

# Inpakken !

Hendrik Jan Hoogeboom | Proefstudereren

29-11-2019



Universiteit  
Leiden

Bij ons leer je de wereld kennen

# Informatica

Bachelor

2019-2020 ▾



De andere specialisaties van de bachelor Informatica vind u hier: [Informatica & Economie](#) [Bioinformatica](#)  
[Informatica: Variant: Kunstmatige intelligentie](#)

Eerste jaar

Tweede jaar

Derde jaar

Vak

Algoritmiek

Continue Wiskunde 1

Continue Wiskunde 2

Databases

Fundamentele Informatica 1

Fundamentals of Digital Systems Design

Introduction to Logic

Linear Algebra for Computer Scientists 1

Linear Algebra for Computer Scientists 2

Oriëntatie Informatica

Programming Techniques

Studying and Presenting

Programmeermethoden

Vak

Complexiteit

Computerarchitectuur

Concepts of Programming Languages

Datastructuren

Fundamentele Informatica 2

Fundamentele Informatica 3

Kunstmatige Intelligentie

Operating Systemen

Research Methods in Computer Science

Security

Statistics for Computer Scientists

Keuzevak

Vorbereiding Programmeerwedstrijden

semester 2

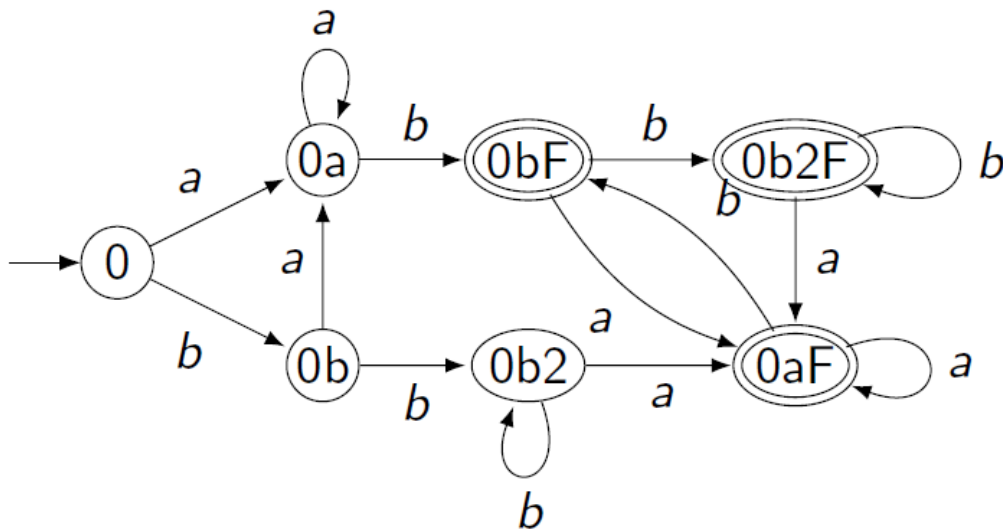
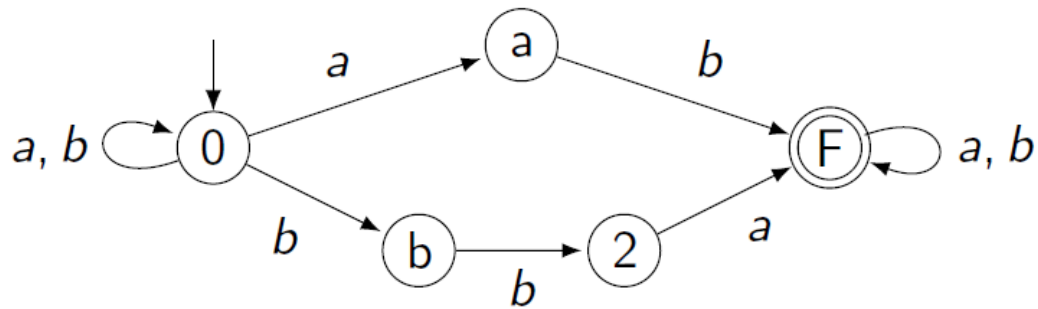
## Specialisaties

Informatica & Economie

Informatica: Dubbele Bachelor  
(Informatica + Wiskunde)

Bioinformatica

# Example: subset construction

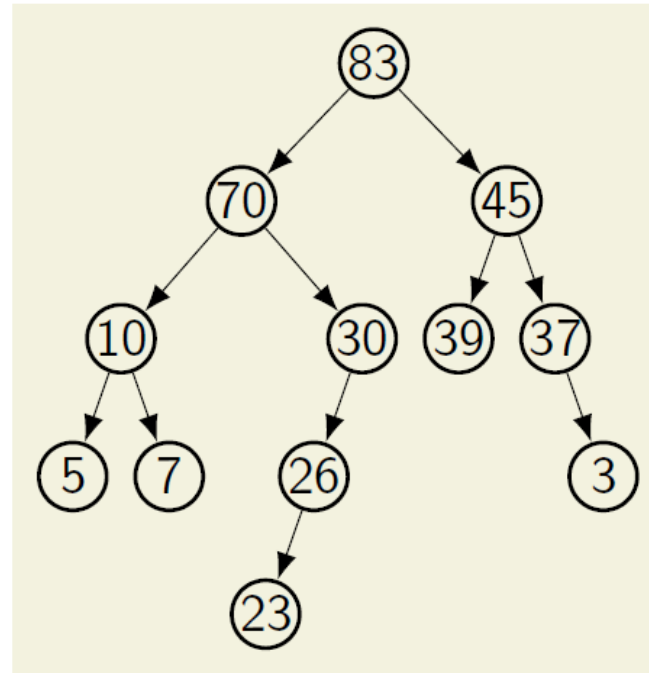
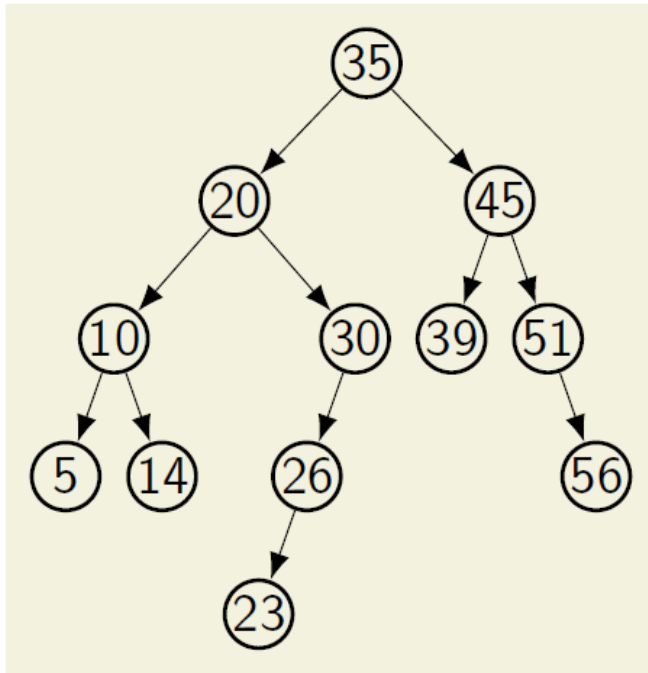


[M] language from  $\leftrightarrow F$  2 18

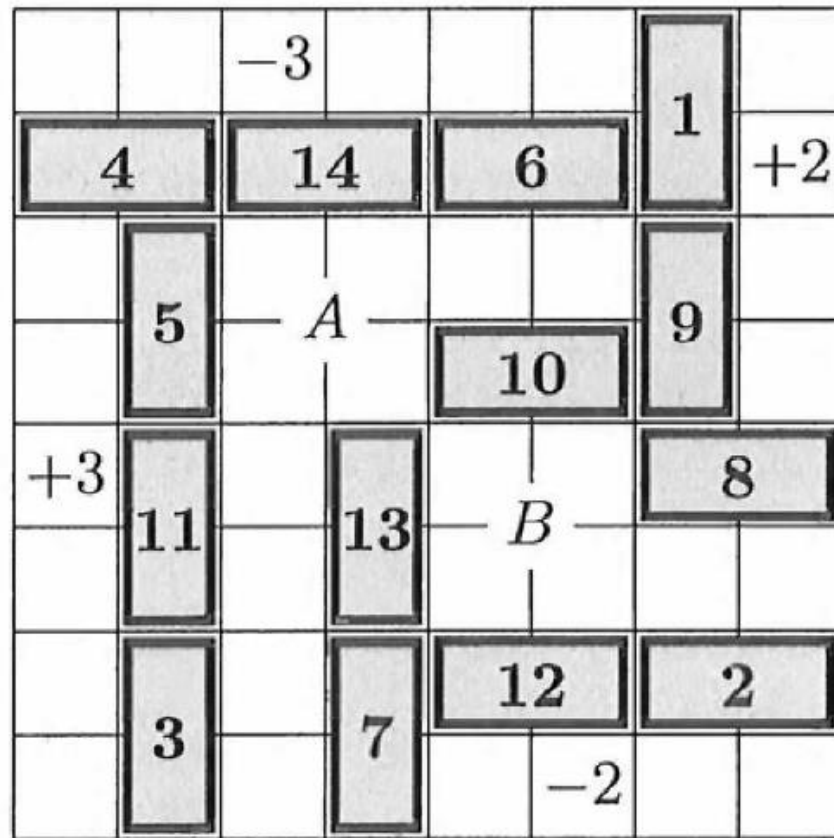
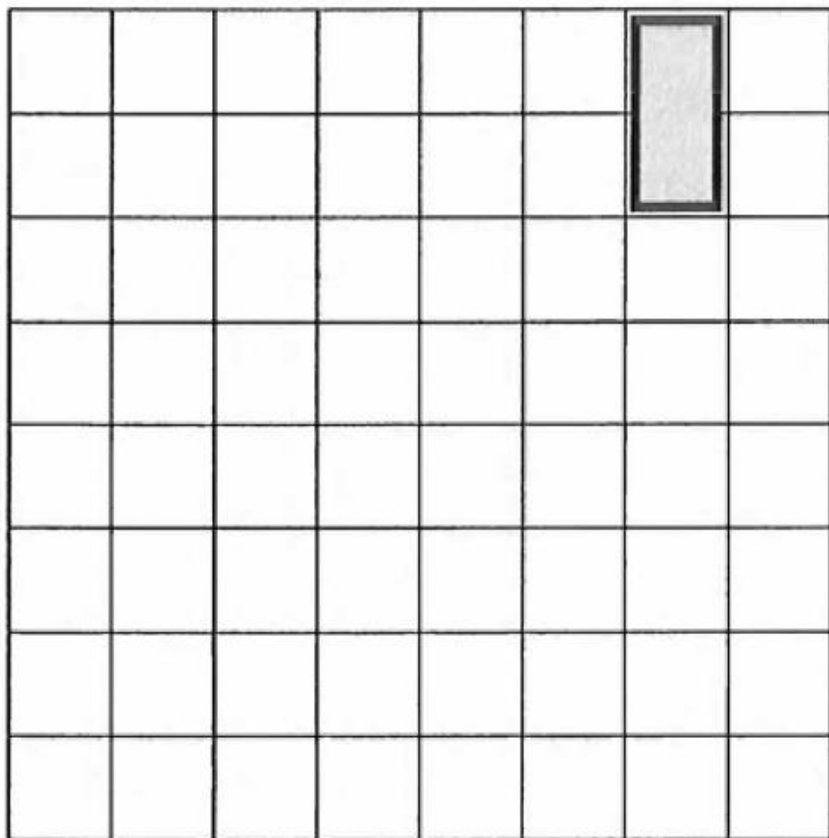
## Fundamentele Informatica 1



# binary search tree vs heap order



Datastructuren / Data Structures



Seminar Algorithms  
 A.Siegel, Combinatorial Games

# Informatica

Bachelor

2019-2020 ▾



De andere specialisaties van de bachelor Informatica vind u hier: [Informatica & Economie](#) [Bioinformatica](#)  
[Informatica: Variant: Kunstmatige intelligentie](#)

Eerste jaar

Tweede jaar

Derde jaar

Vak

Algoritmiek

Continue Wiskunde 1

Continue Wiskunde 2

Databases

Fundamentele Informatica 1

Fundamentals of Digital Systems Design

Introduction to Logic

Linear Algebra for Computer Scientists 1

Linear Algebra for Computer Scientists 2

Oriëntatie Informatica

Programming Techniques

Studying and Presenting

Programmeermethoden

Vak

Complexiteit

Computerarchitectuur

Concepts of Programming Languages

Datastructuren

Fundamentele Informatica 2

Fundamentele Informatica 3

Kunstmatige Intelligentie

Operating Systemen

Research Methods in Computer Science

Security

Statistics for Computer Scientists

Keuzevak

Vorbereiding Programmeerwedstrijden

semester 2

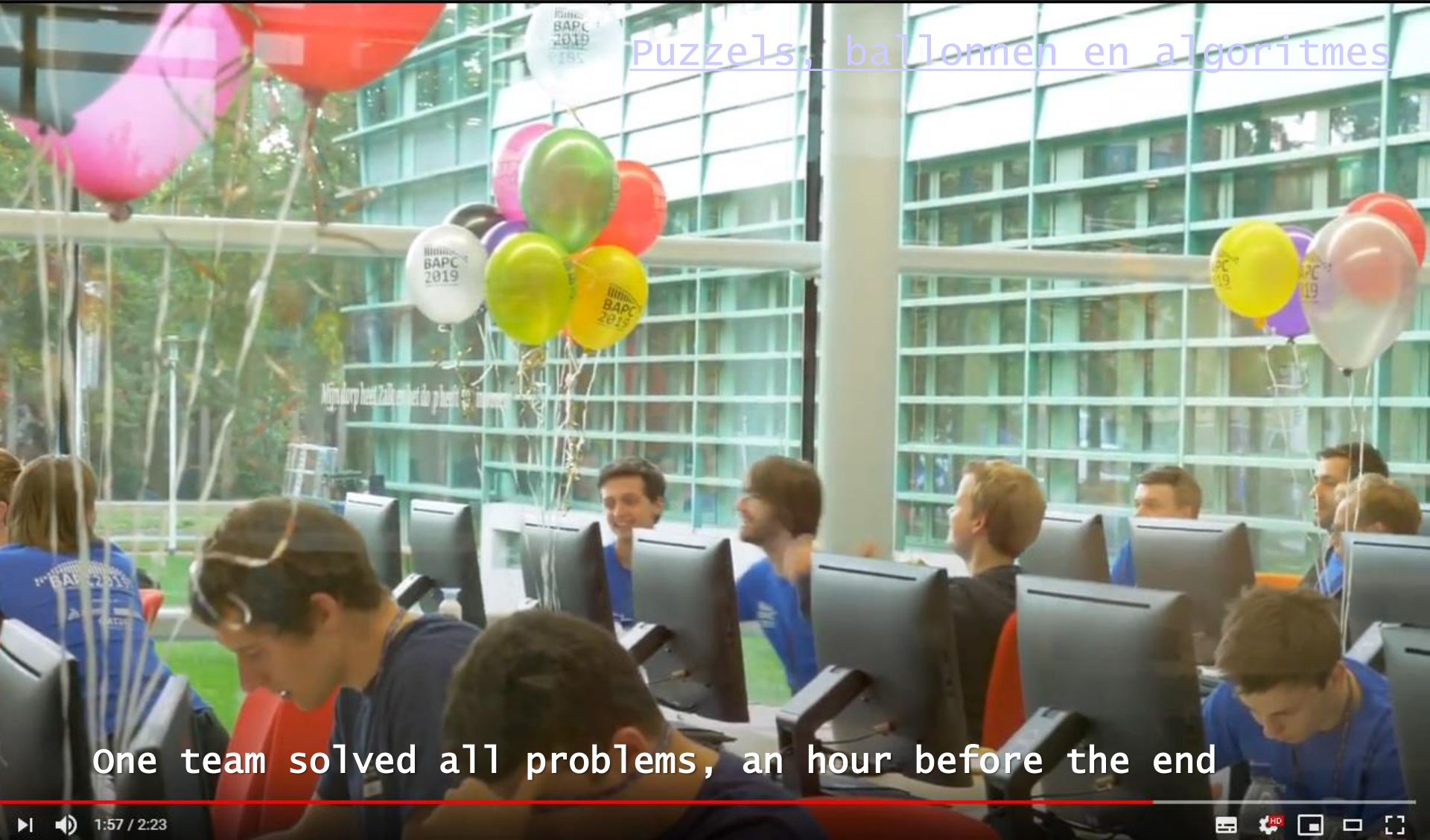
## Specialisaties

Informatica & Economie

Informatica: Dubbele Bachelor  
(Informatica + Wiskunde)

Bioinformatica

Puzzels, ballonnen en algoritmes



One team solved all problems, an hour before the end

# Opgave: wrap 'em up

Christo heeft besloten een hele stad in te pakken; liefst met **zo weinig mogelijk** materiaal.

Schrijf een programma dat de **benodigde hoeveelheid** touw berekent voor **twee-dimensionale steden**. De onderkant wordt niet ingepakt.

**Invoer.** Op elke regel een gebouw, linker kant, rechter kant en hoogte, elk tussen 0 en 10000.

**Uitvoer.** Lengte in drie decimalen.

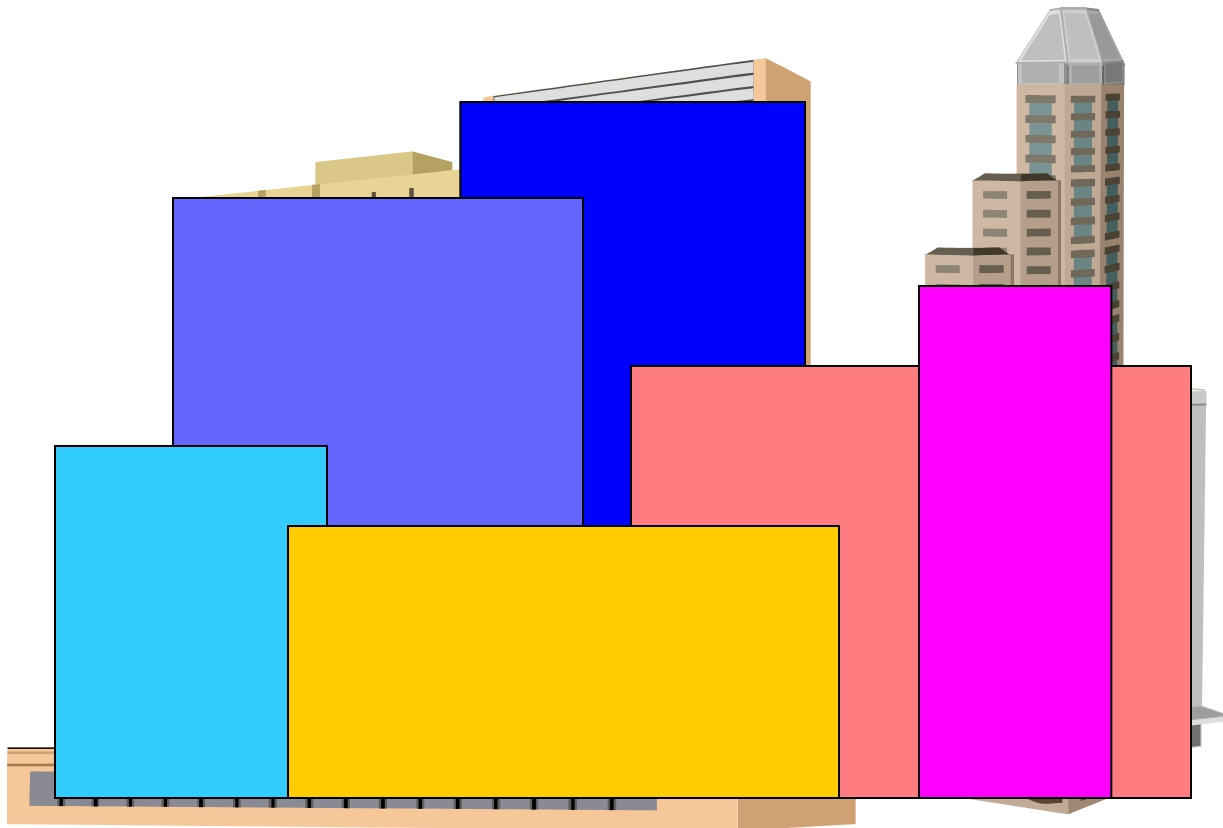




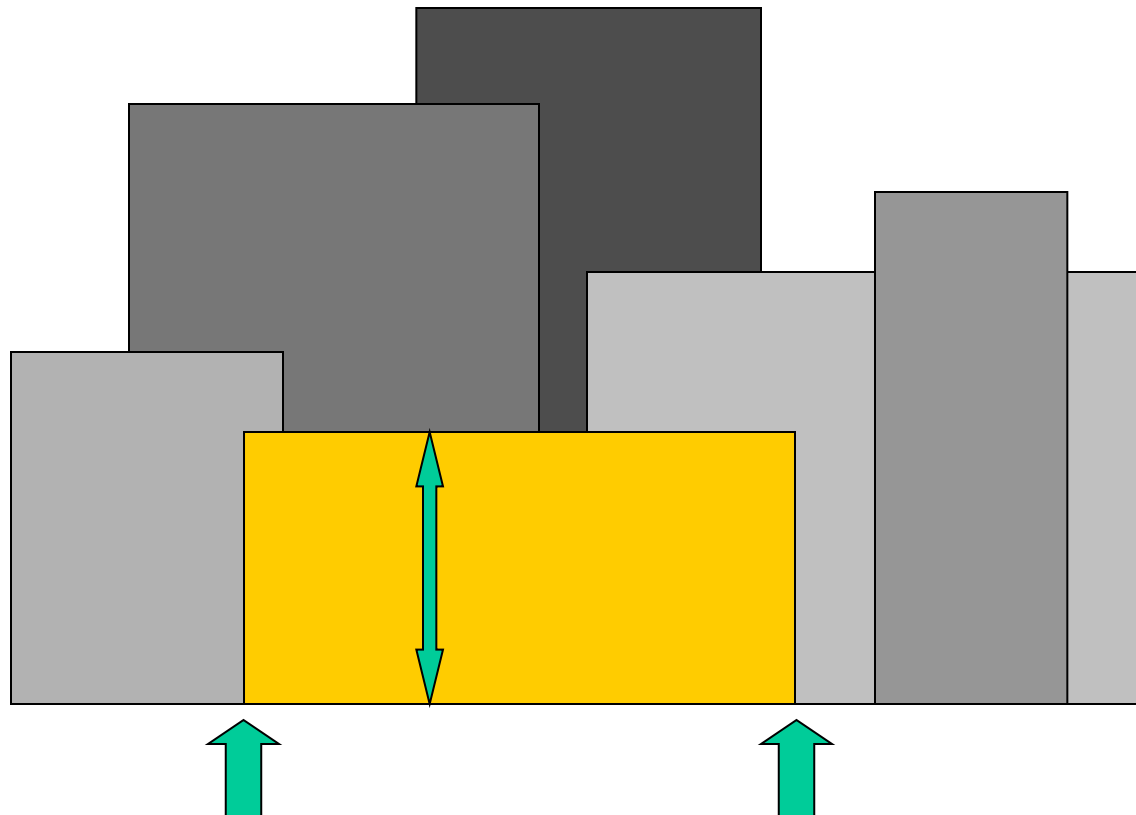
gegevens



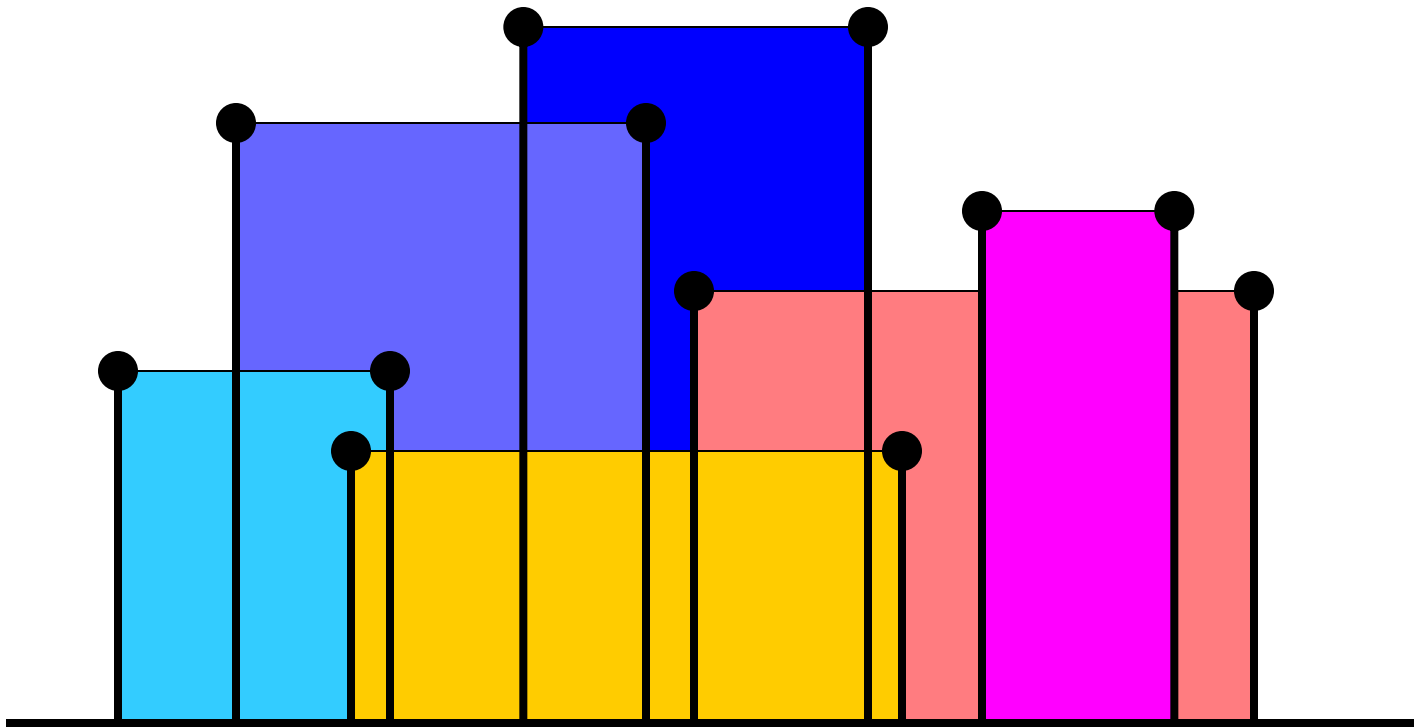
gegevens



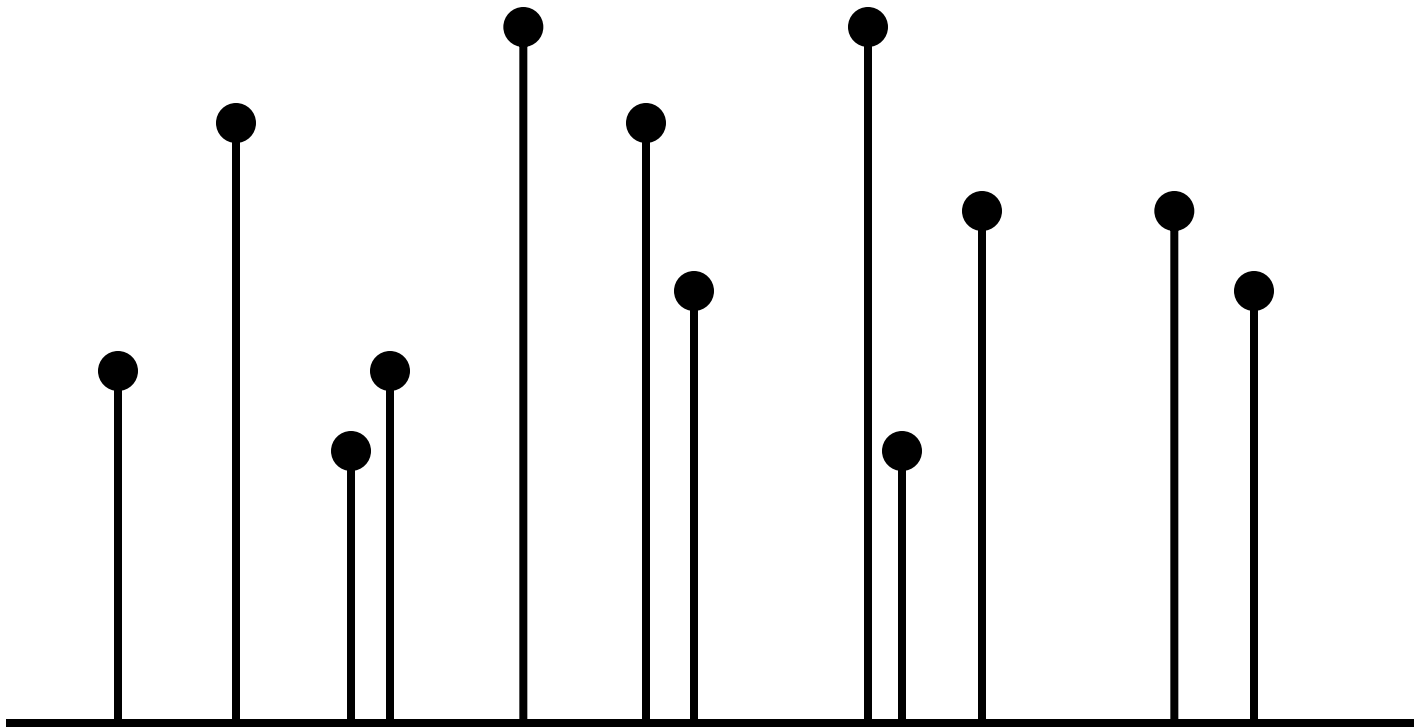
gegevens



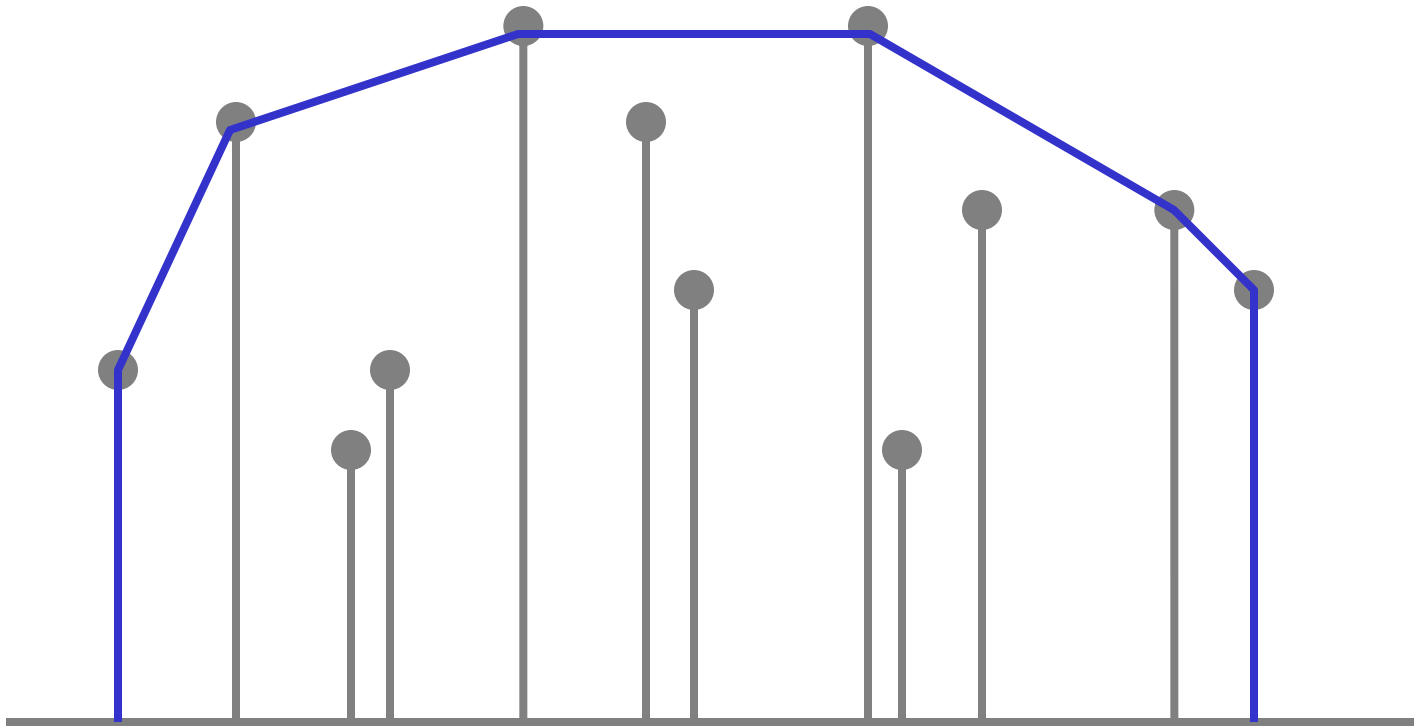
abstractie



data



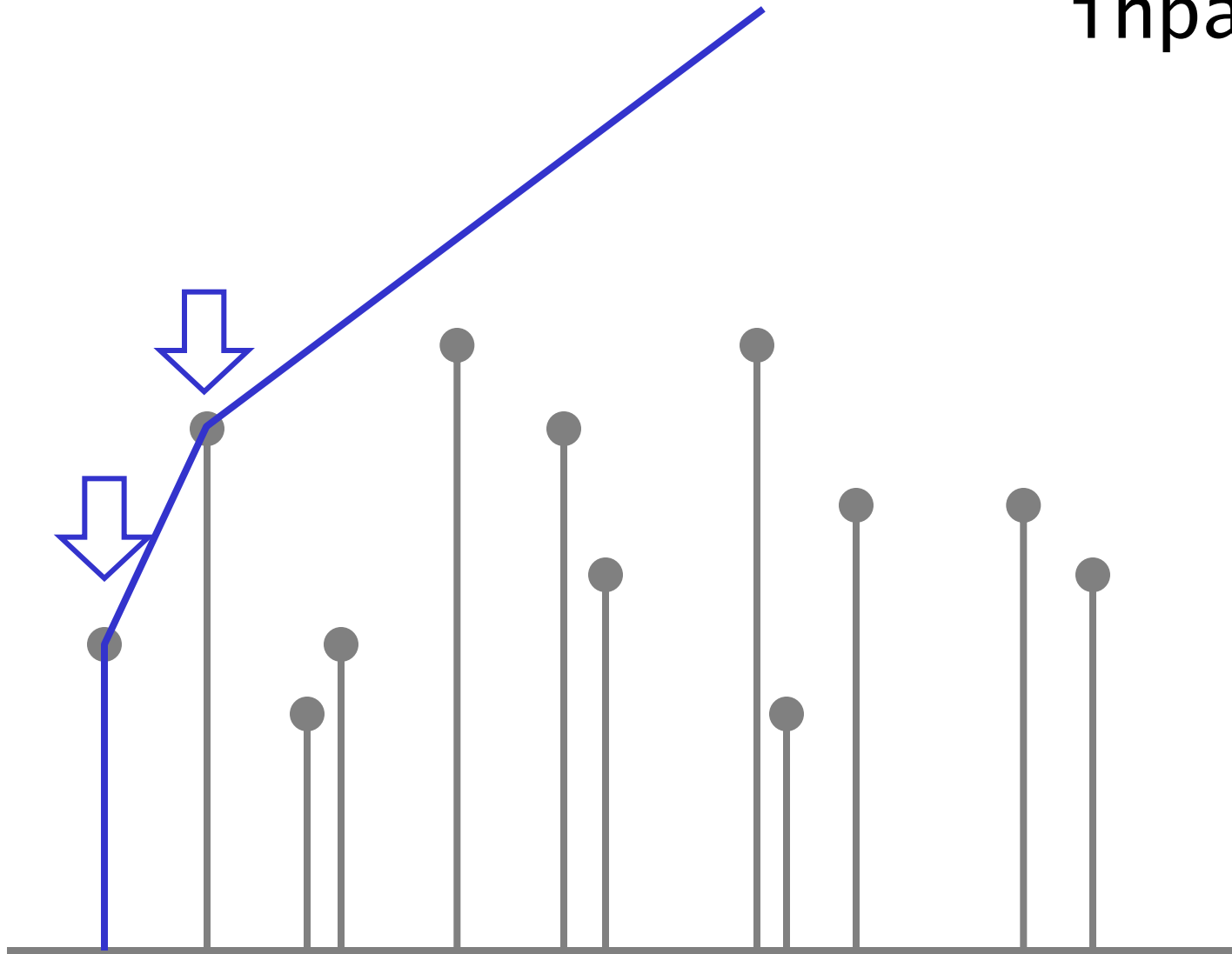
# de oplossing



eerste idee

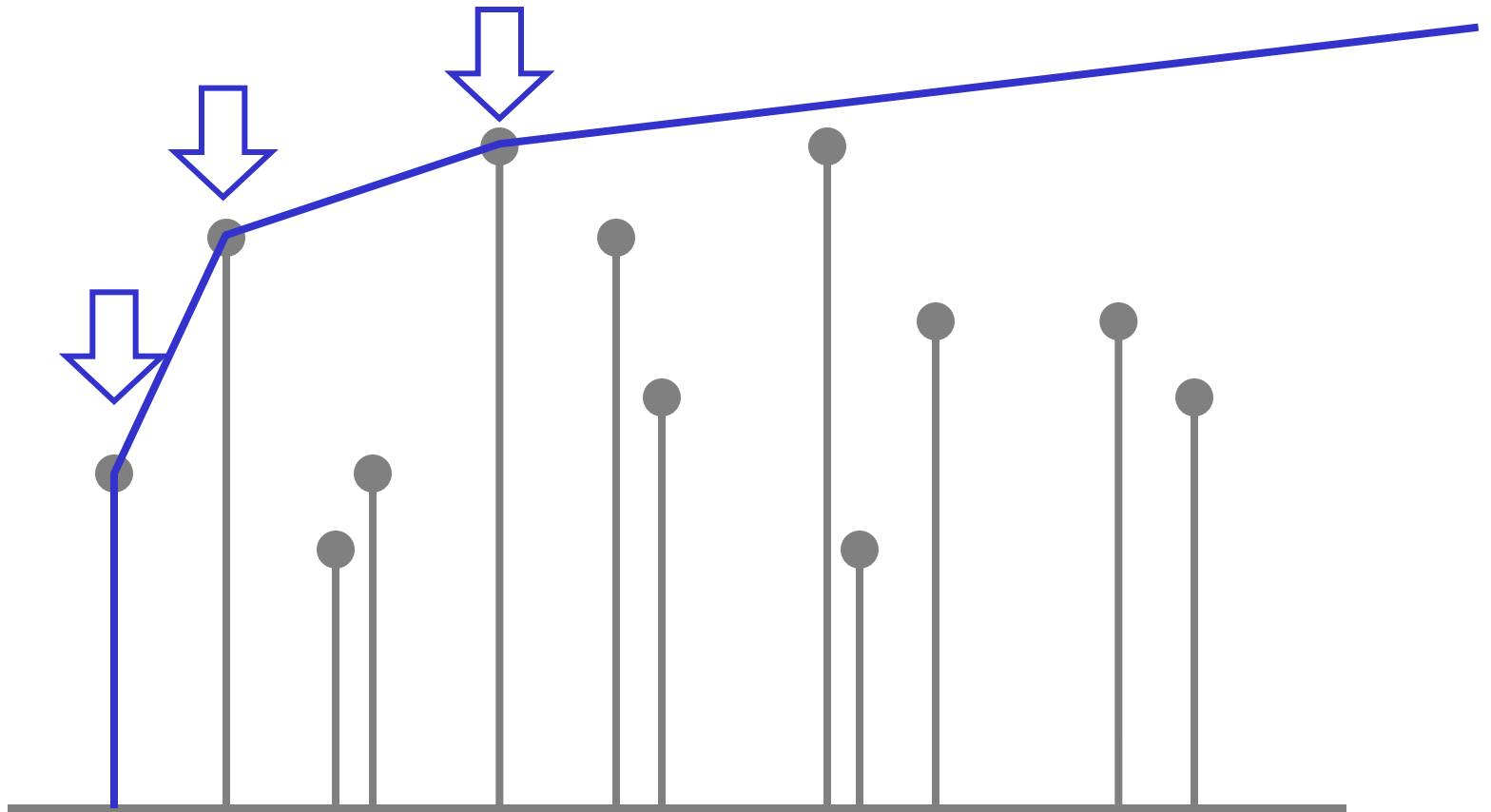


inpakken

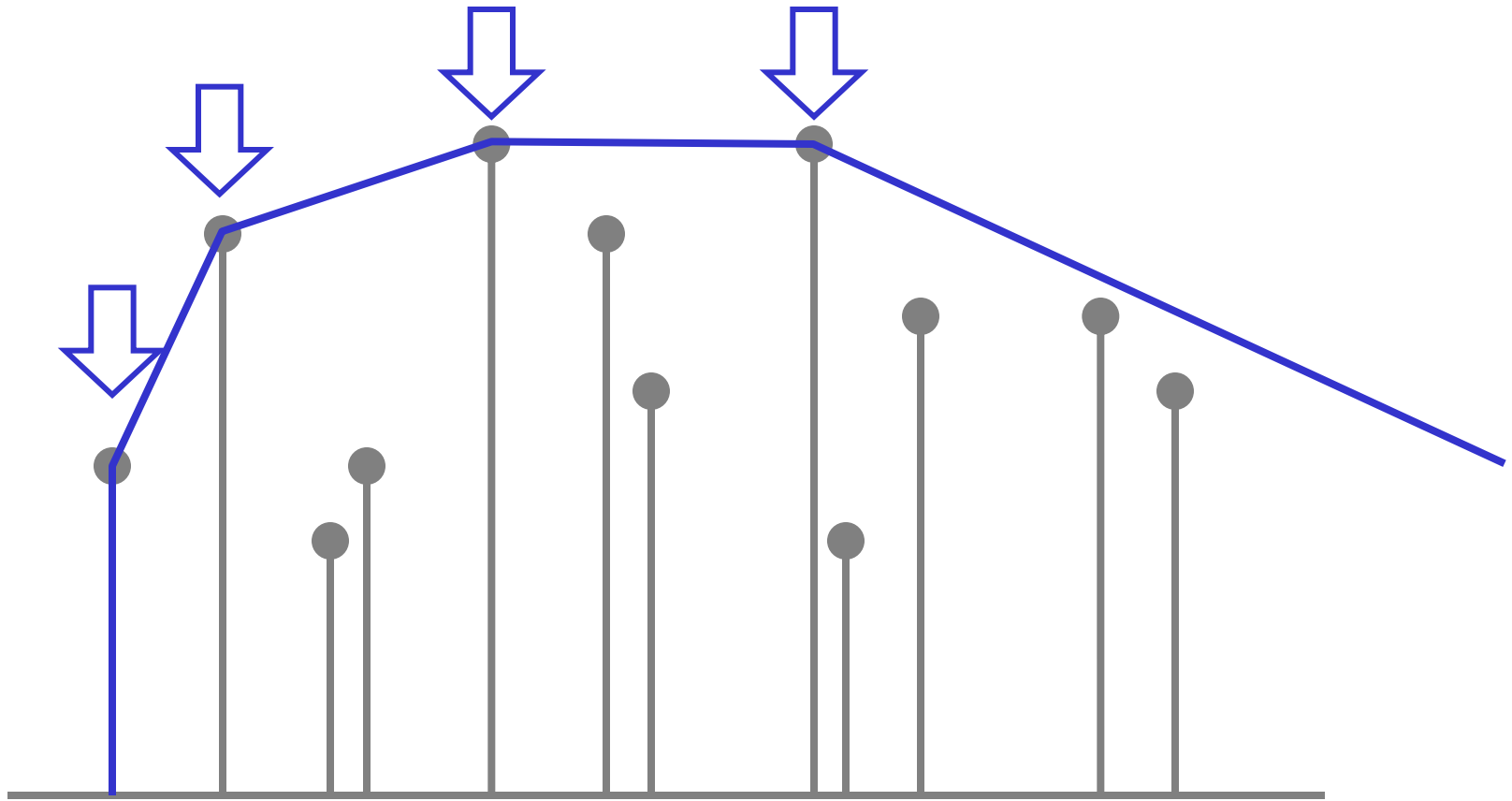




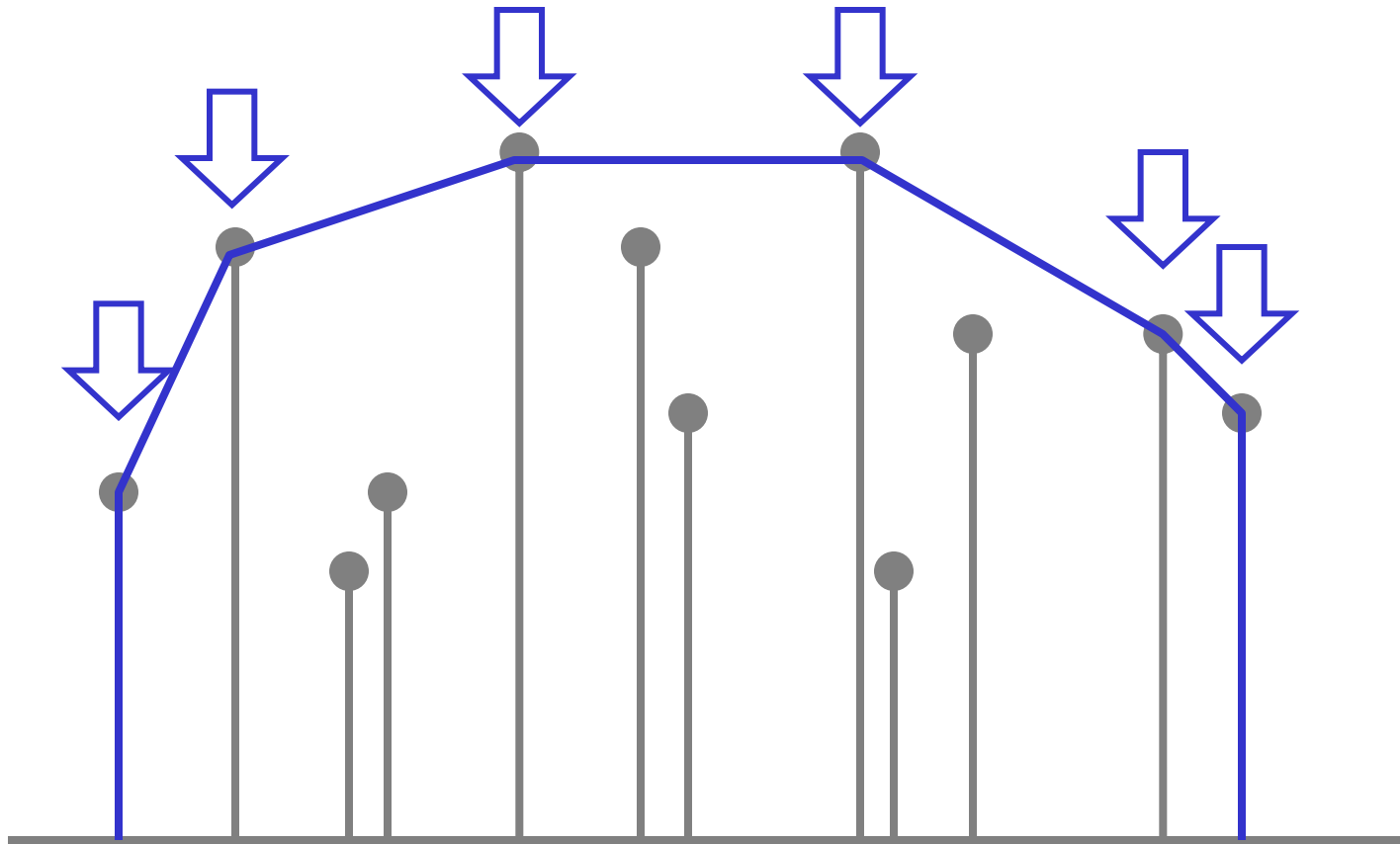
inpakken



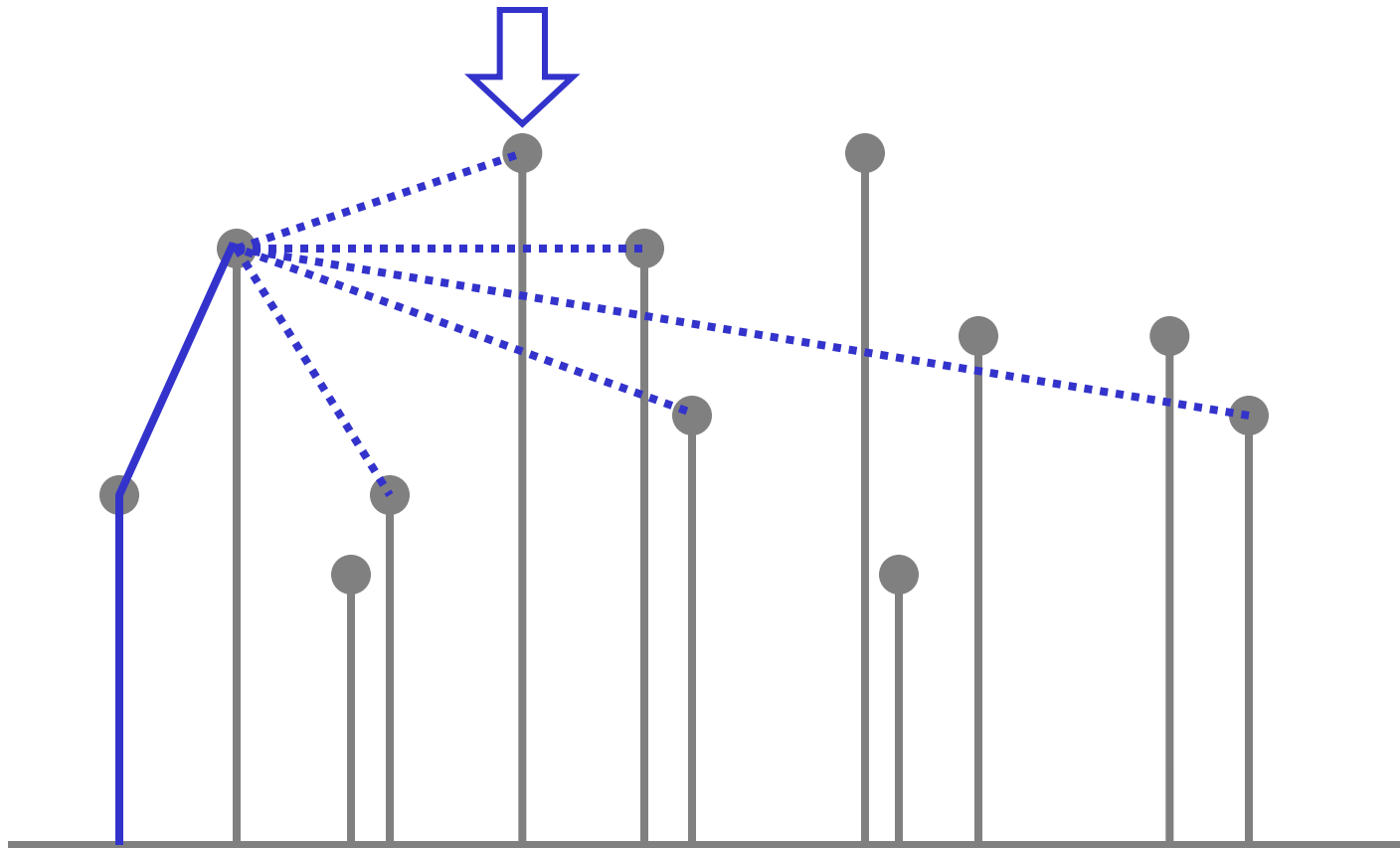
inpakken



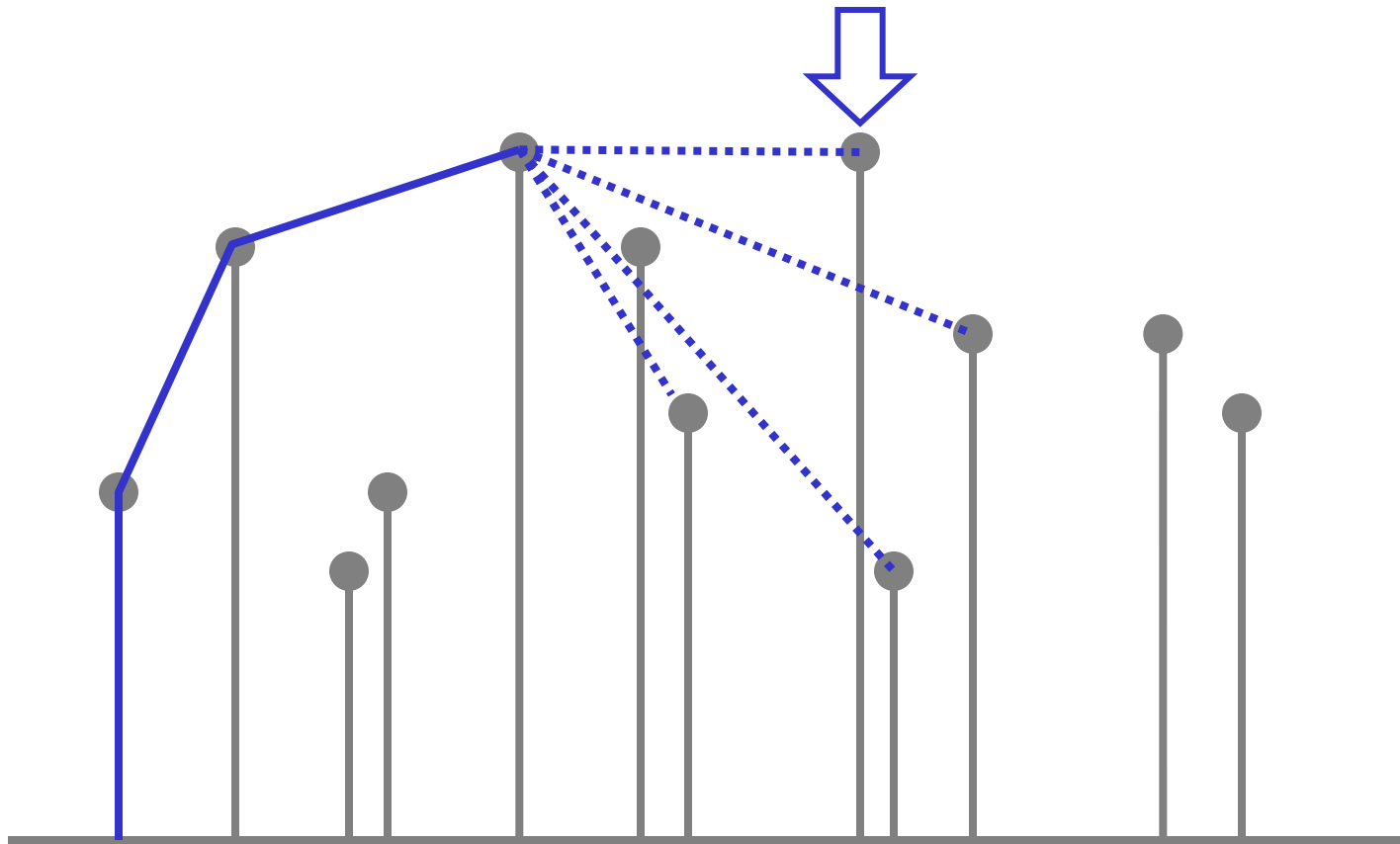
inpakken



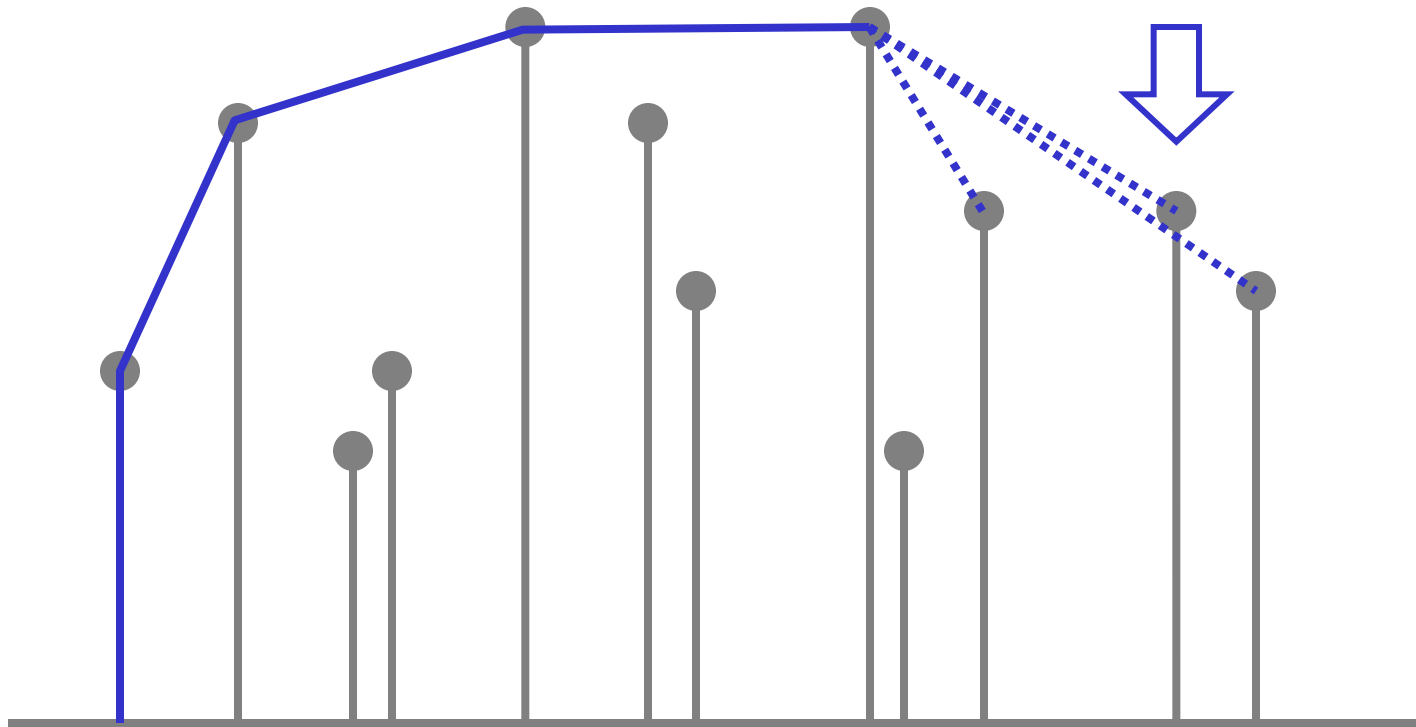
inpakken



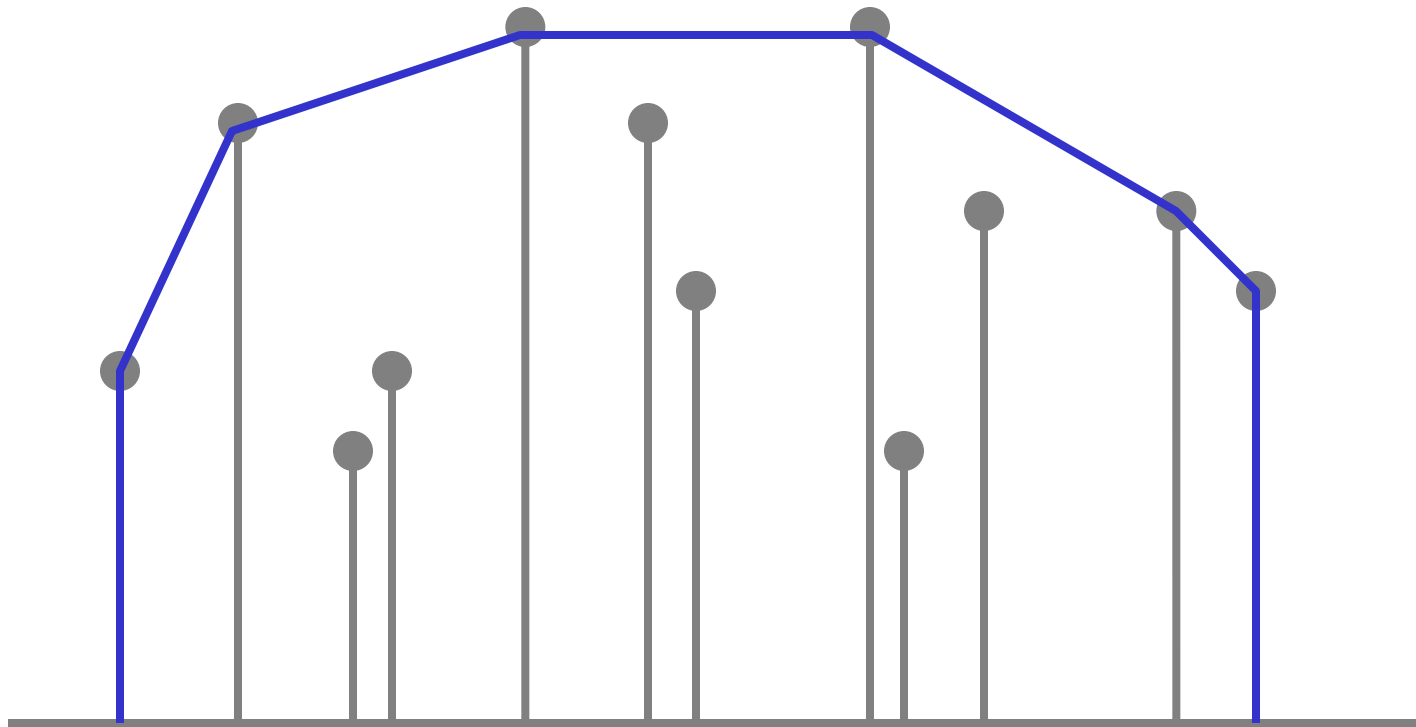
inpakken



inpakken



inpakken

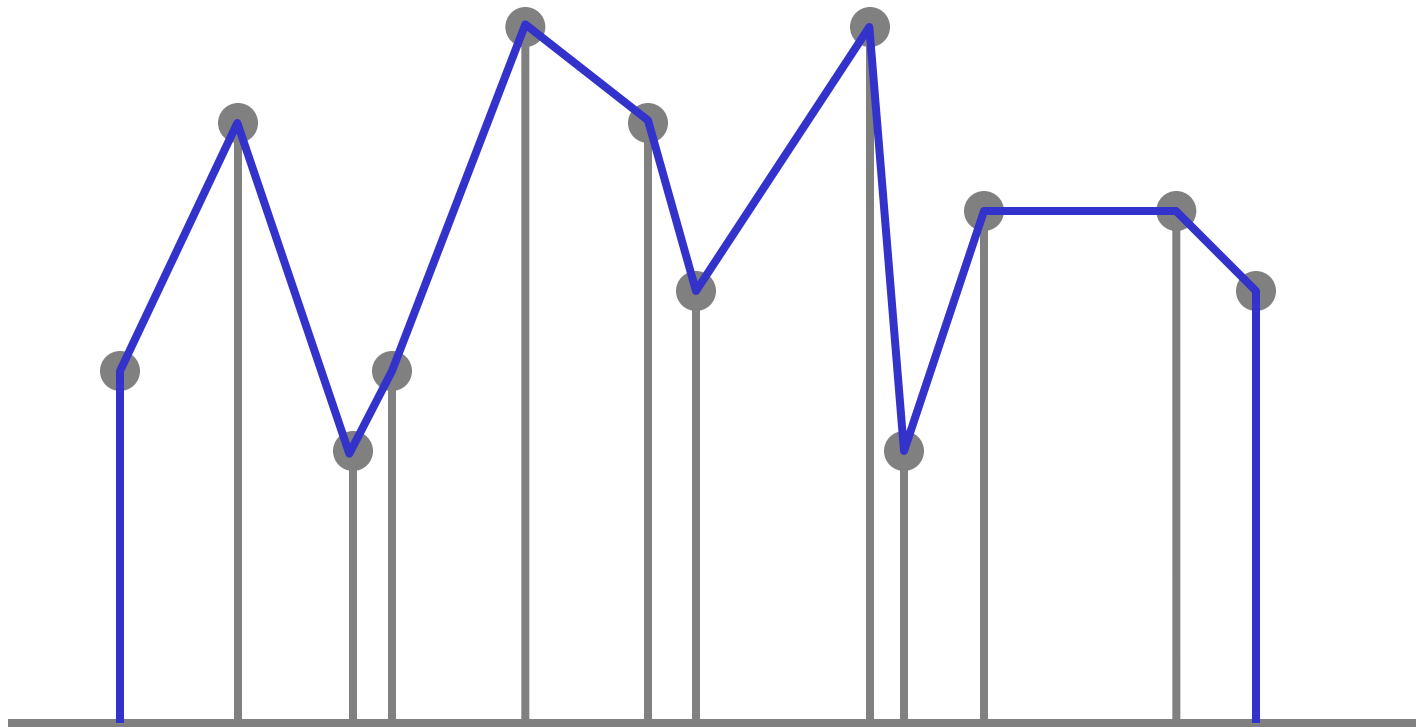


tweede idee

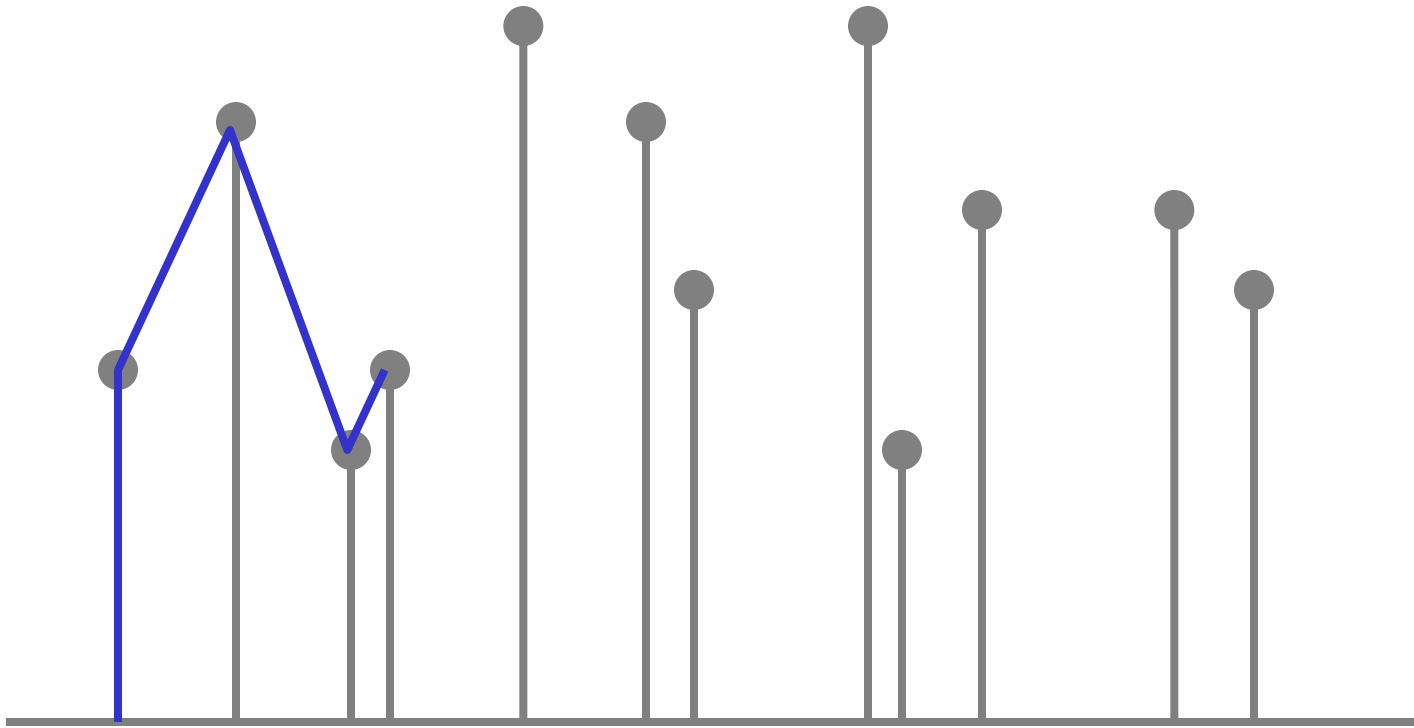




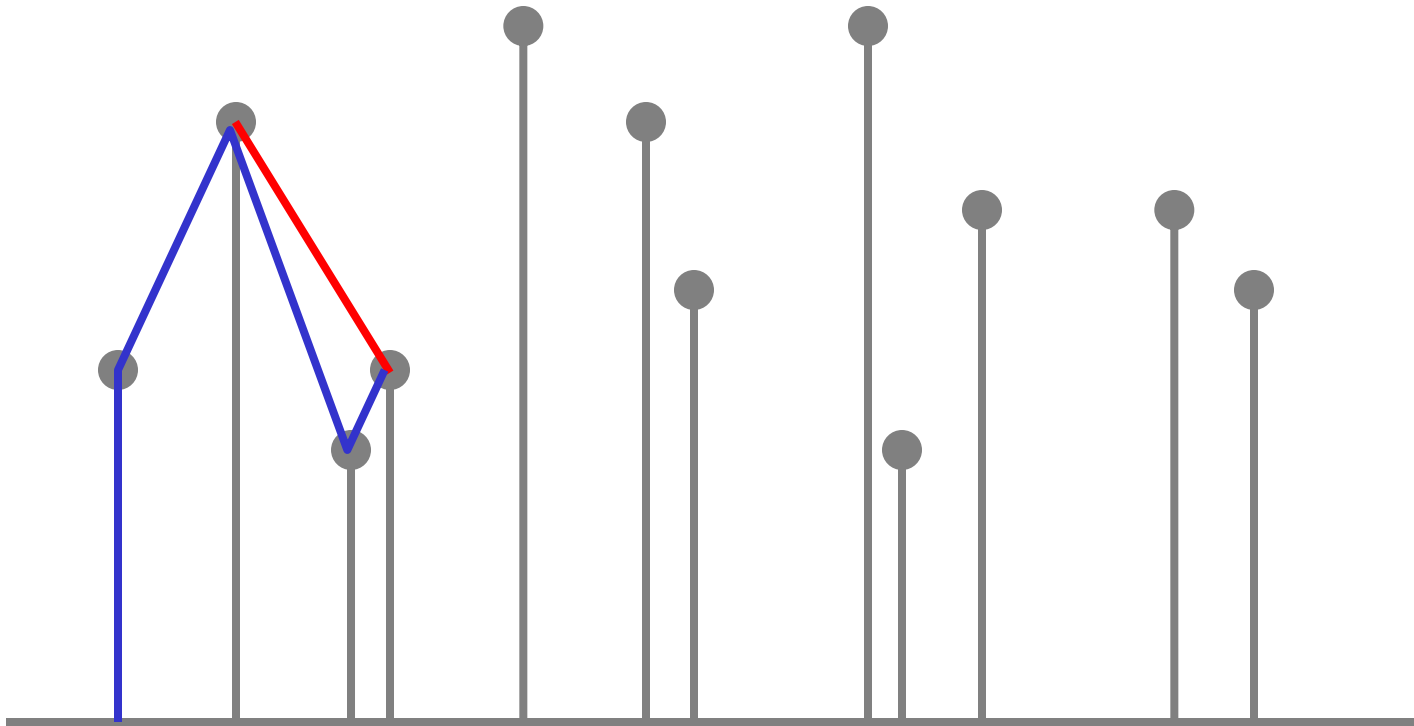
strak trekken



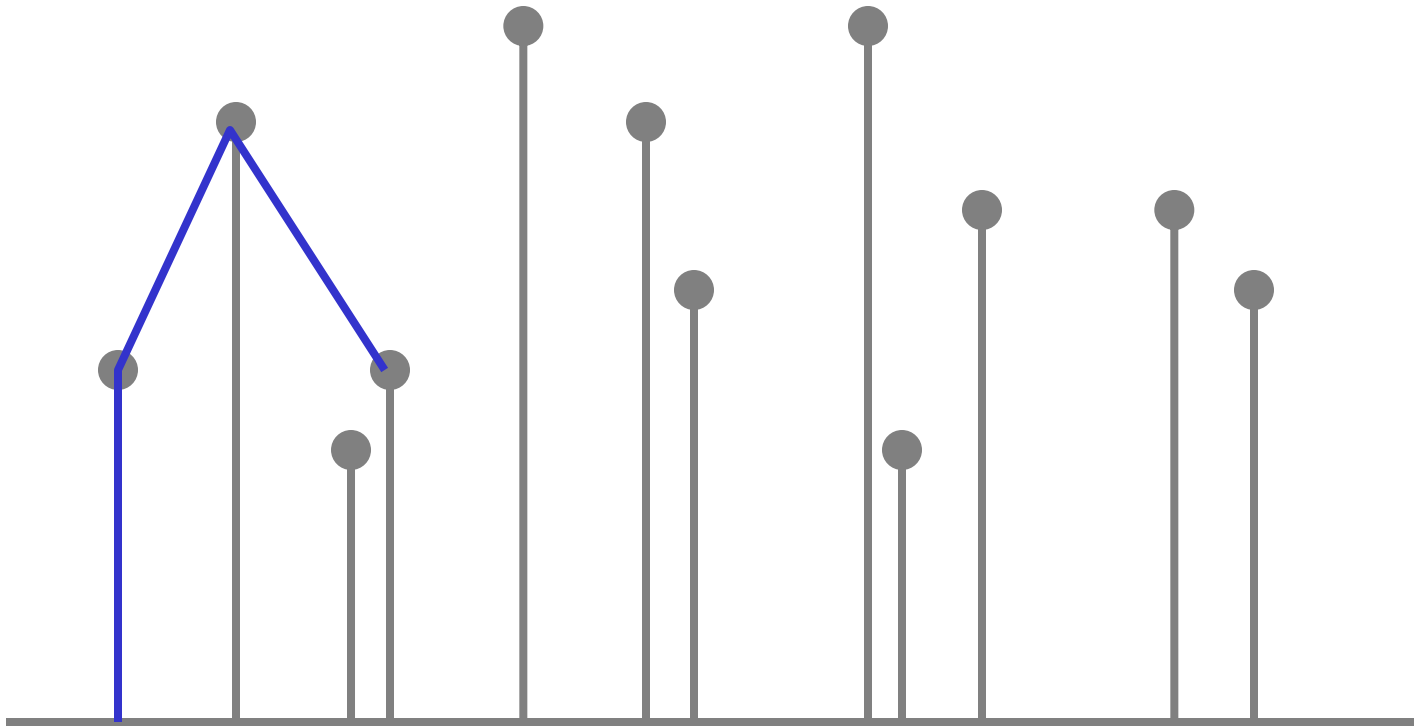
strak trekken



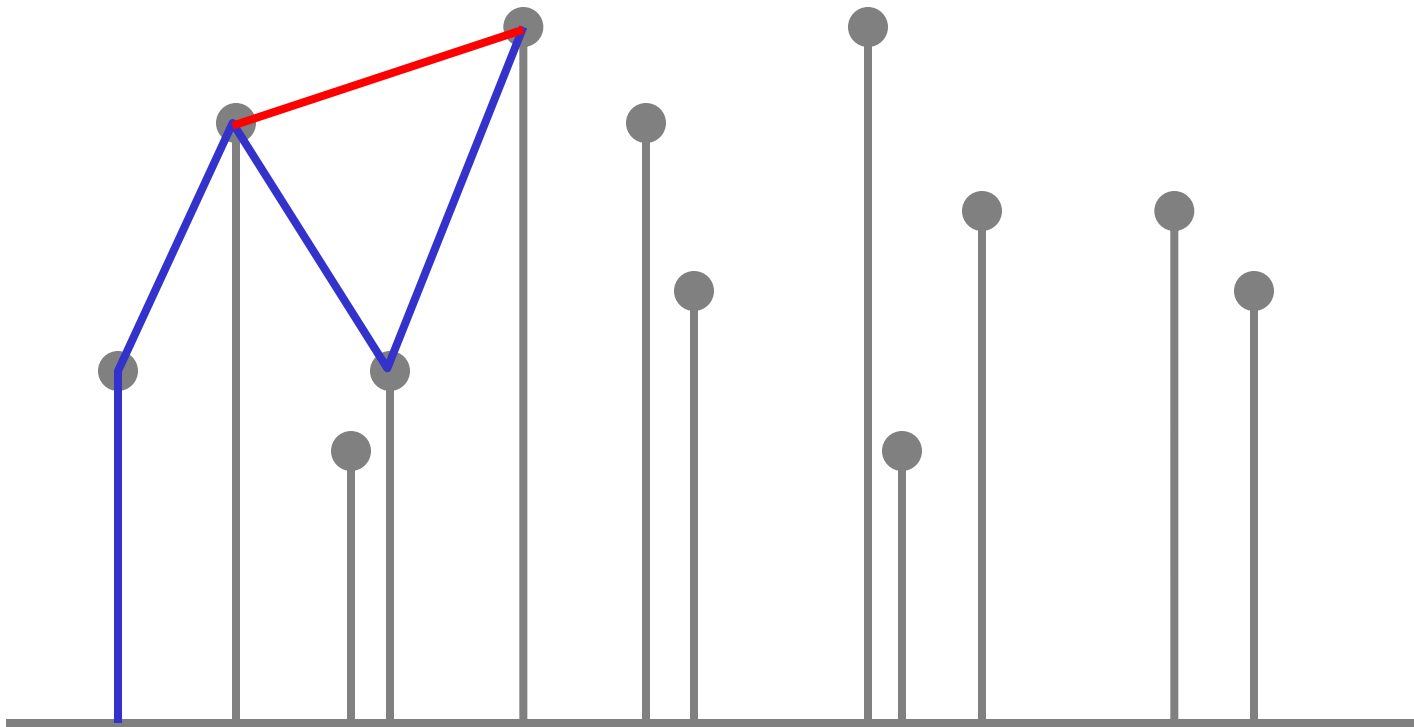
strak trekken



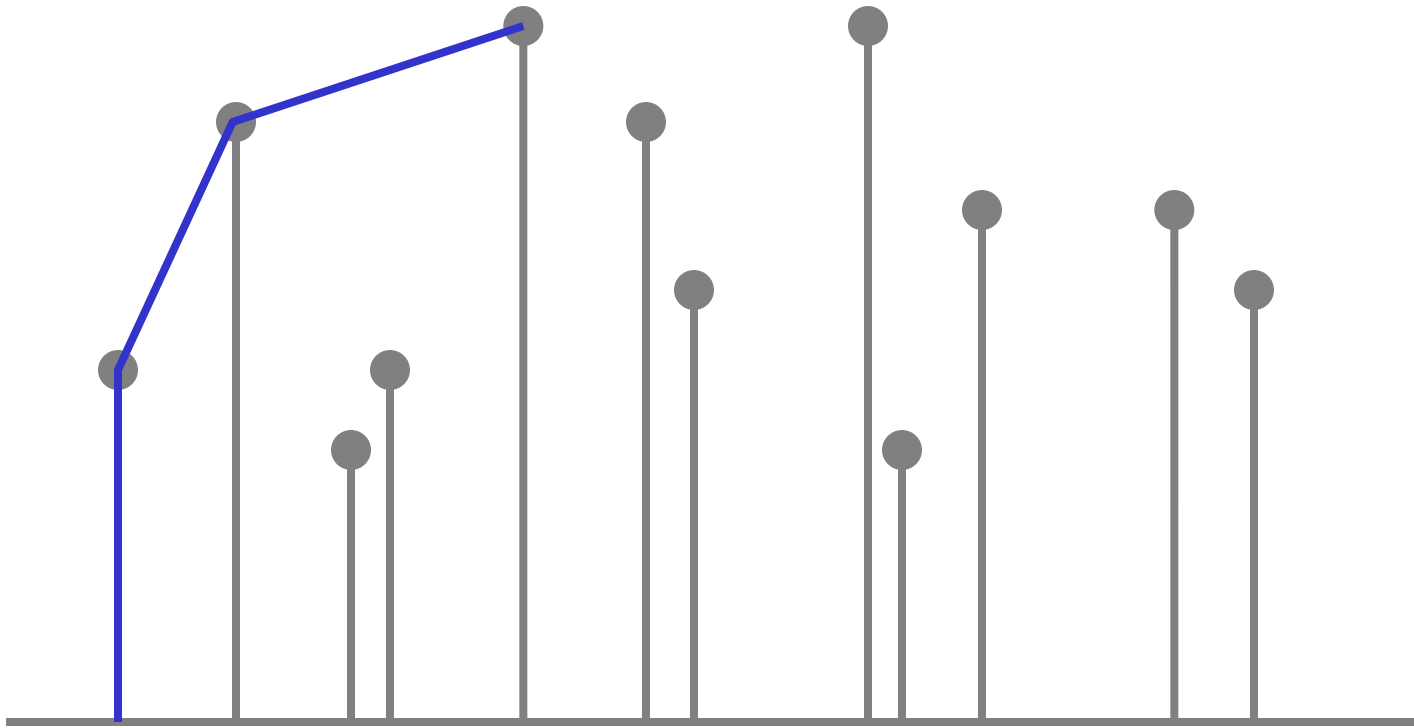
strak trekken



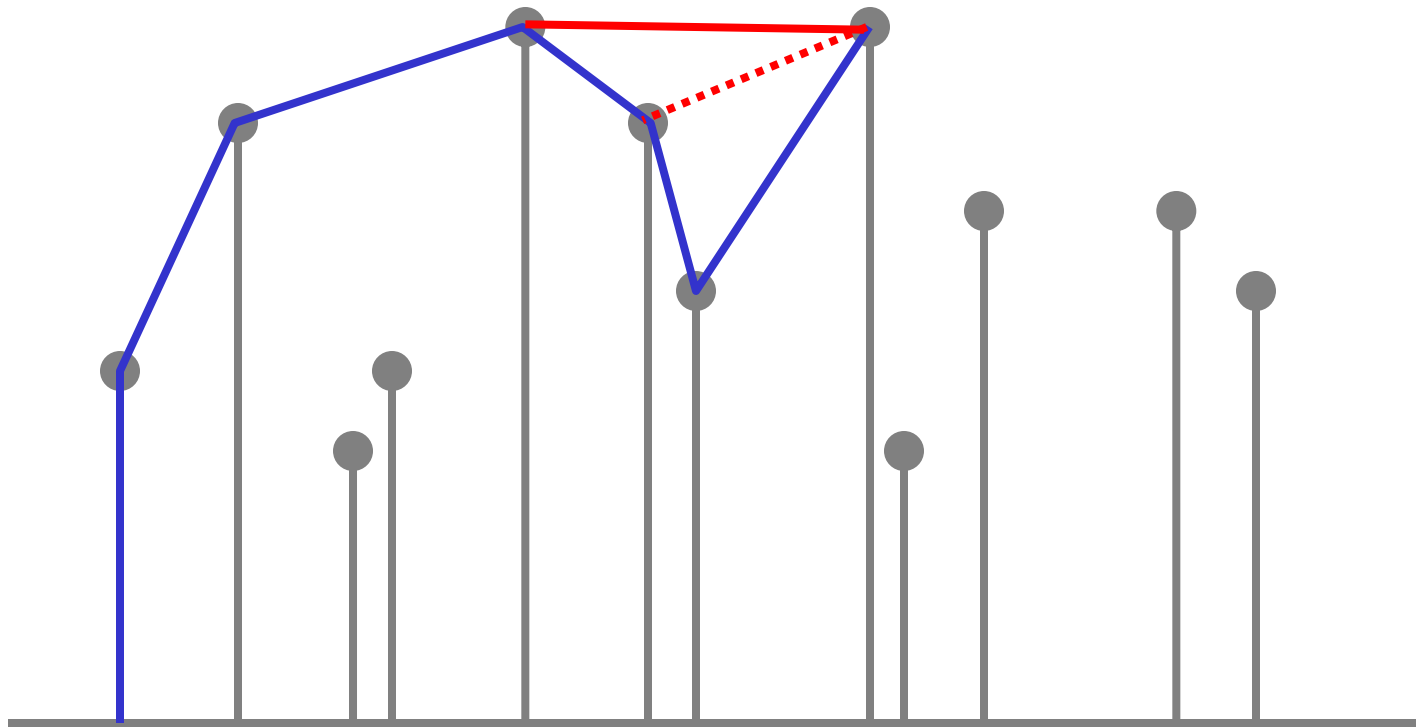
strak trekken



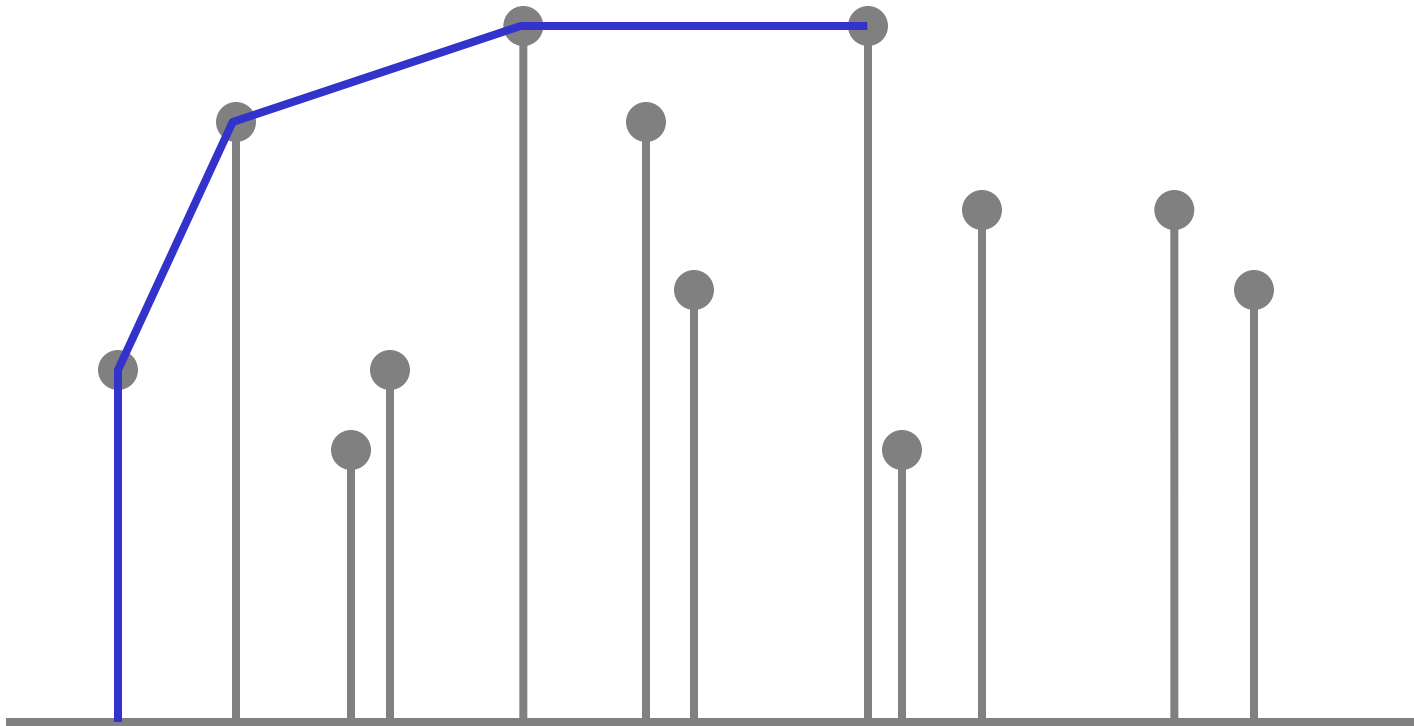
strak trekken



strak trekken

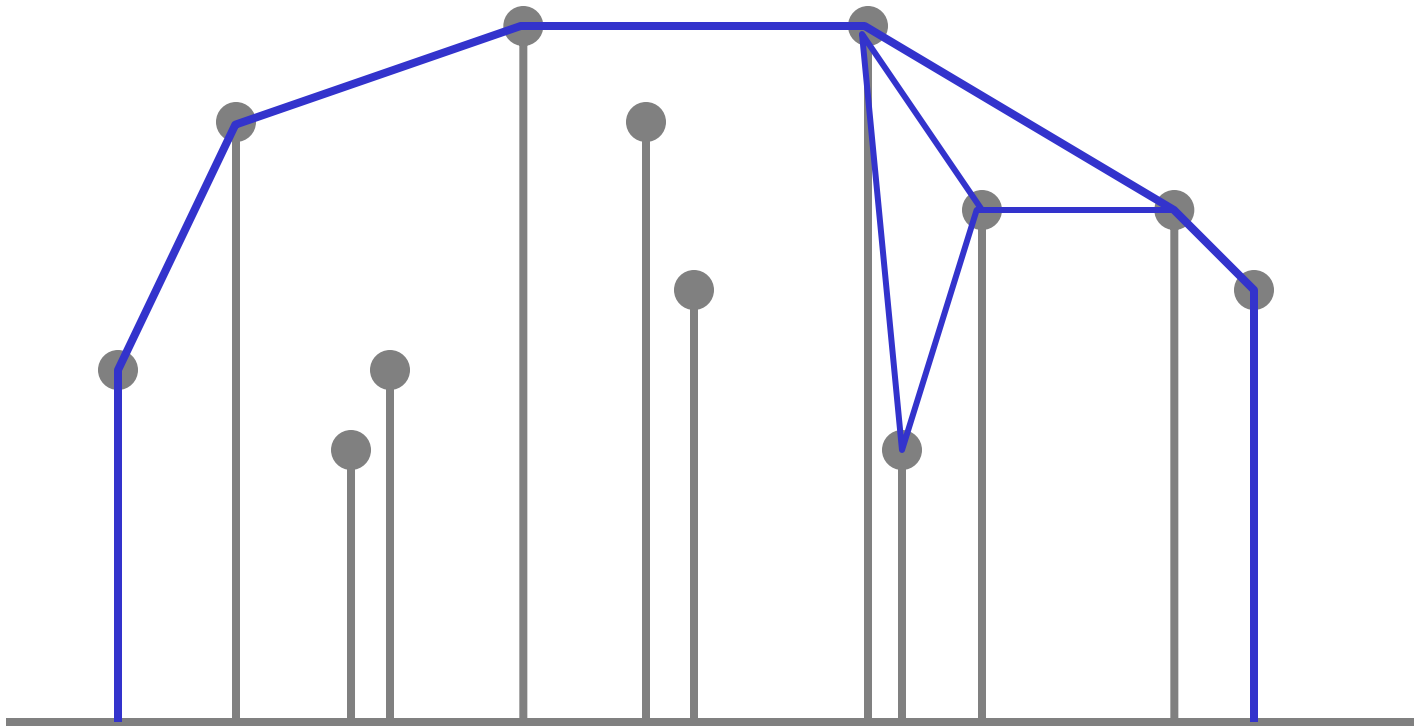


strak trekken

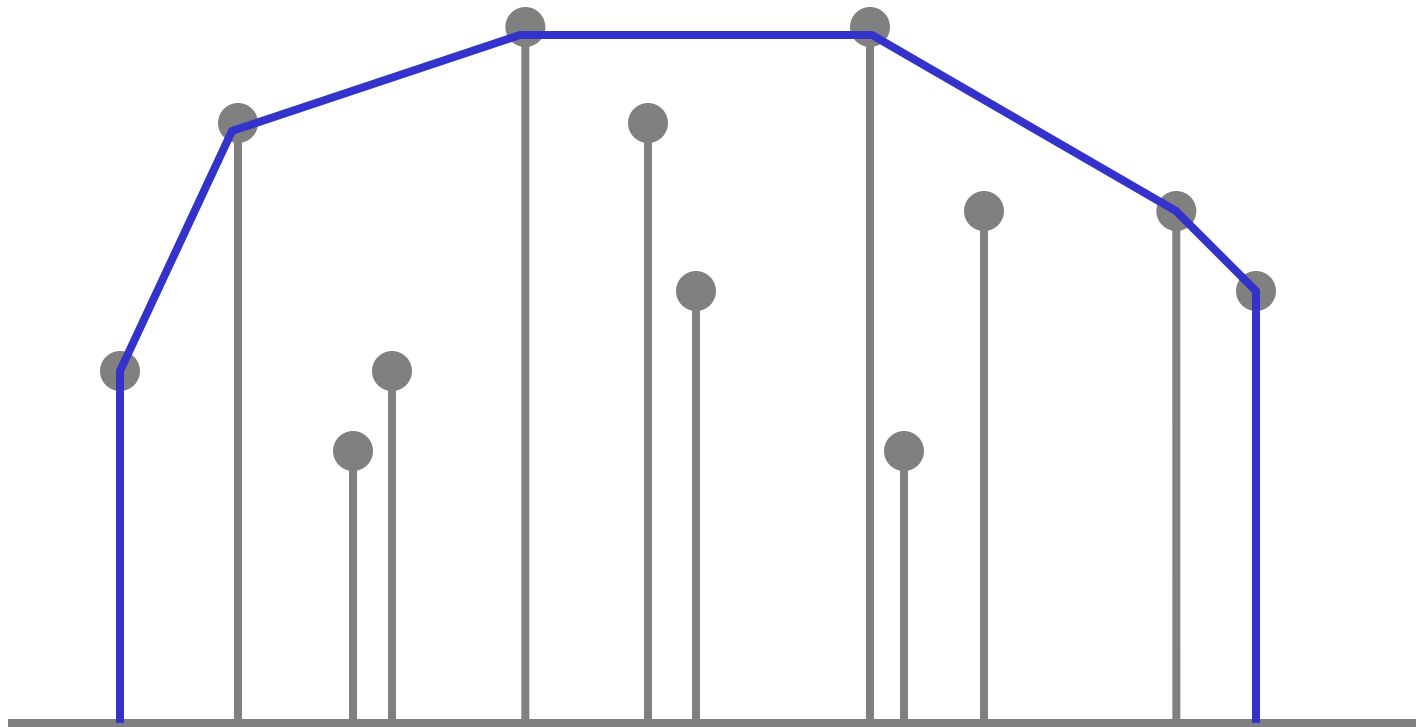




strak trekken



strak trekken



twee  
manieren?

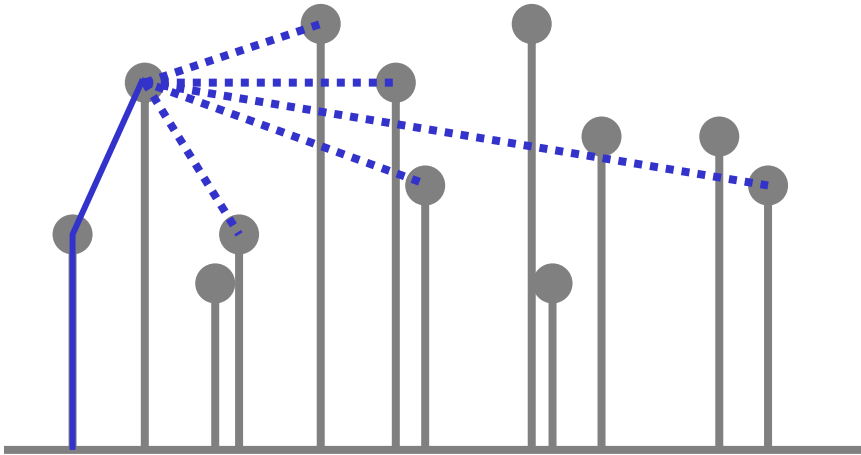


vergelijken

n punten

inpakken

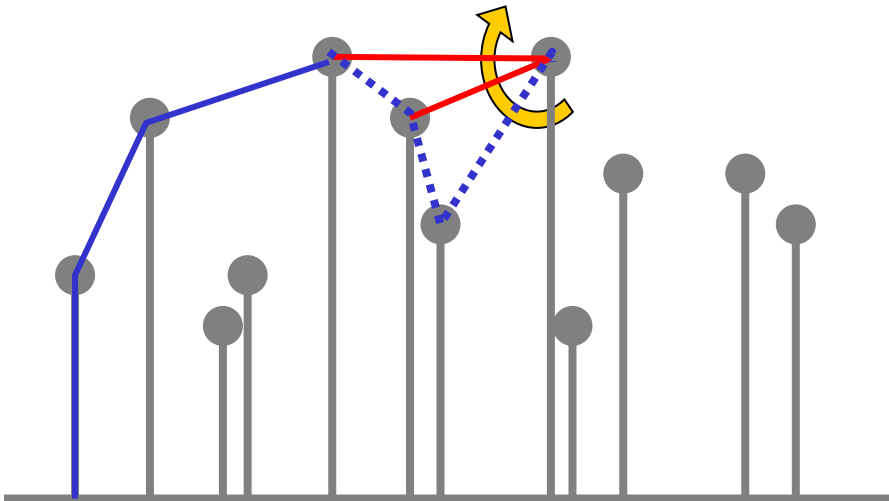
$n+n-1+ \dots + 2+1$   
 $\approx \frac{1}{2}n^2$  stappen



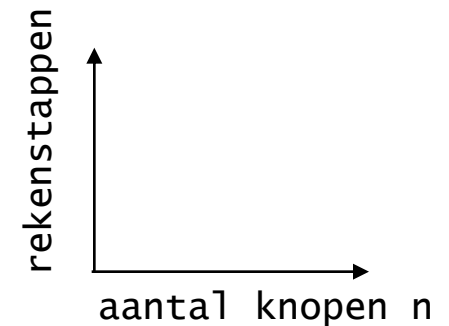
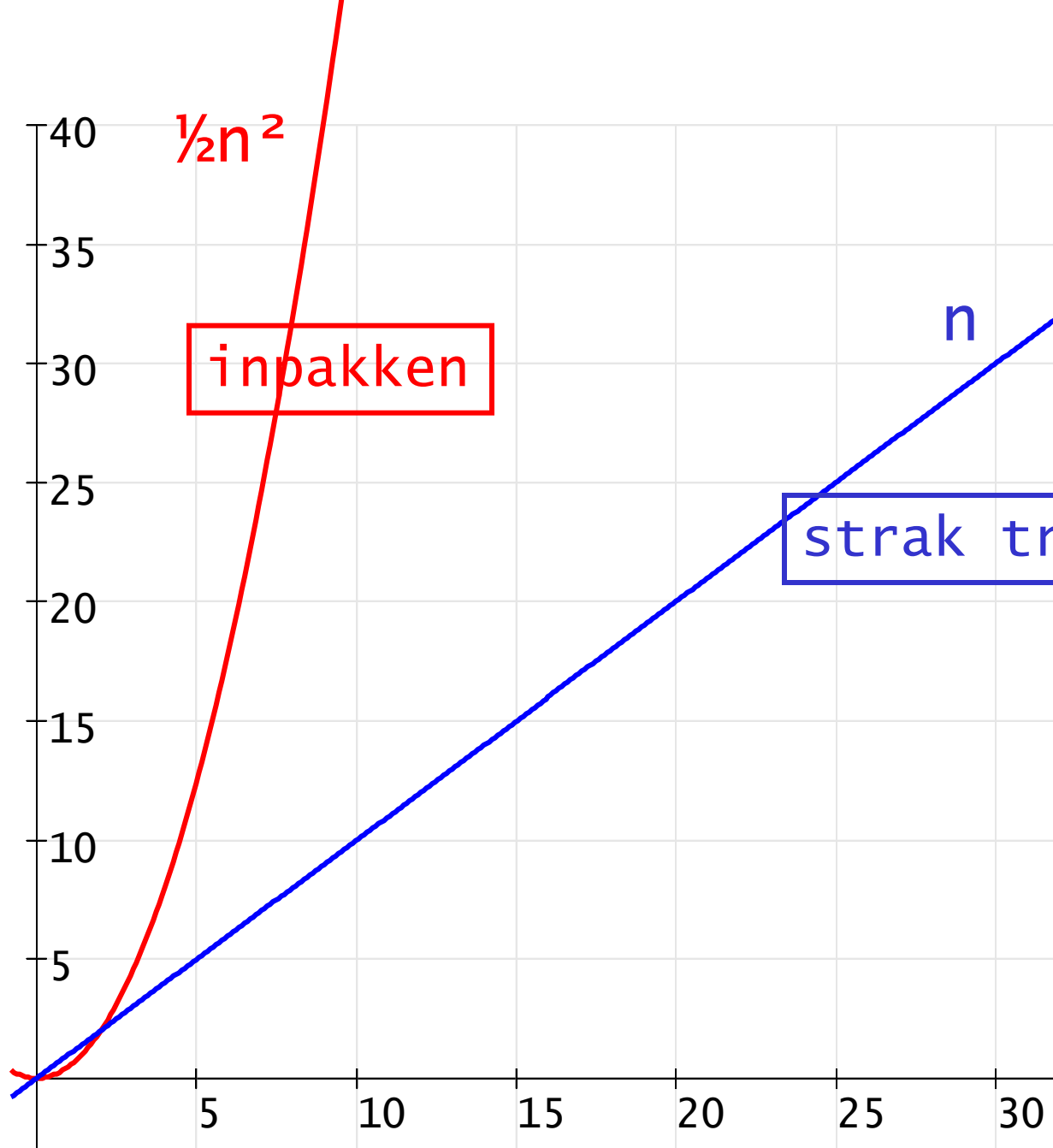
strak trekken

elke knoop max  
één keer

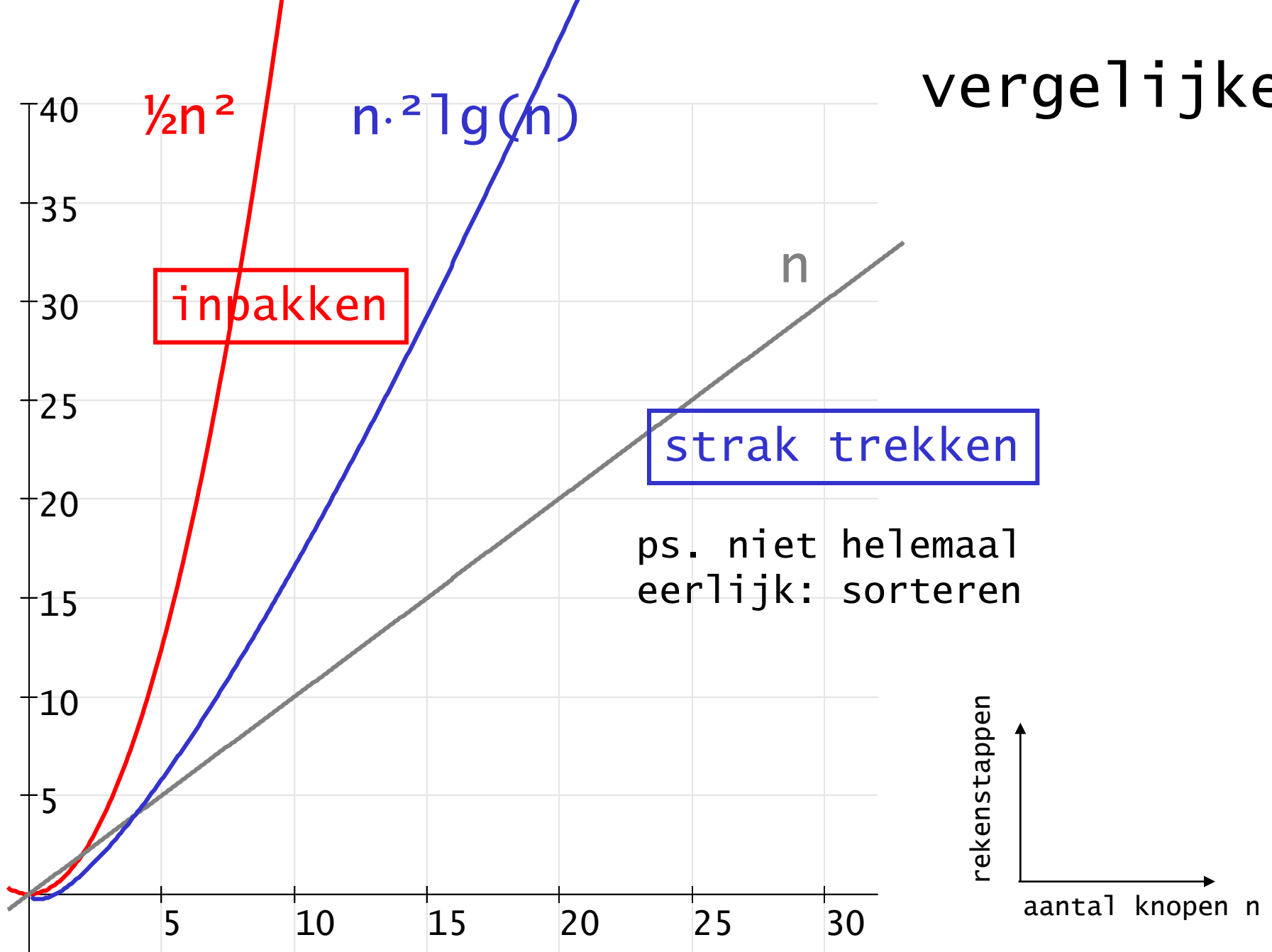
$\approx n$  stappen



# vergelijken

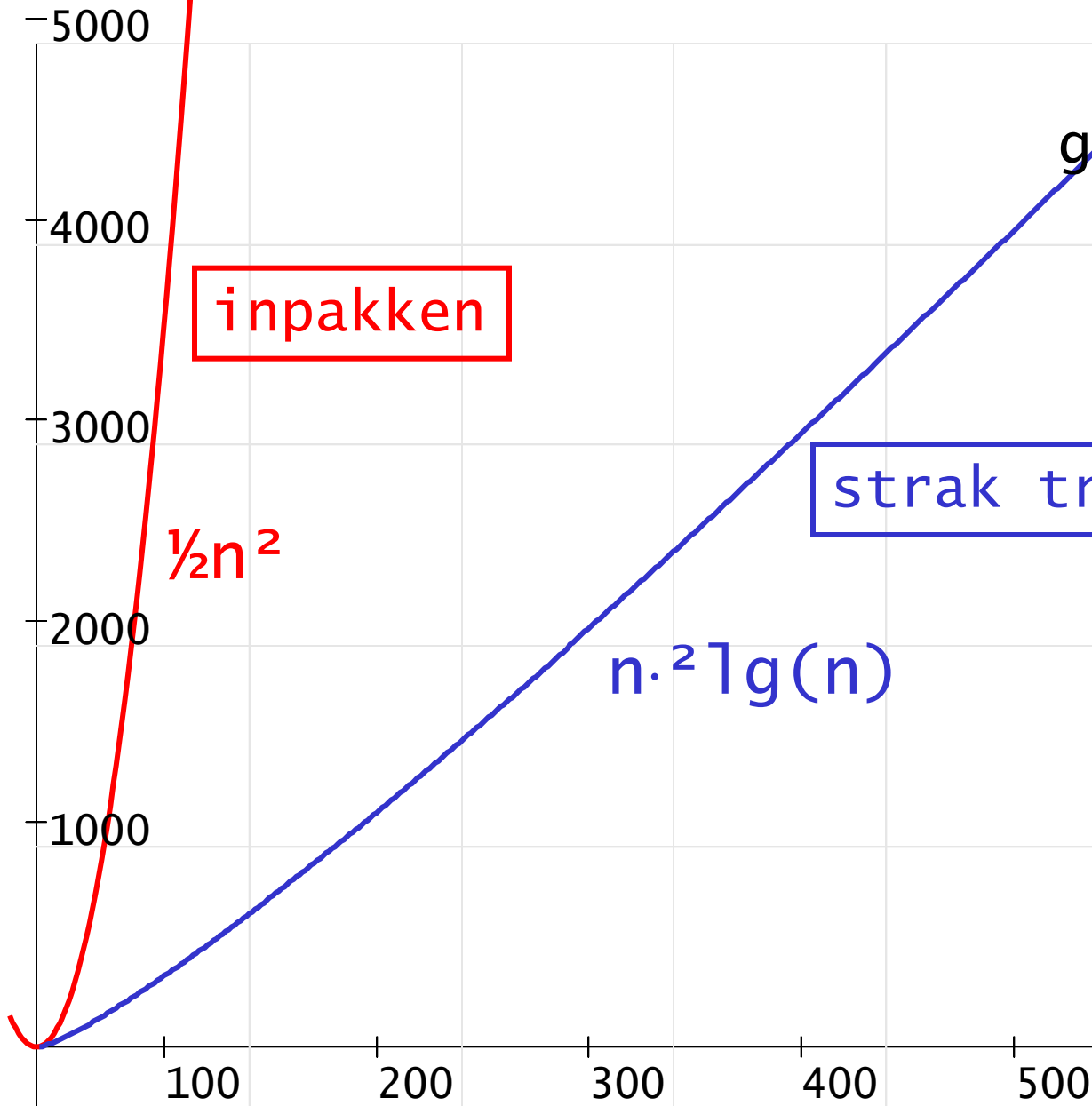


# vergelijken



vergelijken

grotere schaal

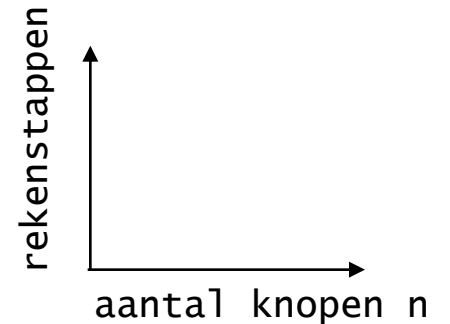


inpakken

$$\frac{1}{2}n^2$$

strak trekken

$$n \cdot 2 \lg(n)$$

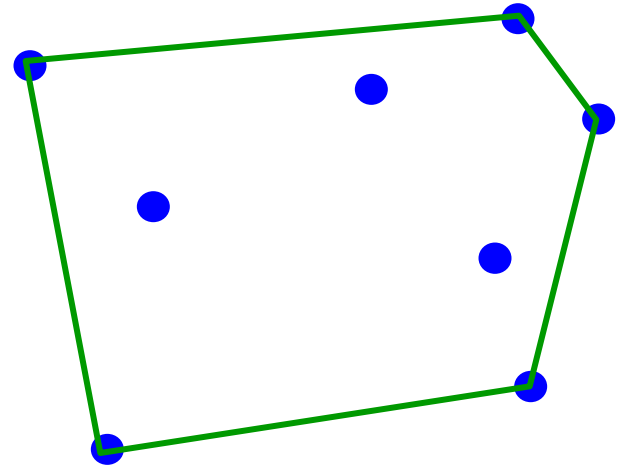
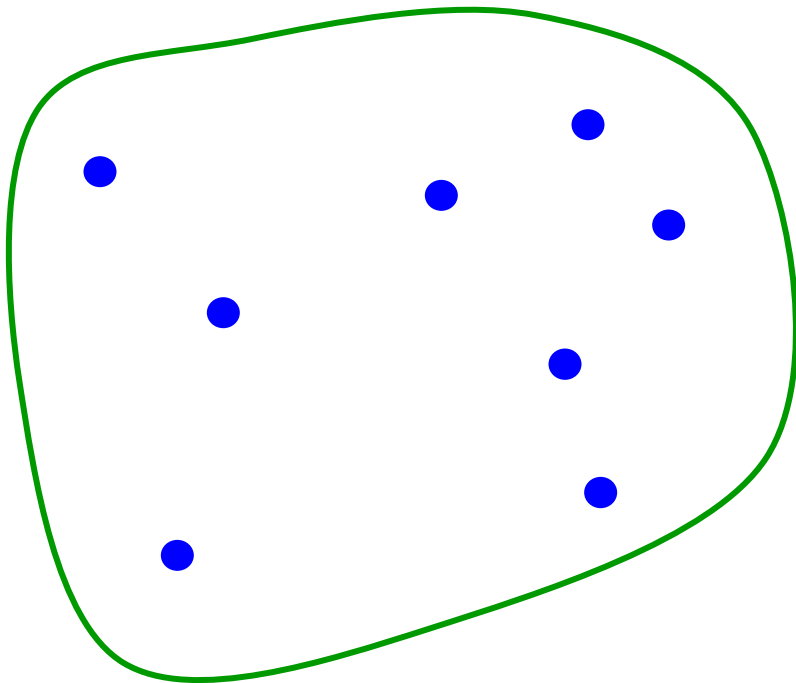




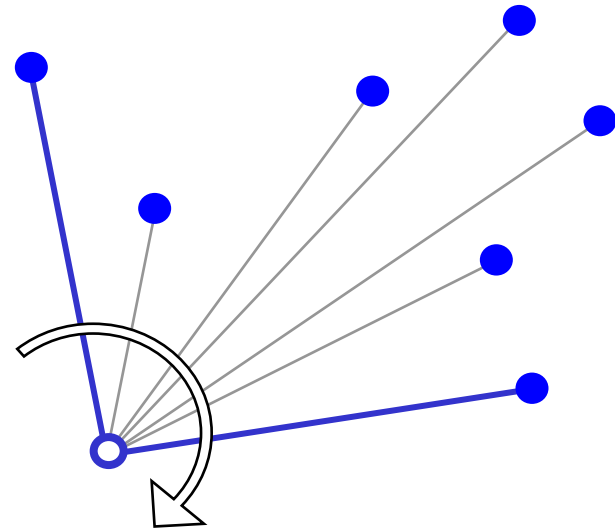
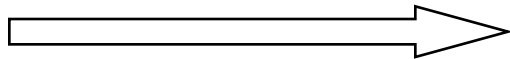
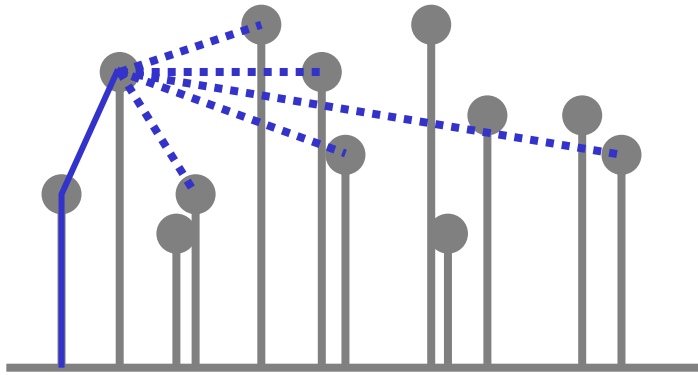
wetenschap?



# convex hull



# ordenen

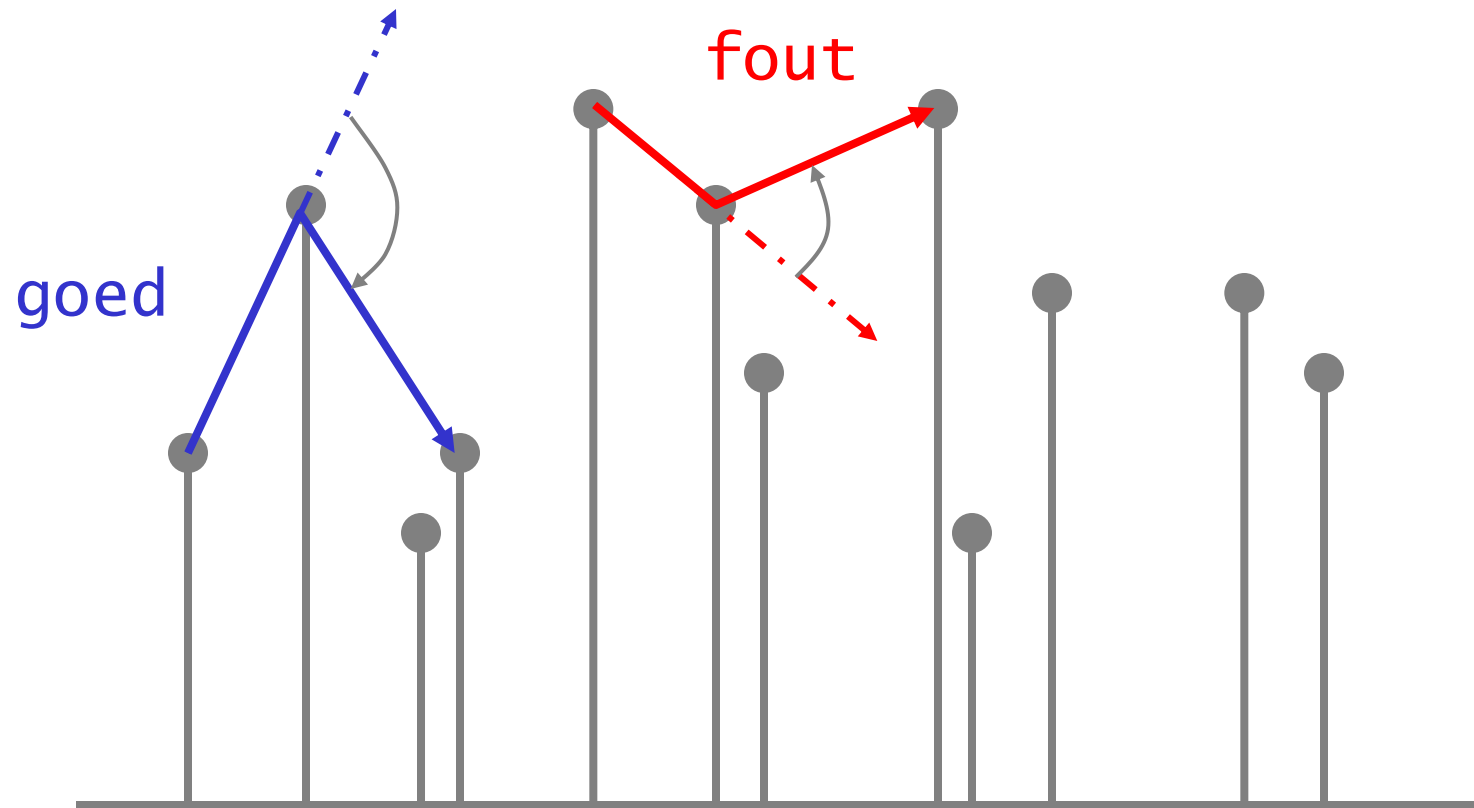


basis voor  
'inpakken' en  
'strak trekken'

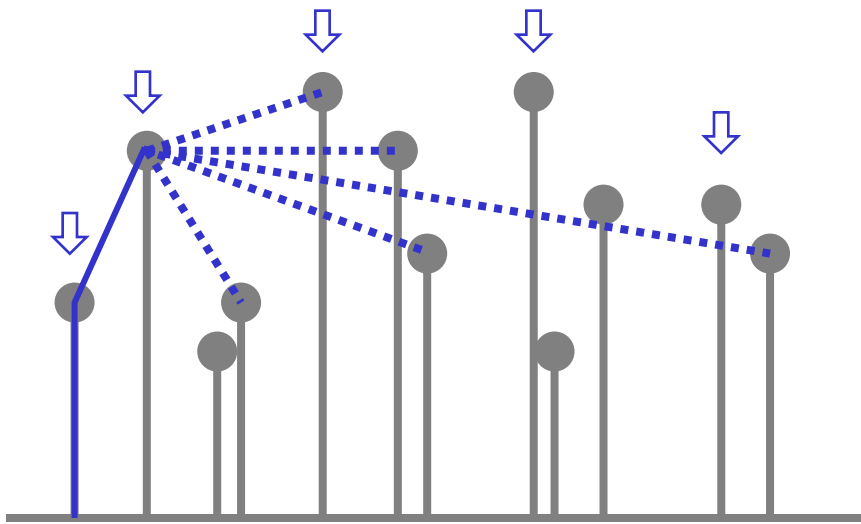
# toepassingen

- collision avoidance in robotics;
- determining whether a point lies amongst a set of other points;
- determining whether two sets of points 'overlap';
- finding the smallest rectangular box that will encompass a set of points; and
- finding a 'rough' description of the shape defined by a series of points.

# principe



# opnieuw: vergelijking



n punten

inpakken

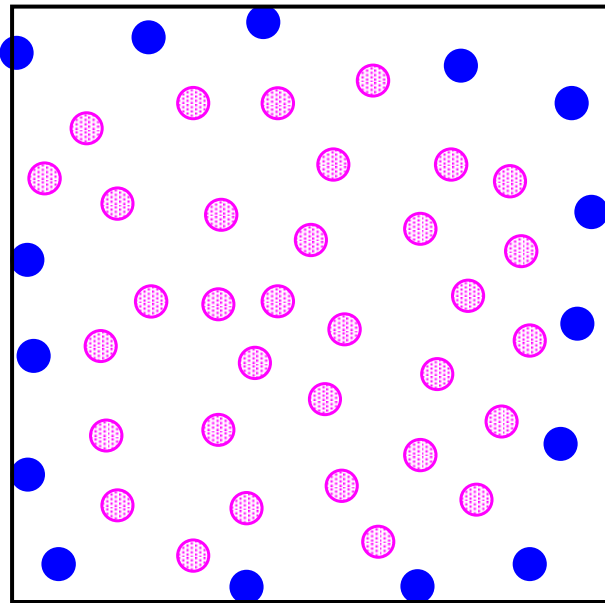
$$n+n-1+ \dots + 2+1 \\ \approx \frac{1}{2}n^2 \text{ stappen}$$

niet elk punt inpakken !?

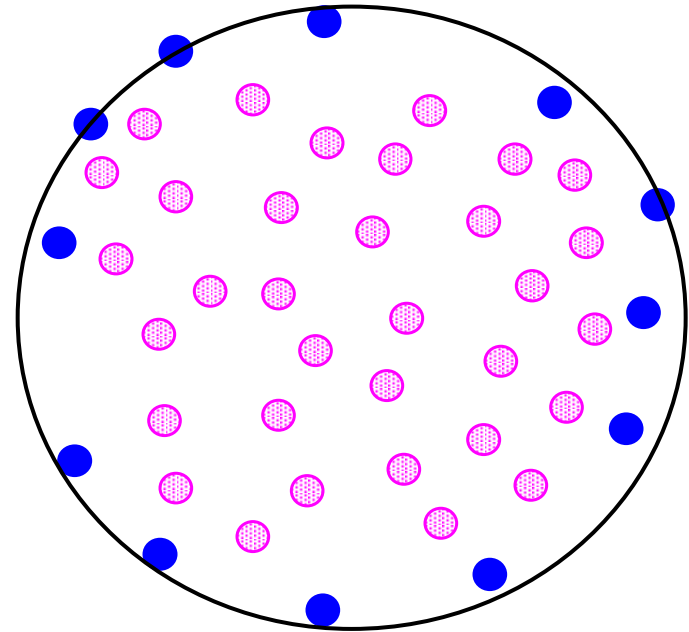
$\approx H \cdot n$  stappen  
met H punten op hull

hoeveel is H ?

$\approx H \cdot n$  stappen



$$H \approx \log n$$



$$H \approx n^{1/3}$$

# vergelijking

50000

40000

30000

20000

10000

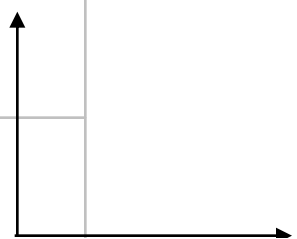
$$n^{4/3}$$

cirkel

vierkant

$$n \cdot \lg(n)$$

rekenstappen



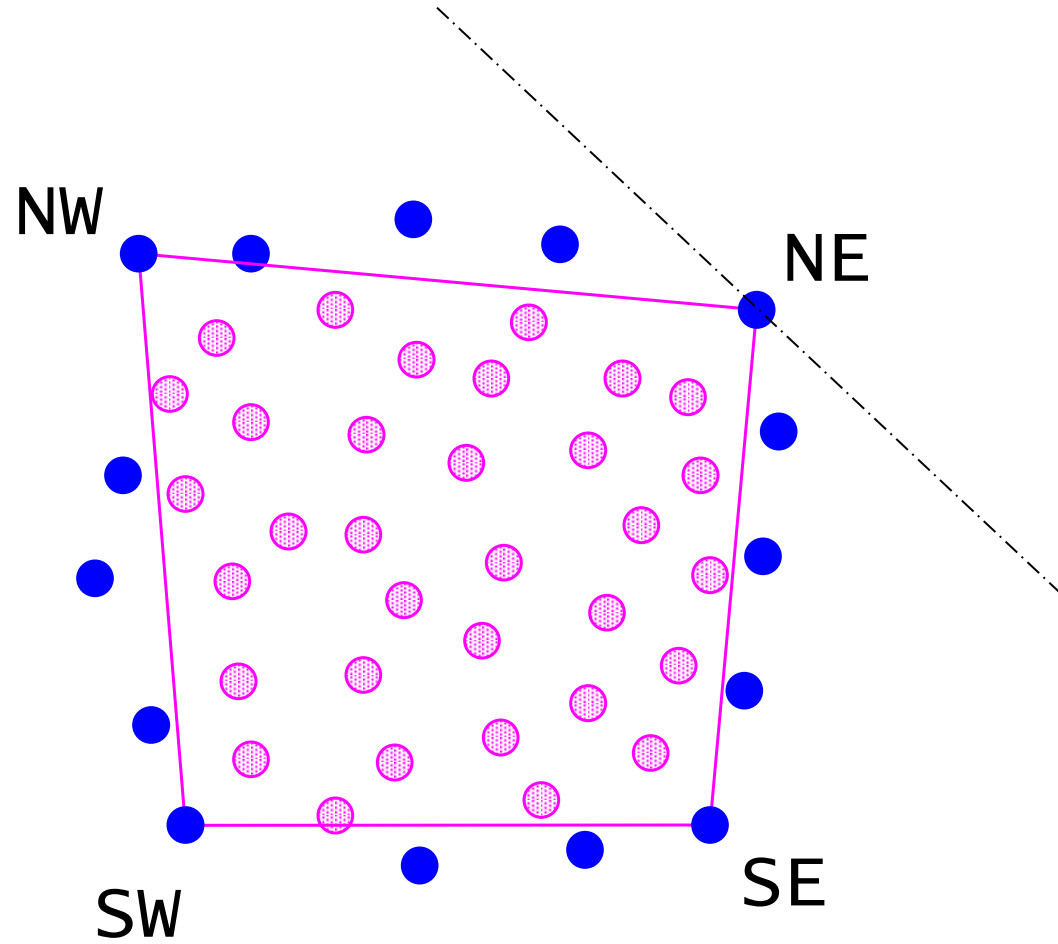
aantal knopen n

algoritmen:  
meer varianten

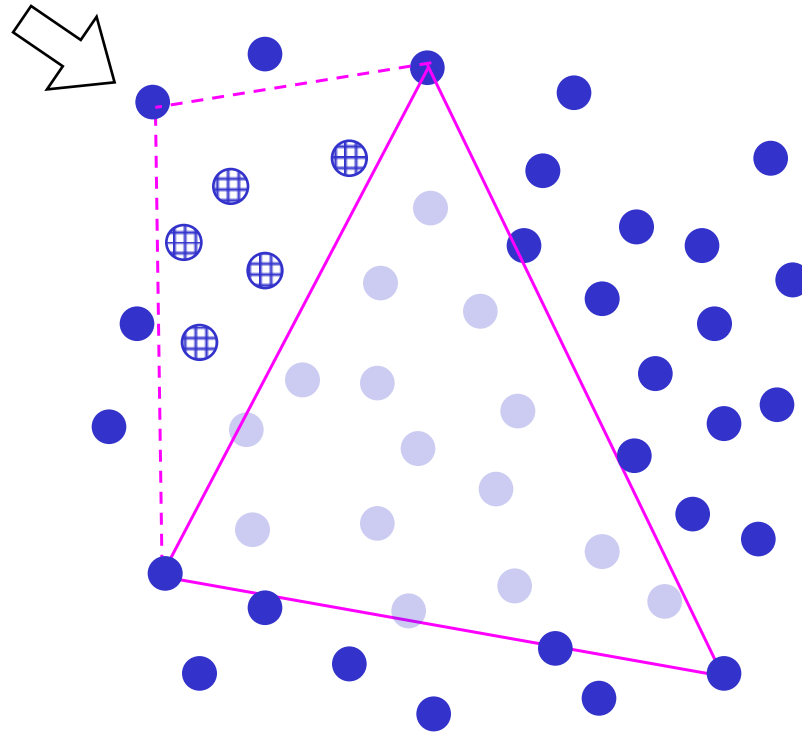




reductie



# quick hull



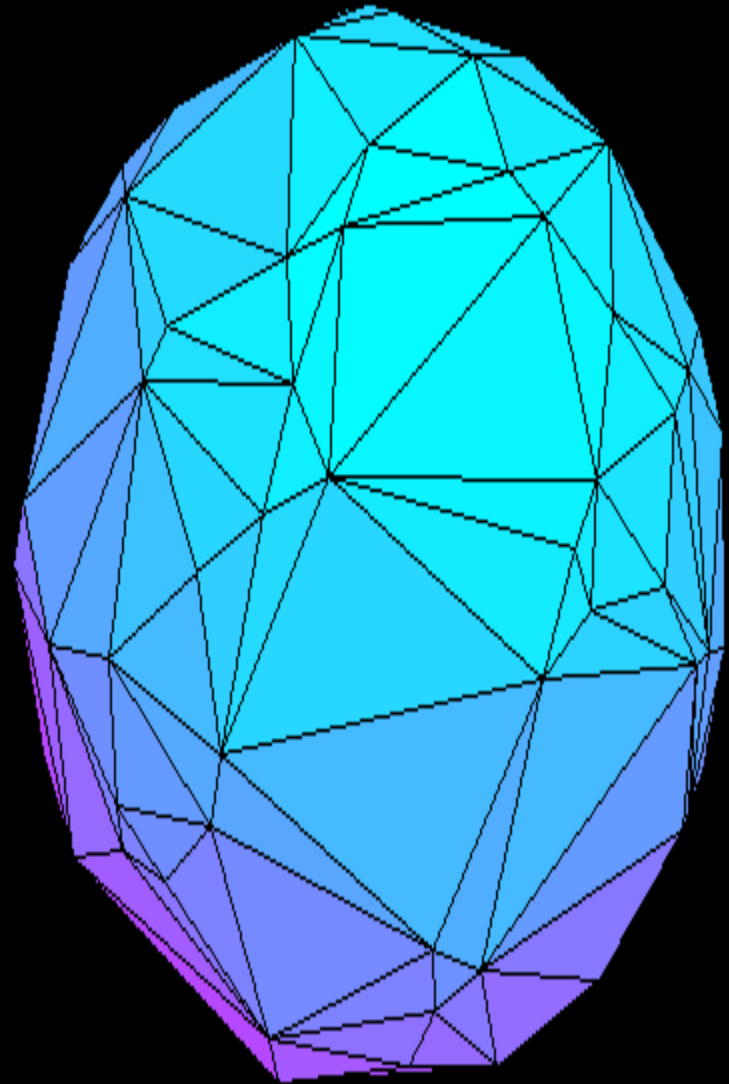
niet gegarandeerd snel

## convex hull algoritmes

- *Jarvis' march*  
‘inpakken’
- *Graham's scan*  
‘strak trekken’
- *Quick hull*  
*heuristiek*

## toepassingen

- robotics
- beeldinformatica



# Bedankt, en tot ziens

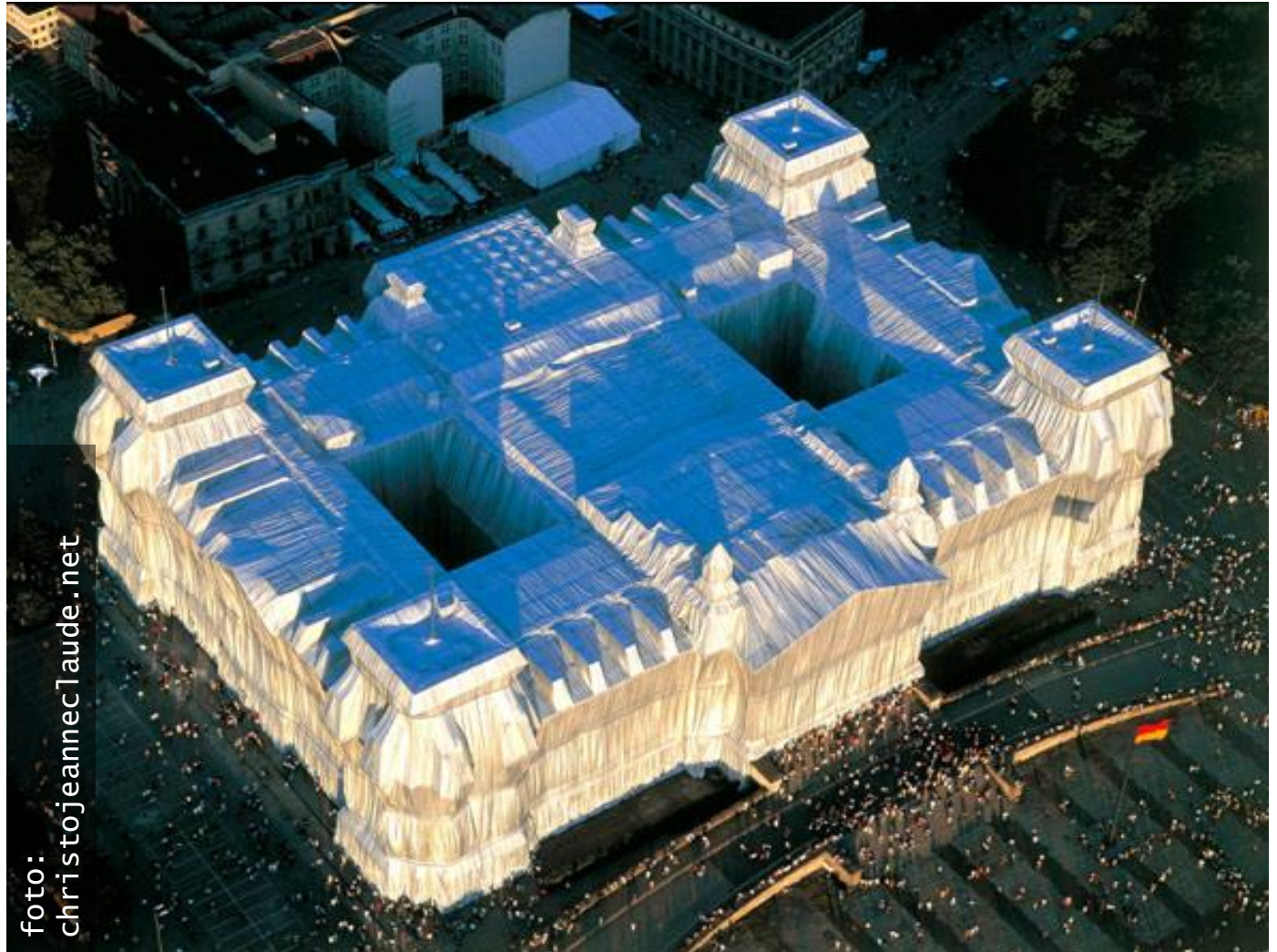


foto:  
christojeanneclaude.net