbeelden van problemen voor het karakteriseren van berekeningen op verschillende niveaus zijn: fundamenteel-conceptueel, eindige automaten, Turingmachines, berekenbaarheid, machinerealist, digitale schakelingen, machinerealist, en instructies in assemblertaal en hogere programmeertalen. Deze voorbeelden dienen niet als een lijst te denken.*

# Waarop kunnen we voortbouwen?

## ■ Kernprogramma!

### **Algoritmen**

De kandidaat kan

- een oplossingsrichting voor een probleem uitwerken tot een algoritme;
- daarbij standaardalgoritmen herkennen en gebruiken;
- de correctheid en efficiëntie van digitale artefacten onderzoeken via de achterliggende algoritmen.

### **Datastructuren**

De kandidaat kan verschillende abstracte datastructuren met elkaar vergelijken op en efficiëntie.

### **Automaten**

De kandidaat kan eindige automaten gebruiken voor de karakterisering van bepaald algoritmen.

### **Grammatica's**

De kandidaat kan grammatica's hanteren als hulpmiddel bij de beschrijving van talen.



### **Doelstellingen**

De kandidaat kan doelstellingen voor informatie- en gegevensverwerking onderscheiden, waaronder *zoeken* en *bewerken*.

### **Identificeren**

De kandidaat kan informatie en gegevens identificeren in contexten, daarbij rekening houdend met de doelstelling.

### **Representeren**

De kandidaat kan gegevens representeren in een geschikte datastructuur, daarbij rekening houdend met de doelstelling, en kan daarbij verschillende representaties met elkaar vergelijken op elegantie, efficiëntie en implementeerbaarheid.

### **Standaardrepresentaties**

De kandidaat kan standaardrepresentaties van numerieke gegevens en media gebruiken en aan elkaar relateren.

### **Gestructureerde data**

De kandidaat kan een informatiebehoefte vertalen in een zoekopdracht op een verzameling gestructureerde data.

### **Ontwikkelen**

De kandidaat kan, voor een gegeven doelstelling,

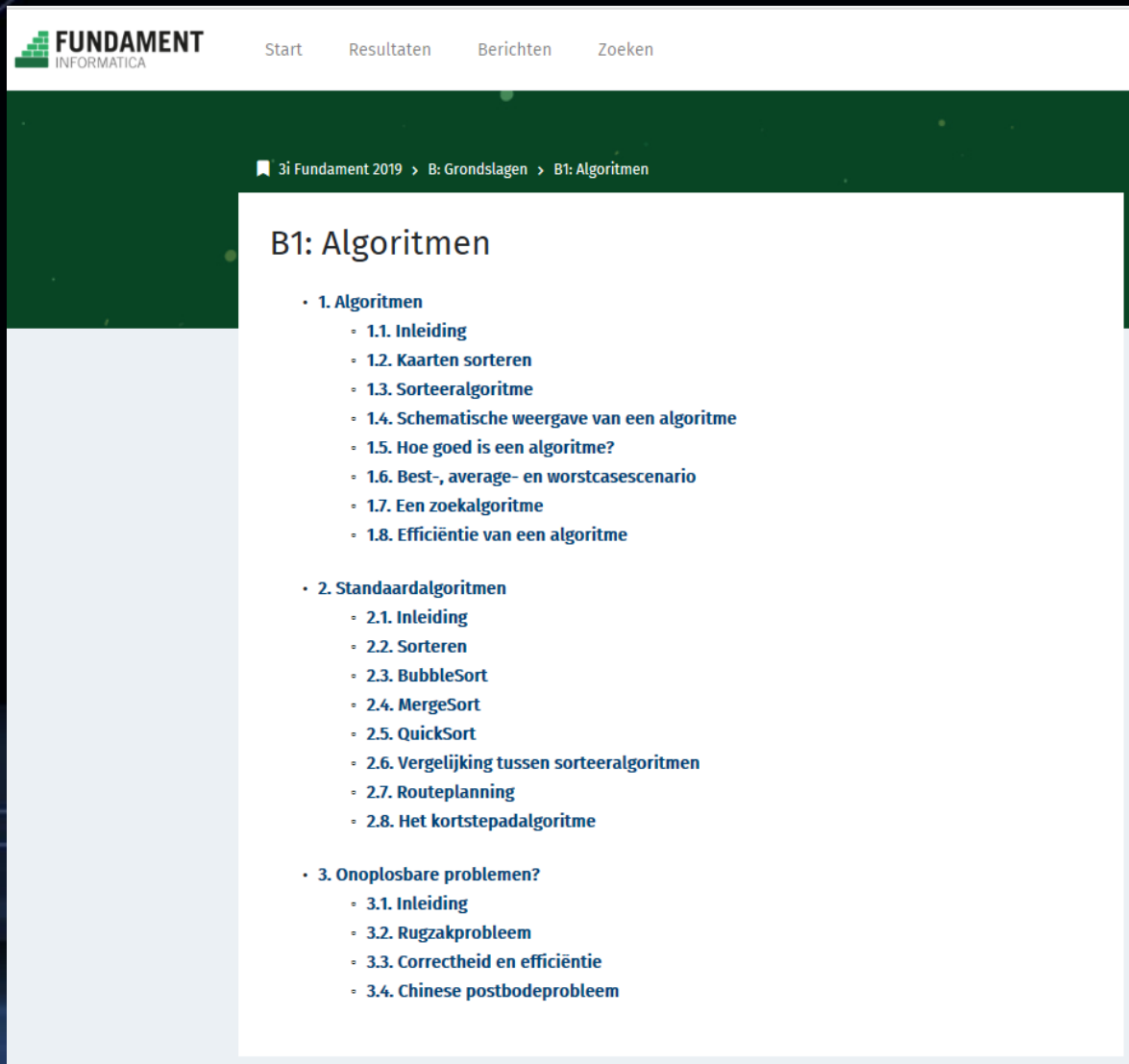
- programmacomponenten ontwikkelen in een imperatieve programmeertaal;
- daarbij programmeertaalconstructies gebruiken die abstractie ondersteunen;
- programmacomponenten zodanig structureren dat ze door anderen gemakkelijk te begrijpen en te evalueren zijn.

### **Inspecteren en aanpassen**

De kandidaat kan

- structuur en werking van gegeven programmacomponenten uitleggen;
- zulke programmacomponenten aanpassen op basis van evaluatie of veranderde eisen.

# Lesmethodes (Instruct, Informatica-actief)



The screenshot shows the website for 'FUNDAMENT INFORMATICA'. The navigation bar includes 'Start', 'Resultaten', 'Berichten', and 'Zoeken'. The breadcrumb trail is '3i Fundament 2019 > B: Grondslagen > B1: Algoritmen'. The main heading is 'B1: Algoritmen'. Below it is a table of contents with three main sections: '1. Algoritmen', '2. Standaardalgoritmen', and '3. Onoplosbare problemen?'. Each section contains a list of sub-topics.

**FUNDAMENT**  
INFORMATICA

Start Resultaten Berichten Zoeken

3i Fundament 2019 > B: Grondslagen > B1: Algoritmen

## B1: Algoritmen

- 1. Algoritmen
  - 1.1. Inleiding
  - 1.2. Kaarten sorteren
  - 1.3. Sorteeralgoritme
  - 1.4. Schematische weergave van een algoritme
  - 1.5. Hoe goed is een algoritme?
  - 1.6. Best-, average- en worstcasescenario
  - 1.7. Een zoekalgoritme
  - 1.8. Efficiëntie van een algoritme
- 2. Standaardalgoritmen
  - 2.1. Inleiding
  - 2.2. Sorteren
  - 2.3. BubbleSort
  - 2.4. MergeSort
  - 2.5. QuickSort
  - 2.6. Vergelijking tussen sorteeralgoritmen
  - 2.7. Routeplanning
  - 2.8. Het kortstepadalgoritme
- 3. Onoplosbare problemen?
  - 3.1. Inleiding
  - 3.2. Rugzakprobleem
  - 3.3. Correctheid en efficiëntie
  - 3.4. Chinese postbodeprobleem

# Basic ideas (feedback welkom!)

- Een korte intro/herhaling basis algoritmen
  - Optioneel, afhankelijk van voorkennis
  - Incl. zoek-/sorteer algoritmen
- Vooral reflecteren op/evalueren van algoritmen
- Complexiteit/berekenbaarheid 2/3, logica 1/3
- “Explorerende” didactiek
- Naast veel praktische oefeningen ook **duidelijke** focus op concepten
- Doelgroep: H6, V5 en V6

# Onderwerpen algoritmiek en berekenbaarheid

- Selectie, sequentie, iteratie
- Recursie
- Brute force/backtracking/greedy
- Correctheid/efficiëntie
- Minimaal opspannende bomen
- Kortste pad
- Dominerende sets
- Travelling salesman
- Convex Hull
- Zoeken/sorteren
- Datarepresentatie (oa hashing)
- Optioneel: halting probleem



# Onderwerpen logica

- Expressies mbv. waarheidstabellen interpreteren
- Equivalenties
- Impliceert-relatie, als-dan
- Verzamelingenleer, venn-diagram (omzetten naar expressie)
- Problemen oplossen met logisch nadenken/redeneren

# Constructive Alignment?!

- Leerdoelen -> Assessment criteria/methoden -> Inhoud/opzet
- Geen voorbeelden/richtlijnen gevonden...
- Dus...

# Hergebruik (aka of course not invented here)!

- Nieuw-Zeeland

- <http://csfieldguide.org.nz/en/curriculum-guides/apcsp/guide-algorithms-introduction.html>
- <http://csfieldguide.org.nz/en/chapters/complexity-tractability.html>

- Duitsland

- <https://www.inf-schule.de/algorithmen/algorithmen/effizienz>

- VS

- <https://sites.google.com/a/jcu.edu/mt513/resources/cs-principles-key-ideas-and-learning-out>

# Nog te bepalen (1)

- Concrete leerdoelen
- Toetscriteria
- Toetsvormen
- Gebruikte tools
- Gebruikte notatie/s

## Nog te bepalen (2)

- Datastructuren?
- Is er ook een vertaalslag naar realisatie/context (of alleen “droogzwemmen”)?
- Welke context/en (passend bij concepten én belevingswereld van leerlingen)?
- Keuze aan specifieke onderwerpen.
- Concrete didactische aanpakken

Zeker een toevoeging:



<https://www.flickr.com/photos/21561428@N03/5076724542>