

WELKOM

3i workshop

Grondslagen I

Rien den Besten
Adriaan Gijssen

PROGRAMMA

- Voorstellen en introductie
- Standaardalgoritmen langslopen
- Correctheid en efficiëntie
- Brute force en complexe problemen
- Pauze
- Recursie
- Implementatie

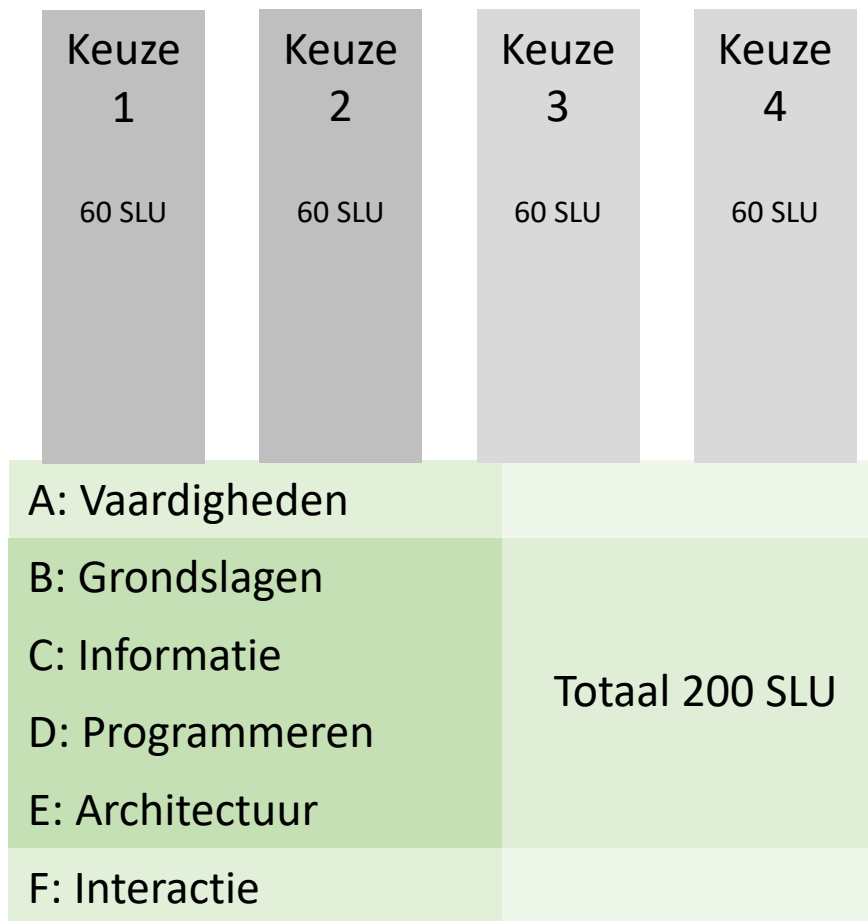
WIE ZIJN WIJ



VOORSTELLEN

- Welke school, welke klassen?
- Wat weet je van het nieuwe examenprogramma?
- Al eerder een evenement bijgewoond?
- Verwachtingen voor deze middag

EXAMENPROGRAMMA



VOLG ALLE ONTWIKKELINGEN 3iblog.nl



UITPROBEREN

MEEDENKEN

SUGGESTIES EN VRAGEN

CONTACT

HET NIEUWE INFORMATICA IN DE BOVENBOUW

Lesmateriaal en nieuws



INFO VOOR DOCENTEN



INFO VOOR SCHOOLLEIDERS

EVENEMENTEN

10/10: Grondslagen I

12/12: Grondslagen II

POPULAIRE BLOGS



Workshops
Grondslagen

TIJDPAD

- Complete kernprogramma beschikbaar september 2018
- Start ontwikkeling projecten / keuzethema's september 2018
- Cursussen maart 2018 – juli 2019

HOE IS ONS MATERIAAL GEMAAKT?

- In nauw overleg met SLO
- Redacteuren, grafisch ontwerper en programmeur
- Grote groep 3i referenten (11 personen)
 - Docenten
 - Vakdidactici
 - Vakinhoudelijke specialisten (TU informatica)
- Praktijktest door 3i Early Adopters (17 personen)

DIDACTISCHE OPZET

- Steeds afwisseling **korte tekst – vragen en opdrachten**
- Geen losse onderwerpen, meer **gestapelde uitleg**
- **Filmpjes, animaties en interactieve elementen**
- Alles **online**

ONZE ERVARING

- Een uitdagend en boeiend examenprogramma
- Leerlingen krijgen een stevige basis
- Leerlingen worden uitgedaagd

FEEDBACK VANUIT ADOPTERS

- Domein D is erg beperkt in examenprogramma
- Grote projecten ontbreken nog
- Feedback op vragen voor leerlingen in de online omgeving

Inhoudelijk

- Standaardalgoritmen langslopen
 - Soorten algoritmen
 - Correctheid en efficiëntie
- Brute force en complexe problemen

Binary Search

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Speel opnieuw

Raad het getal

Pogingen: 0

Binary Search

- Wat is efficiëntie?
 - *Tijd (of stappen?)*
 - *Ruimte (bandbreedte?)*
- Efficiëntie bepalen
 - Bestcasescenario (het beste)
 - Worstcasescenario (het slechtste)
 - Averagecasescenario (het gemiddelde)

Binary Search

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

In 1 stap

Speel opnieuw

Raad het getal

Pogingen: 0

Binary Search

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

In 2 stappen

Speel opnieuw

Raad het getal

Pogingen: 0

Binary Search

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

In 3 stappen

Speel opnieuw

Raad het getal

Pogingen: 0

Binary Search

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

In 4 stappen

Speel opnieuw

Raad het getal

Pogingen: 0

Binary Search

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

In 7 stappen

Worstcasescenario's

Speel opnieuw

Raad het getal

Pogingen: 0

Vragen

- Binary search toepassen
- Ik heb een letter in gedachte. (keuze uit A t/m Z). Jij mag raden door steeds een letter te vragen. Ik zal antwoorden met “eerder in het alfabet” of “later in het alfabet” of “die is het”.
 - Hoeveel keer moet je raden in het bestcasescenario?
 - Hoeveel keer moet je raden in het worstcasescenario?

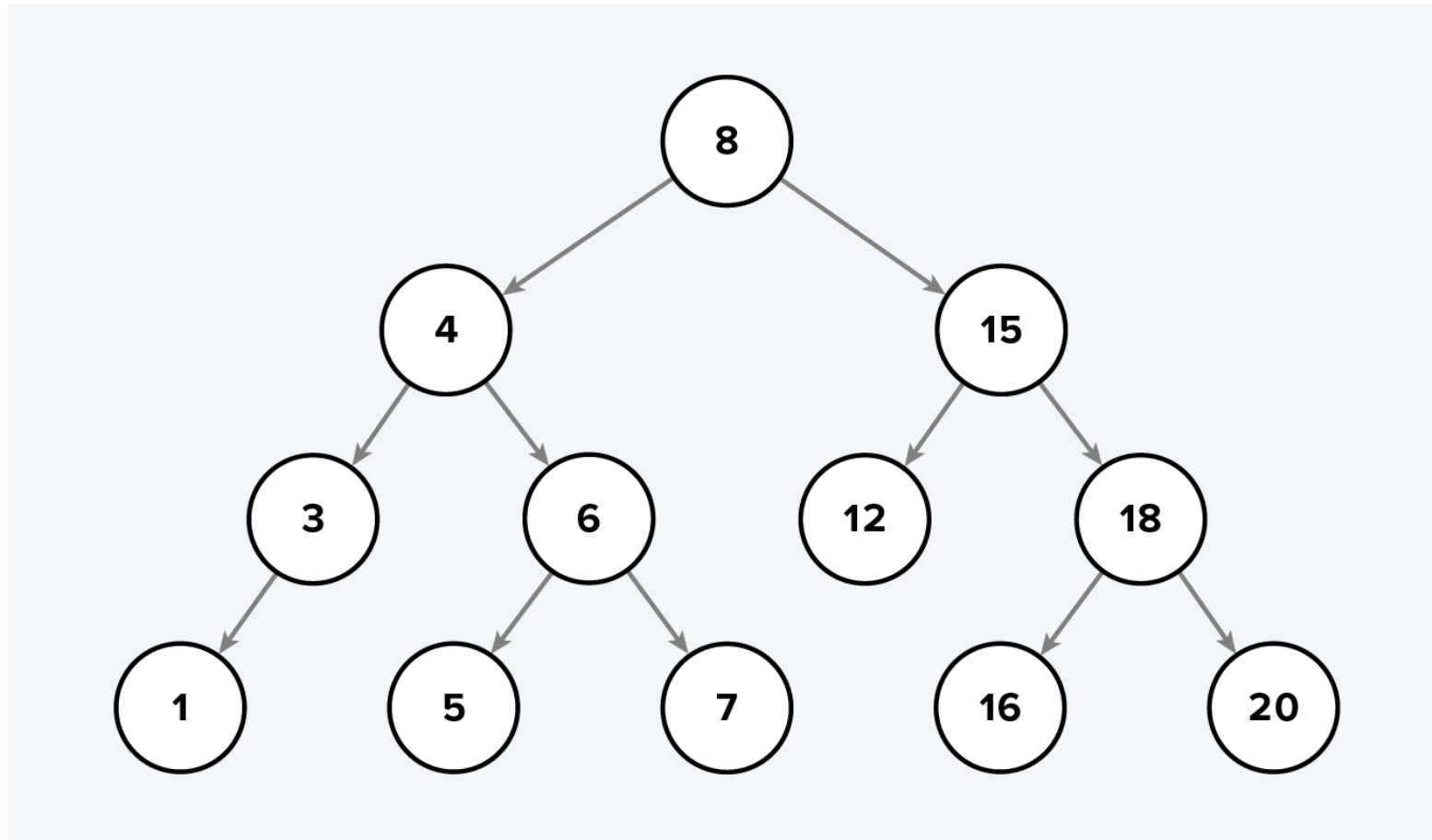
Vragen

- Binary search toepassen
- Ik neem een getal in gedachte. Het getal is minstens 1 (ondergrens). Jij mag gaan raden. Ik mag enkel zeggen dat het getal lager is, of dat het hoger is, of dat het goed is.
Geef de bovengrens waarbij jij nog zeker in maximaal 10 keer mijn getal raadt.

Orde van efficiëntie

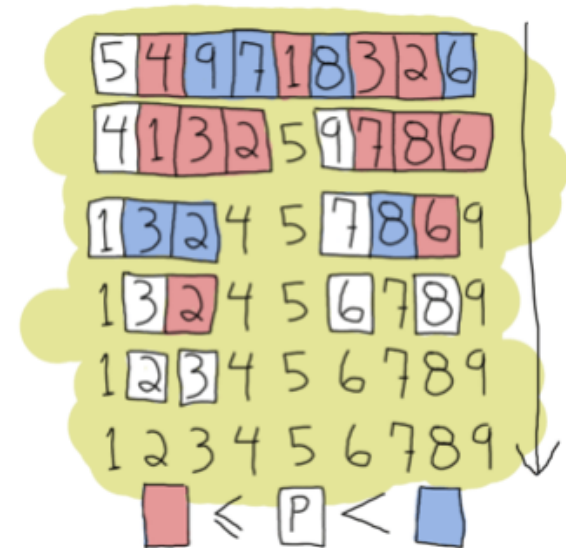
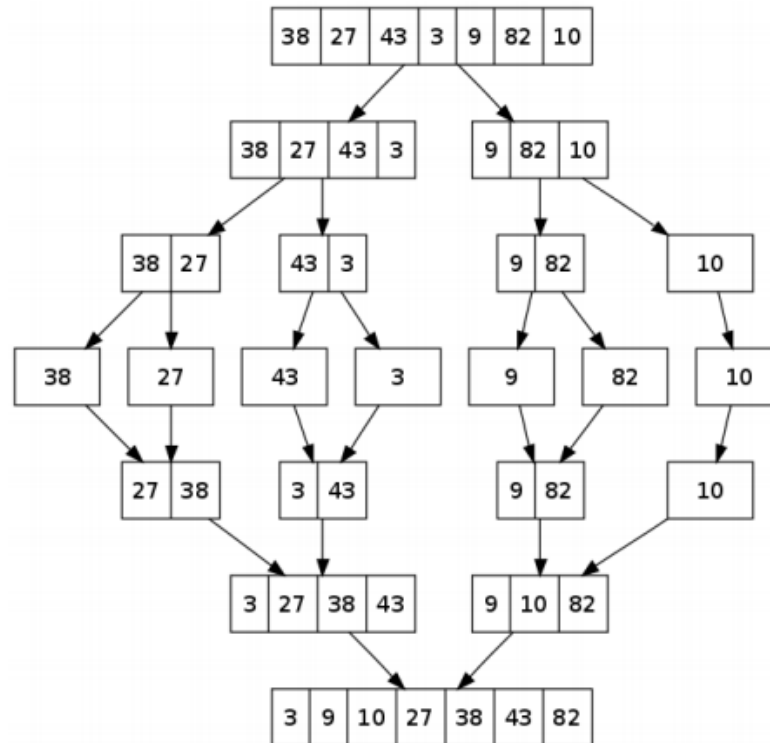
- Aantal elementen: n
- Aantal benodigde stappen: $T(n)$
- Naïeve algoritme: $T(n) = 0,5 \cdot n = O(n)$
- Binary search: $T(n) = O(\log_2 n)$

Binary tree



Sorteeralgoritmen

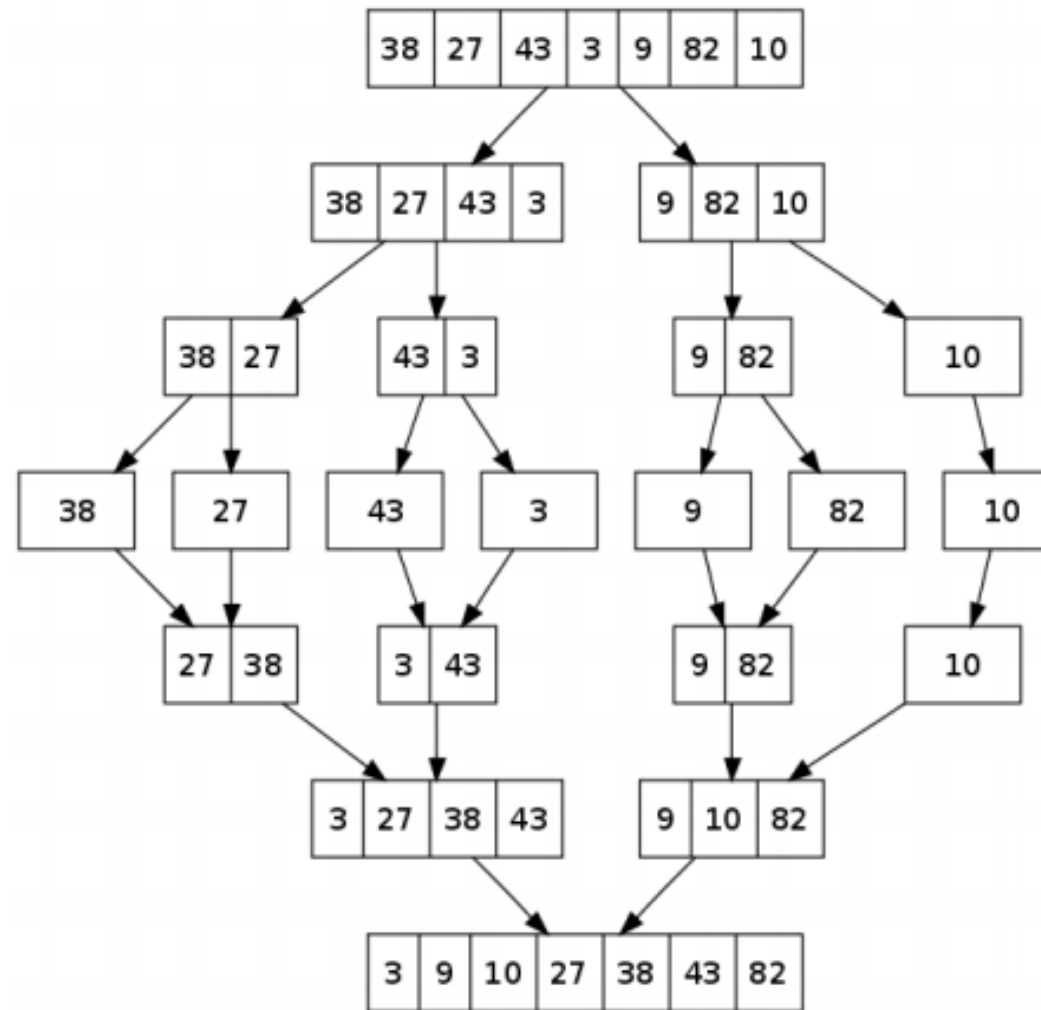
5 3 1 6 8 7 2 4



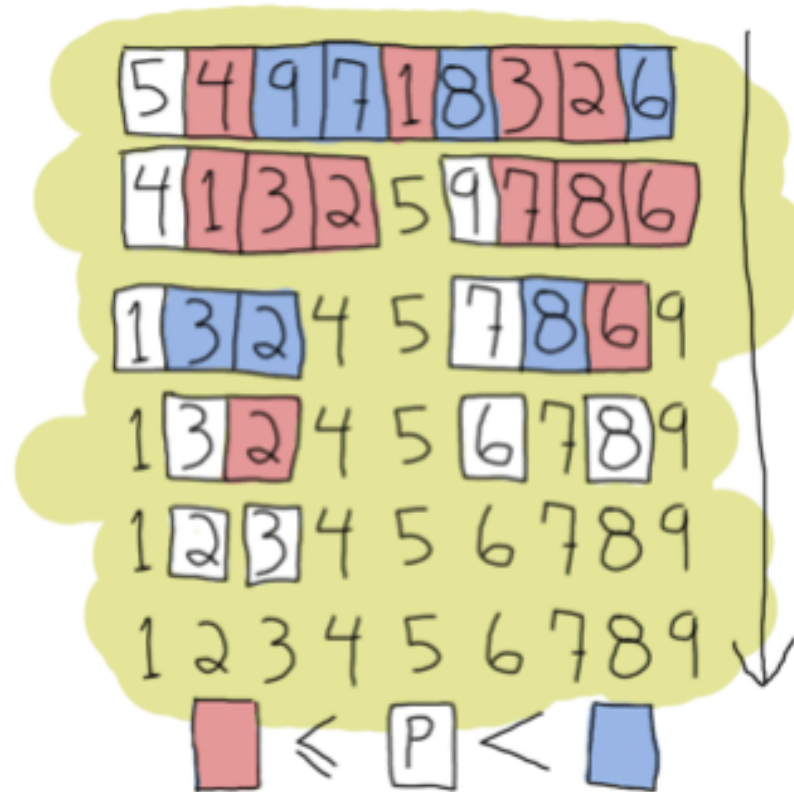
BubbleSort

5 3 1 6 8 7 2 4

MergeSort



QuickSort



Vragen

- Sorteer de volgende lijst (handmatig) met:
 - BubbleSort
 - MergeSort
 - QuickSort

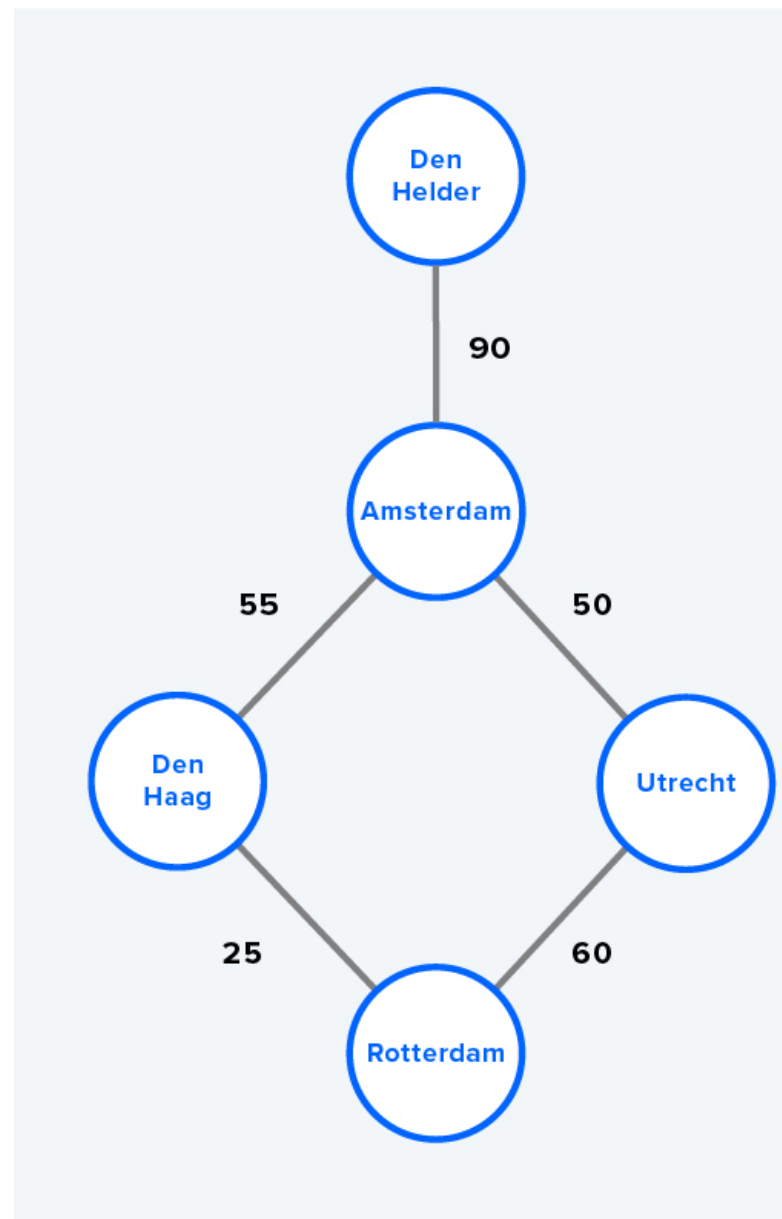
3 7 1 4 8 3

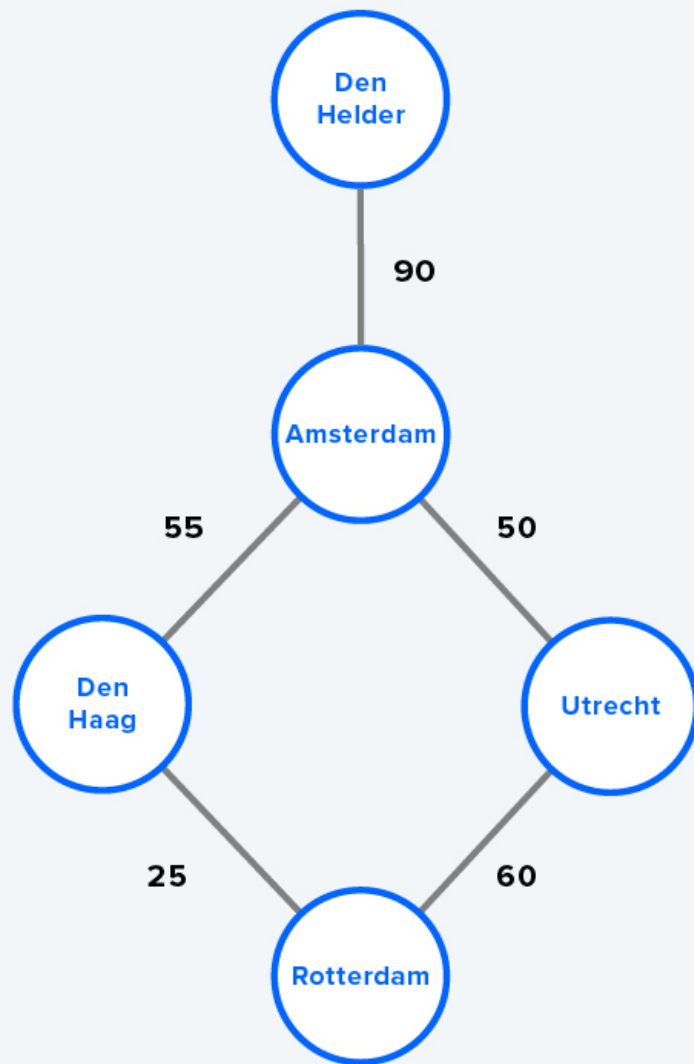
Vraag: minimum / maximum

- Beschrijf zelf een algoritme dat het minimum van een ongesorteerde lijst kan bepalen
 - Kan het efficiënter, waarom wel of niet?
 - Wat is de orde?

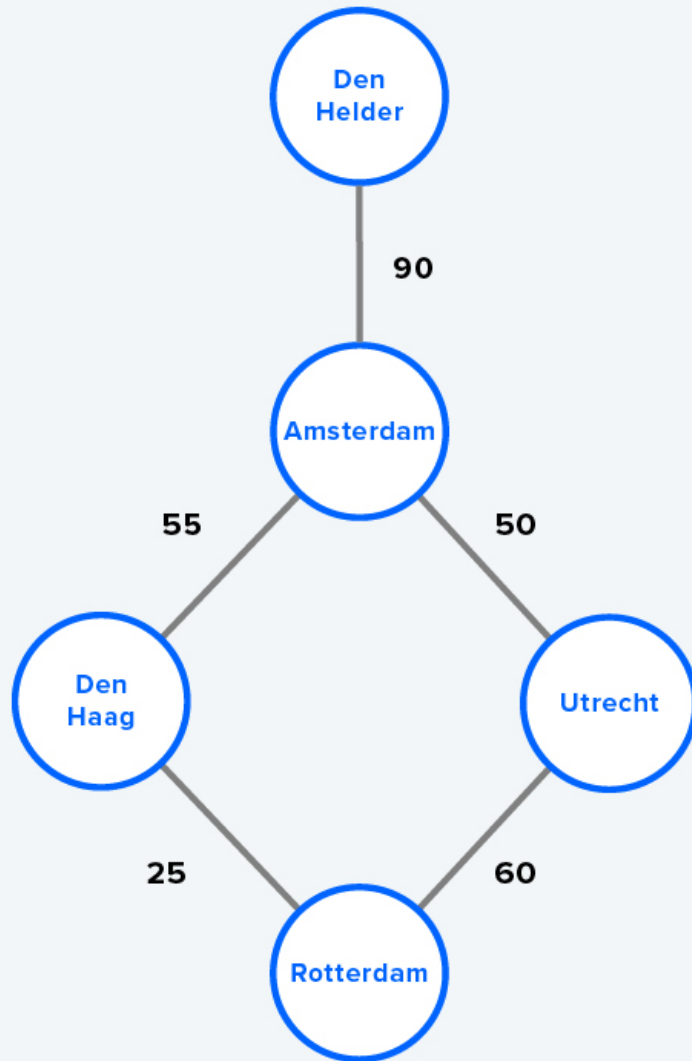
Vraag: minimum / maximum

- Kun je een efficiënt algoritme schrijven dat minimum en maximum bepaalt?





Plaats	Kortste afstand tot Den Helder	Vorige plaats
Den Helder	∞	-
Amsterdam	∞	-
Den Haag	∞	-
Utrecht	∞	-
Rotterdam	∞	-

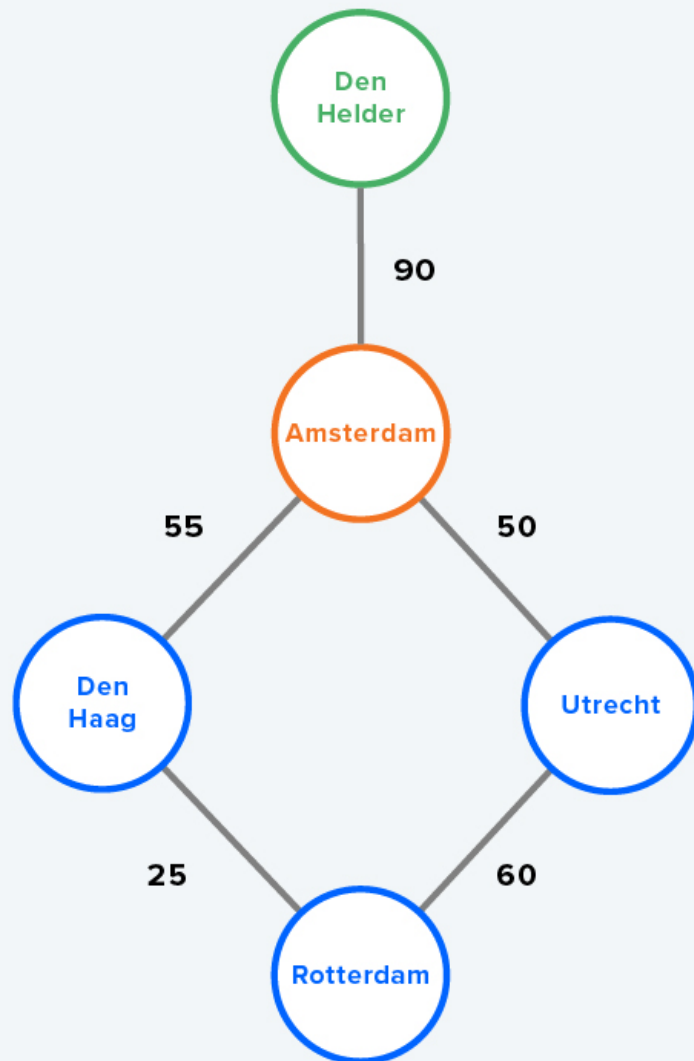


Plaats	Kortste afstand tot Den Helder	Vorige plaats
Den Helder	0	-
Amsterdam	90	Den Helder
Den Haag	∞	-
Utrecht	∞	-
Rotterdam	∞	-

Bereken vanuit **Den Helder** alle afstanden tot de direct verbonden maar **onbezochte plaatsen**.

Zijn er gevonden afstanden kleiner dan de waarden in de tabel? Vervang die.

Alle plaatsen gehad? Markeer Den Helder als bezocht.

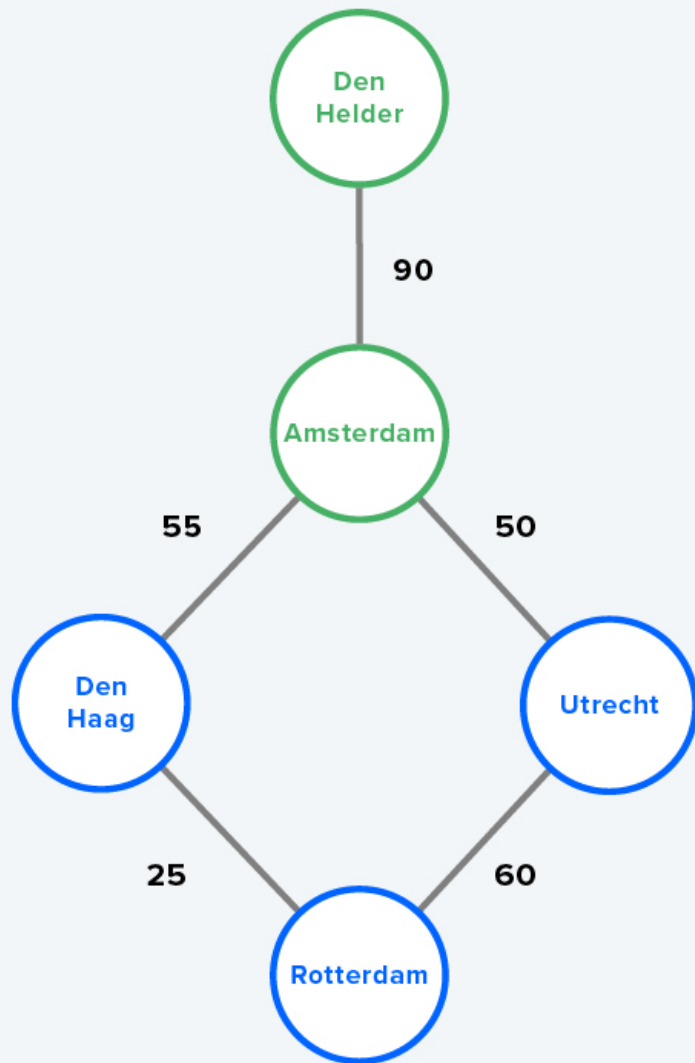


Plaats	Kortste afstand tot Den Helder	Vorige plaats
Den Helder	0	-
Amsterdam	90	Den Helder
Den Haag	$90 + 55 = 145$	Amsterdam
Utrecht	$90 + 50 = 140$	Amsterdam
Rotterdam	∞	-

Kies de stad die nog niet ‘bezoekt’ is en het dichtst bij Den Helder ligt.

Dat is Amsterdam.

Bereken vanuit **Amsterdam** alle afstanden tot de direct verbonden maar **onbezochte plaatsen**. Zijn er gevonden afstanden kleiner dan de waarden in de tabel? Vervang die.

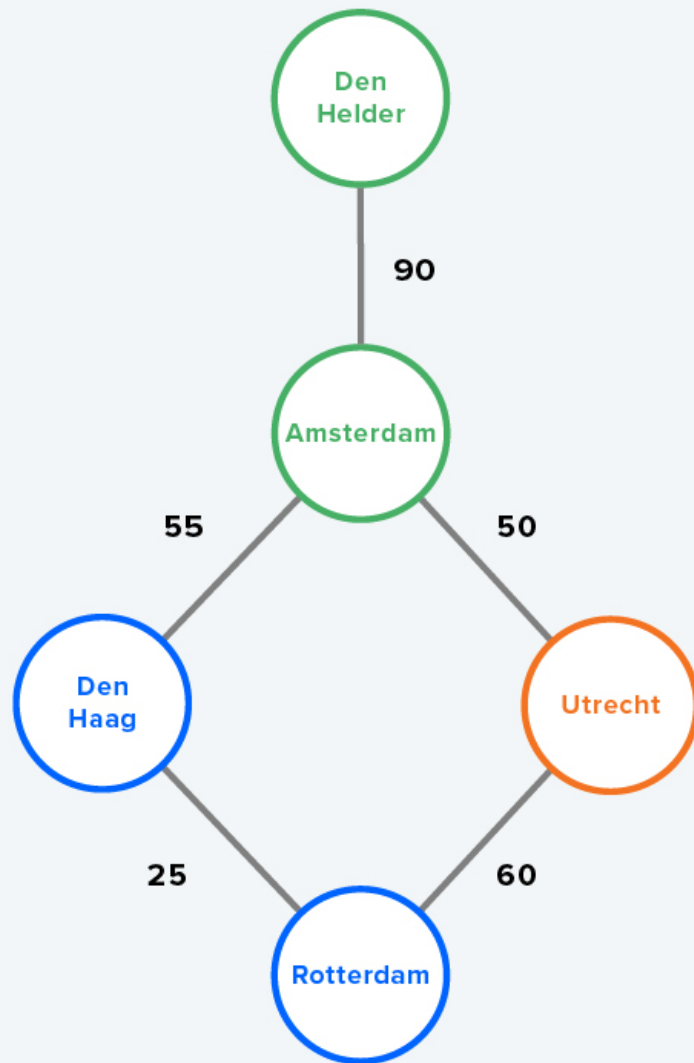


Plaats	Kortste afstand tot Den Helder	Vorige plaats
Den Helder	0	-
Amsterdam	90	Den Helder
Den Haag	145	Amsterdam
Utrecht	140	Amsterdam
Rotterdam	∞	-

Alle plaatsen gehad? Markeer Amsterdam als **bezocht**.

Kies de stad die nog niet '**bezocht**' is en het dichtst bij Den Helder ligt.

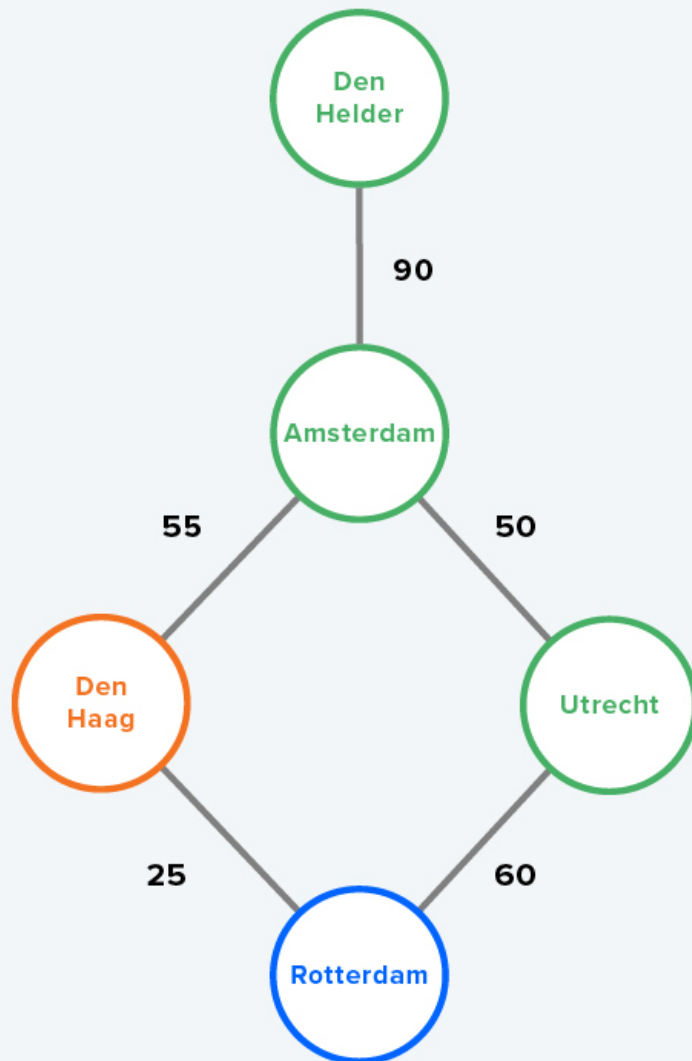
Dat is **Utrecht**.



Plaats	Kortste afstand tot Den Helder	Vorige plaats
Den Helder	0	-
Amsterdam	90	Den Helder
Den Haag	145	Amsterdam
Utrecht	140	Amsterdam
Rotterdam	$140 + 60 = 200$	Utrecht

Bereken vanuit **Utrecht** alle afstanden tot de direct verbonden maar **onbezochte** plaatsen.

Zijn er gevonden afstanden kleiner dan de waarden in de tabel? Vervang die.

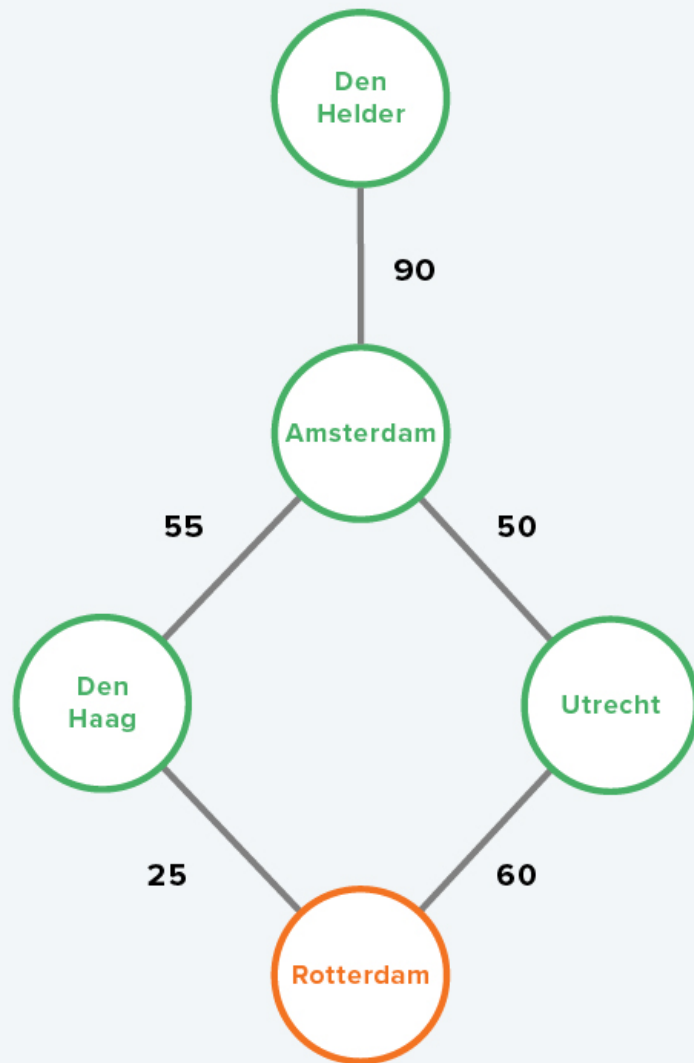


Plaats	Kortste afstand tot Den Helder	Vorige plaats
Den Helder	0	-
Amsterdam	90	Den Helder
Den Haag	145	Amsterdam
Utrecht	140	Amsterdam
Rotterdam	200	Utrecht

Alle plaatsen gehad? Markeer Utrecht als **bezocht**.

Kies de stad die nog niet '**bezocht**' is en het dichtst bij Den Helder ligt.

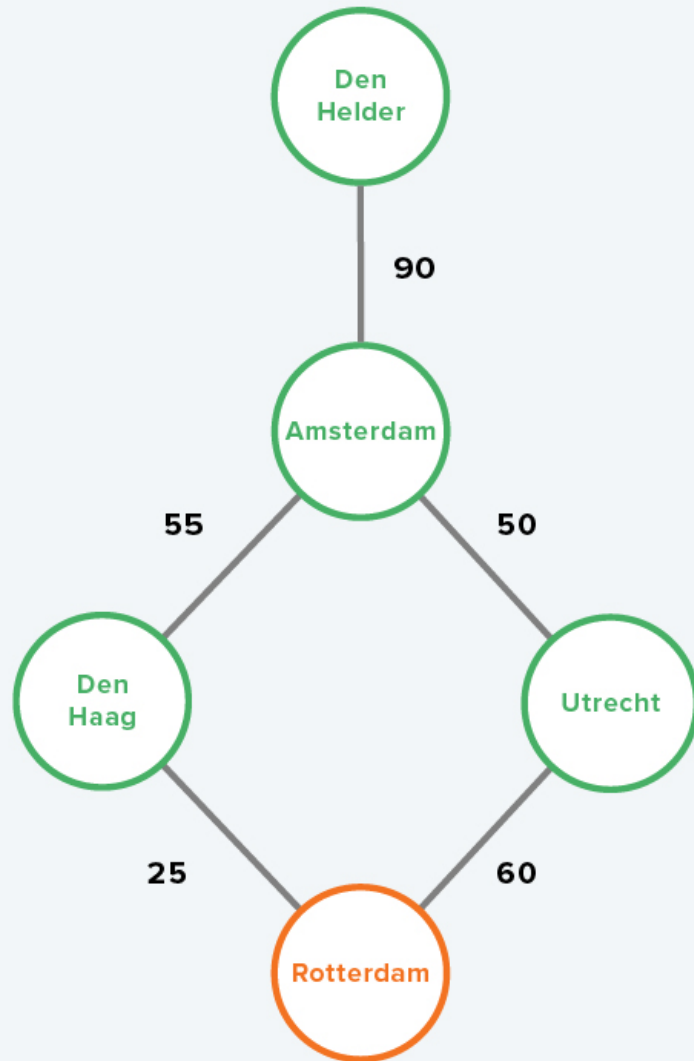
Dat is **Den Haag**.



Plaats	Kortste afstand tot Den Helder	Vorige plaats
Den Helder	0	-
Amsterdam	90	Den Helder
Den Haag	145	Amsterdam
Utrecht	140	Amsterdam
Rotterdam	$145 + 25 = 170$	Den Haag

Bereken vanuit **Den Haag** alle afstanden tot de direct verbonden maar **onbezochte** plaatsen.

Zijn er gevonden afstanden kleiner dan de waarden in de tabel? Vervang die.

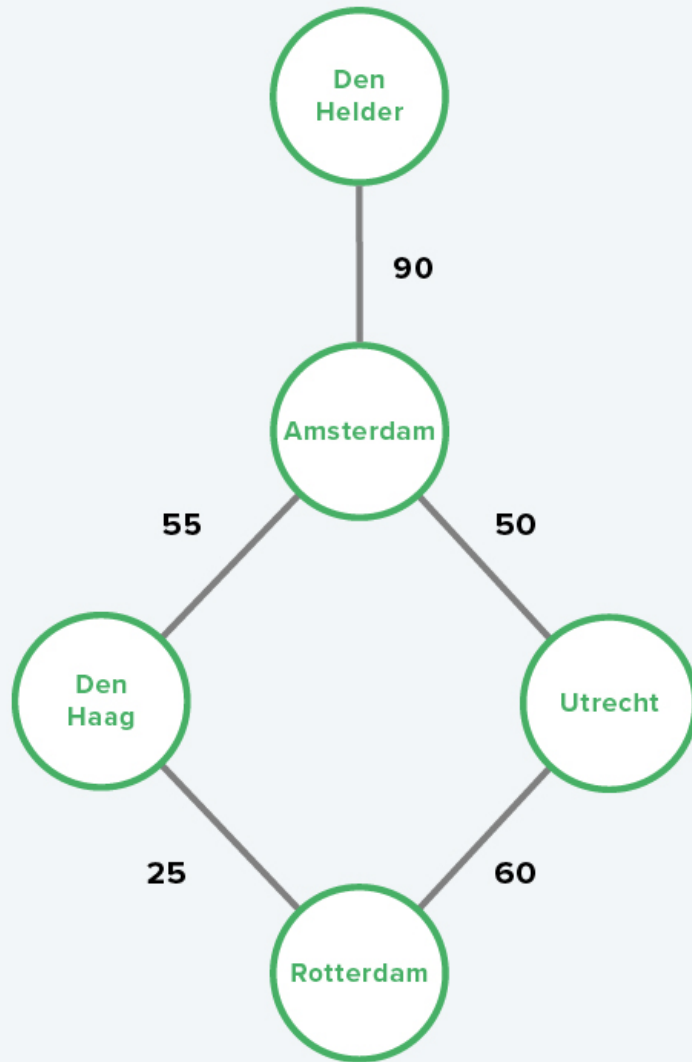


Plaats	Kortste afstand tot Den Helder	Vorige plaats
Den Helder	0	-
Amsterdam	90	Den Helder
Den Haag	145	Amsterdam
Utrecht	140	Amsterdam
Rotterdam	$145 + 25 = 170$	Den Haag

Alle plaatsen gehad? Markeer Den Haag als **bezocht**.

Alleen Rotterdam is nog **onbezocht**. Maar dat is de eindbestemming.

We zijn klaar!



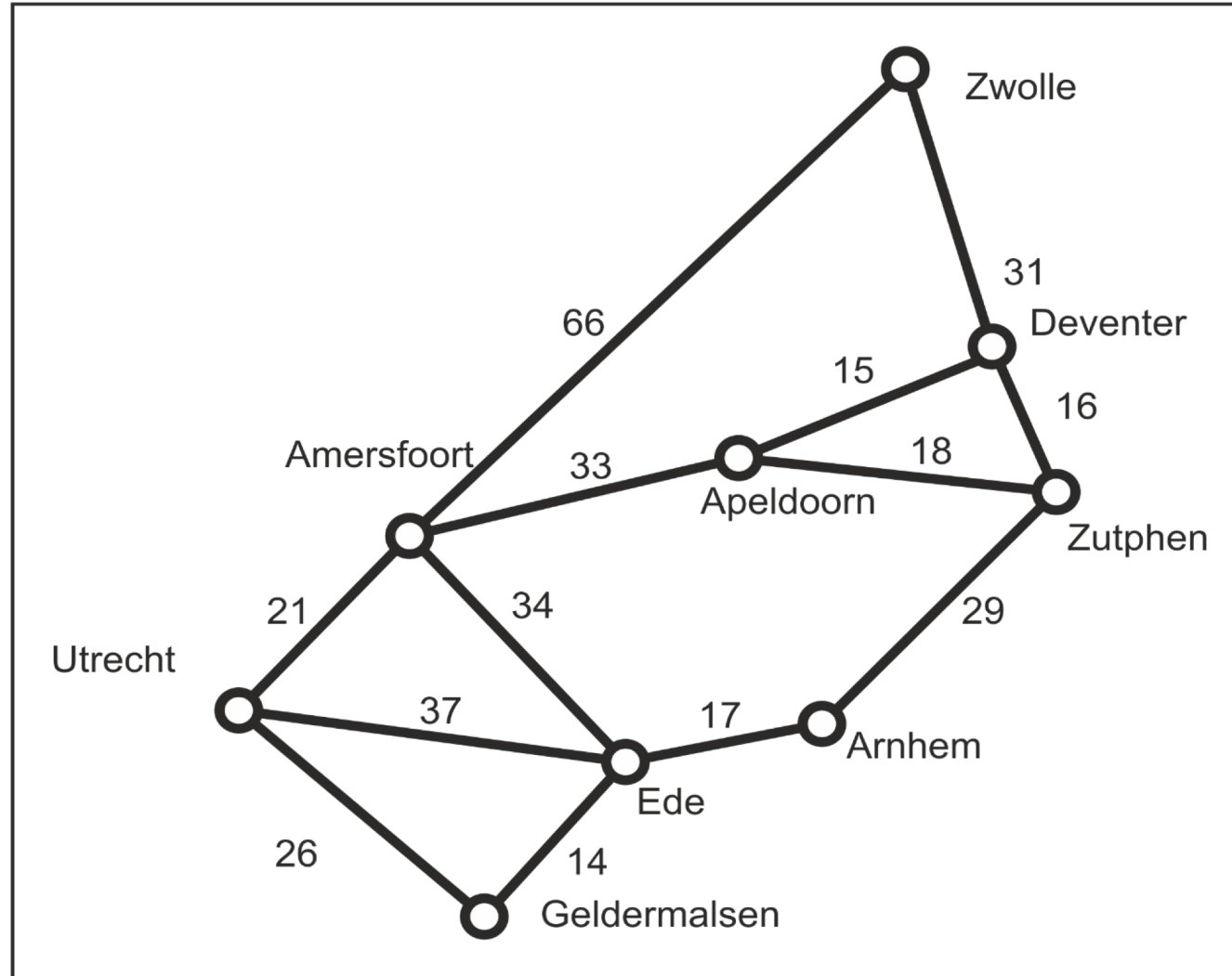
Plaats	Kortste afstand tot Den Helder	Vorige plaats
Den Helder	0	-
Amsterdam	90	Den Helder
Den Haag	145	Amsterdam
Utrecht	140	Amsterdam
Rotterdam	170	Den Haag

Wat is de route?

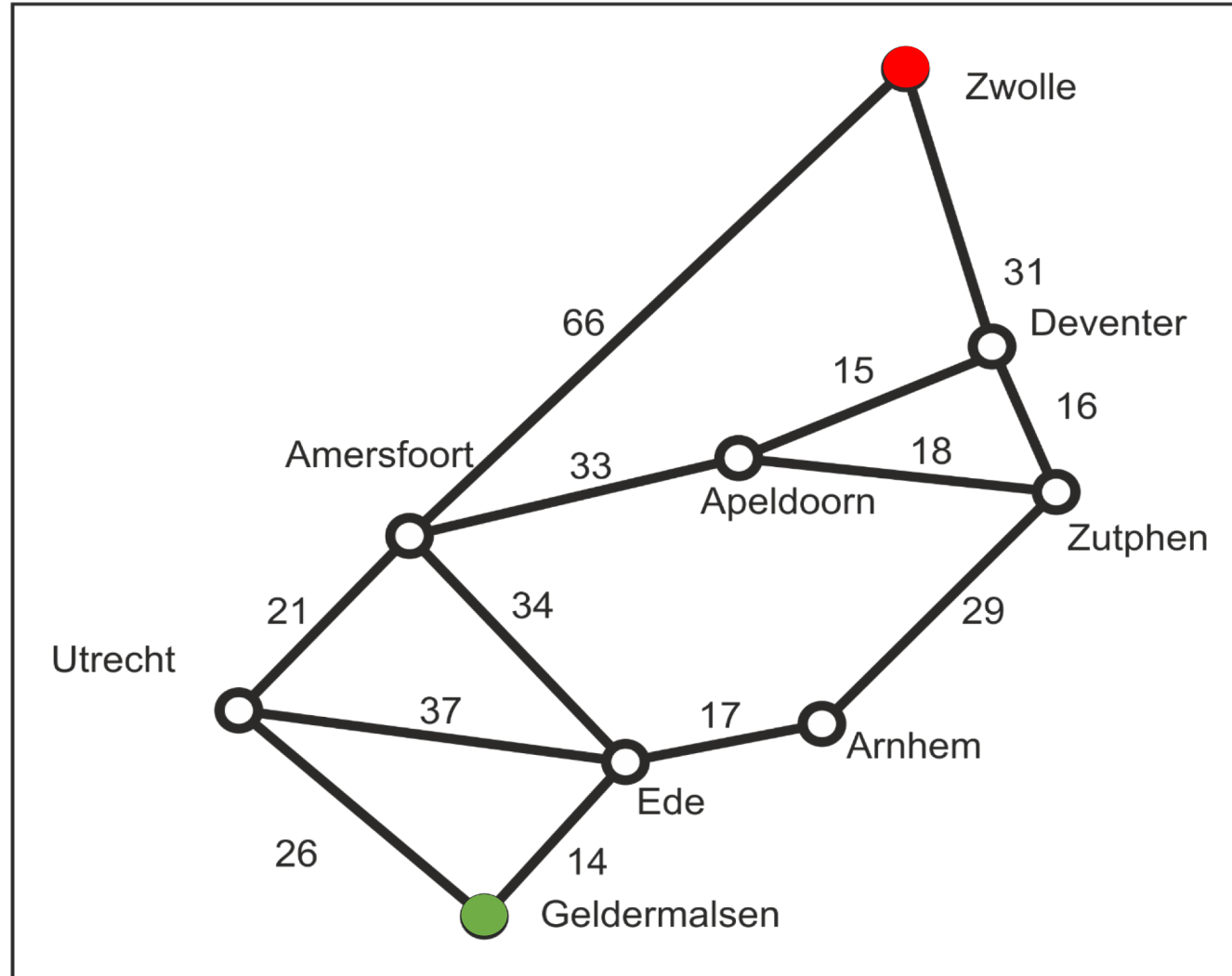
Bepalen vanuit de eindbestemming:

Rotterdam – Den Haag – Amsterdam – Den Helder

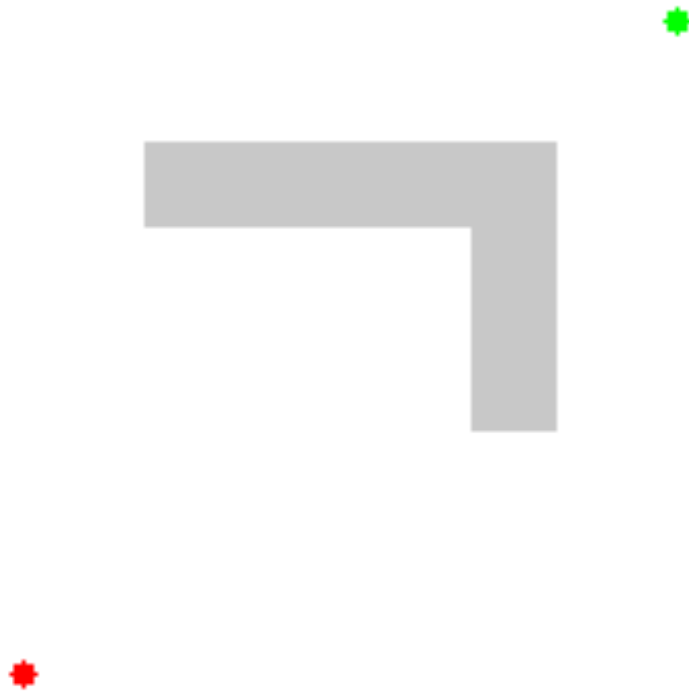
Kortstepadalgoritme



Kortstepadalgoritme



A*



Brute force en complexe problemen

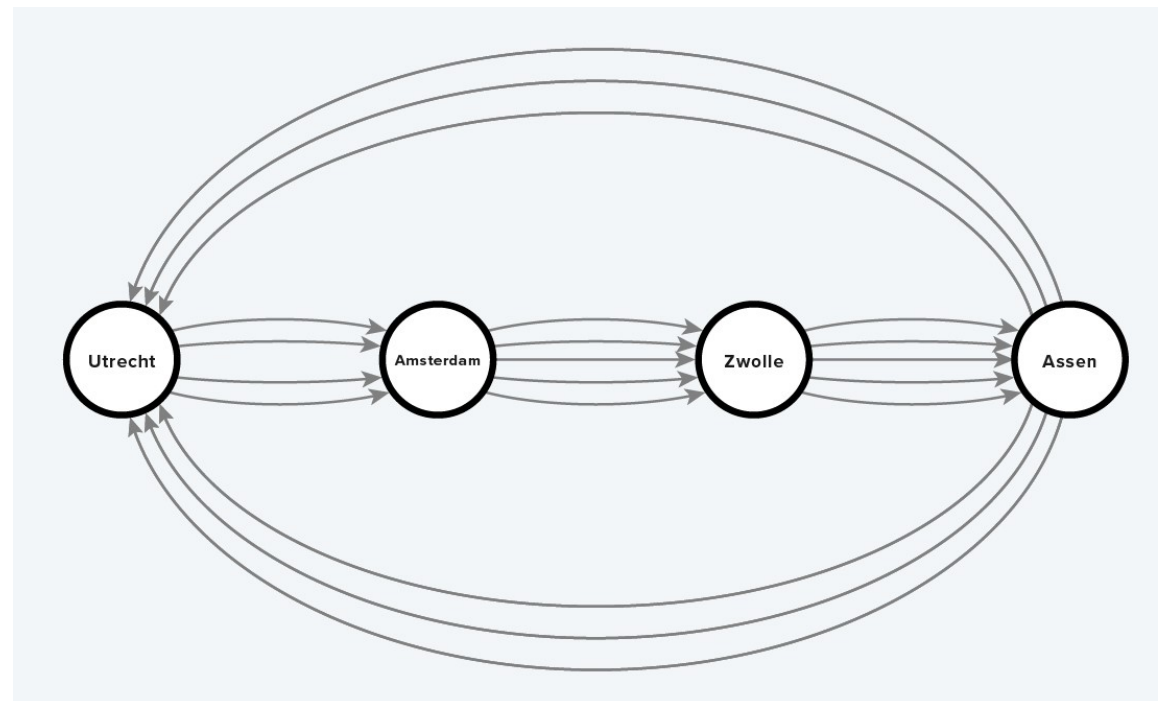
- Correctheid en efficiëntie
- Rugzakprobleem
- Chinese postbodeprobleem

Correctheid en efficiëntie

- Alle mogelijkheden langsgaan
 - *Brute force*
- De oplossing benaderen

Vraag

- Een vrachtwagen moet spullen bezorgen in Amsterdam, Zwolle en Assen. Er zijn verschillende routes mogelijk. Hoeveel mogelijke routes moet het algoritme onderzoeken met brute force?

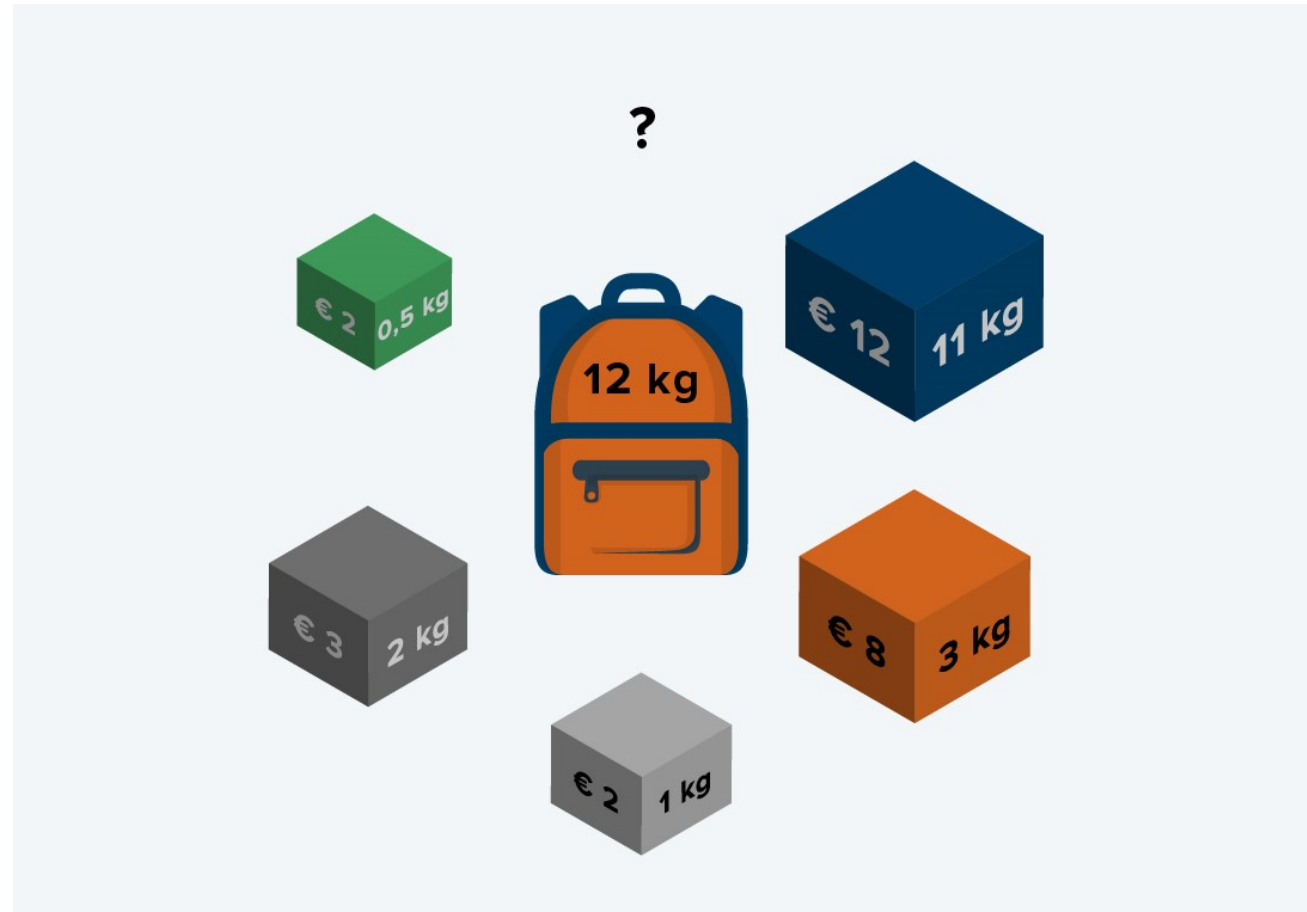


Vraag

- Je hebt drie sorteeralgoritmen gezien: BubbleSort, MergeSort en QuickSort.

Welke van de drie lijkt erg veel op een brute-force-algoritme?

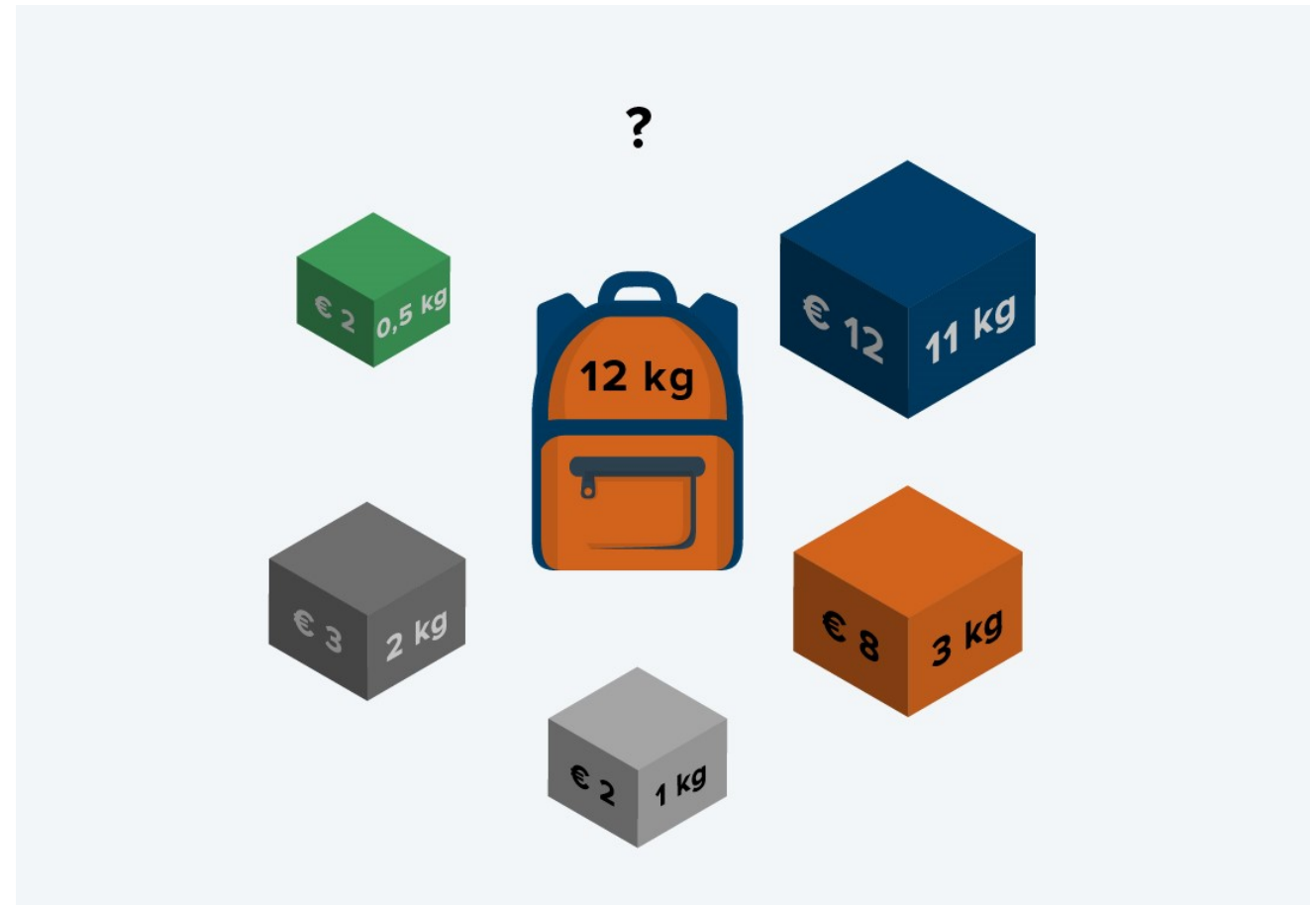
Rugzakprobleem



Rugzakprobleem oplossen

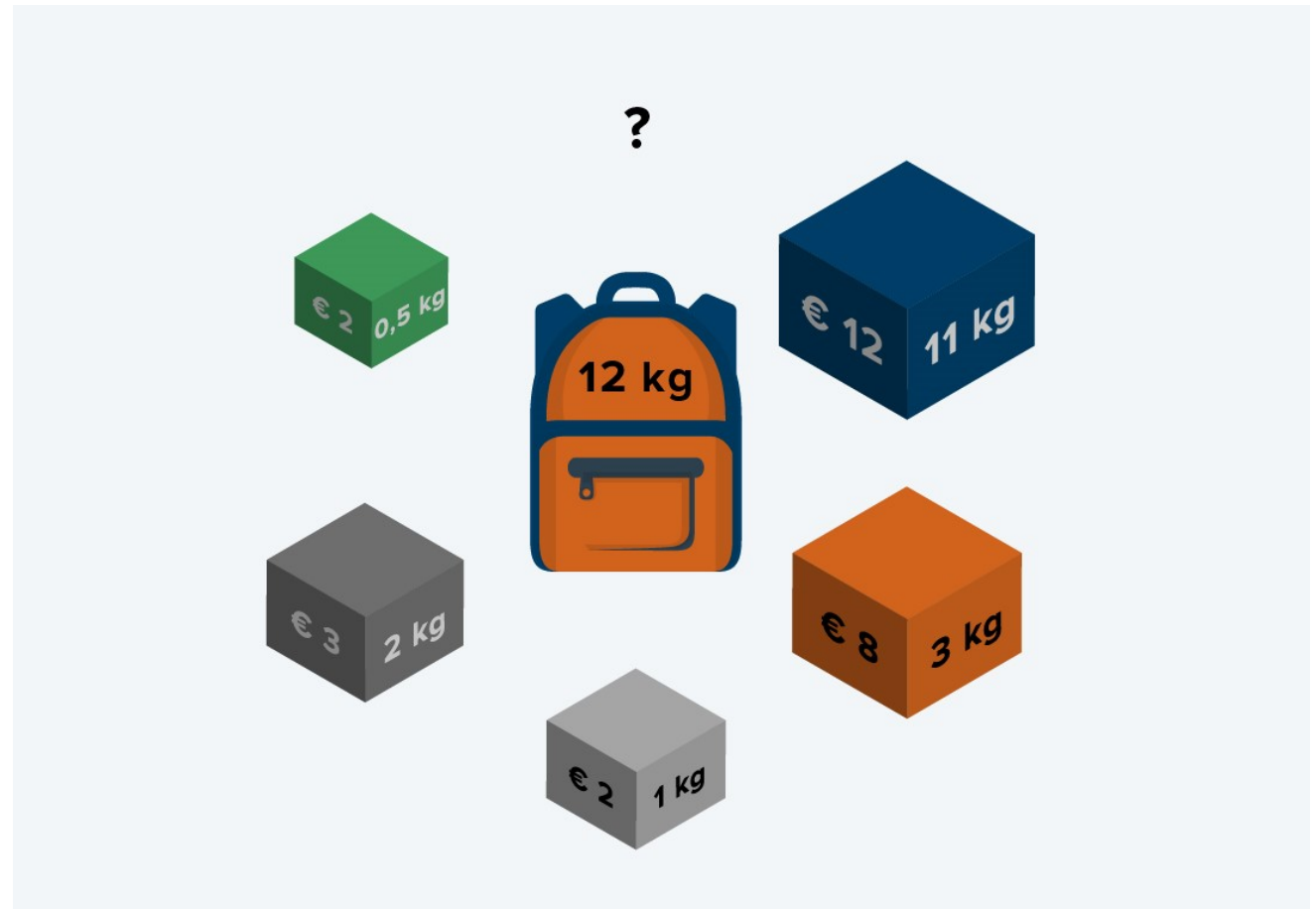
Oplossing van Dantzig

1. Bereken voor elke doos de waarde per kilogram
2. Sorteert de dozen vervolgens op die waarde van groot naar klein
3. Vul de rugzak in diezelfde volgorde tot het niet meer past

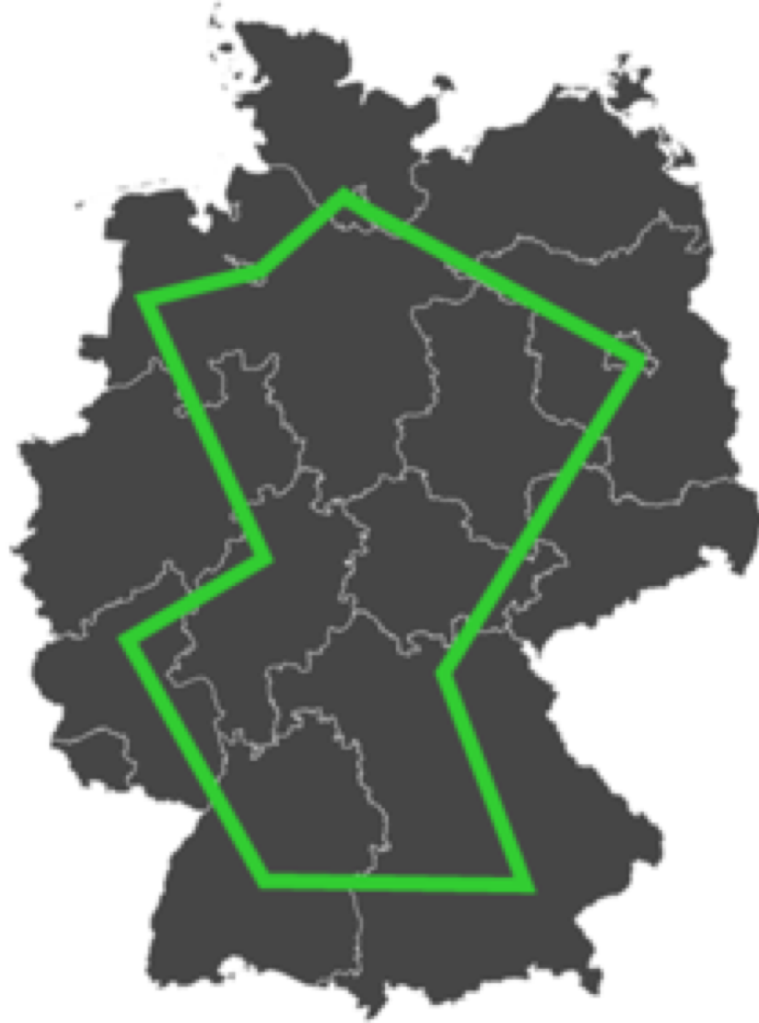


Rugzakprobleem oplossen

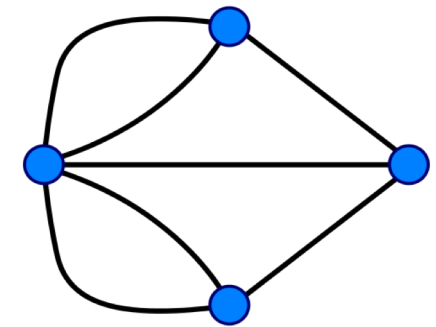
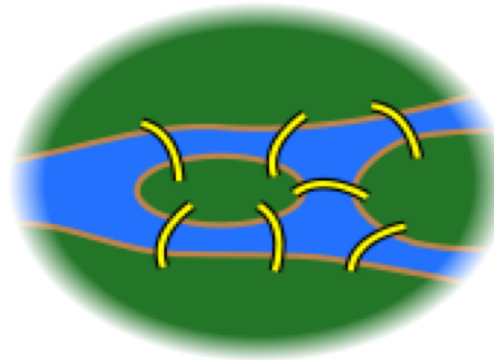
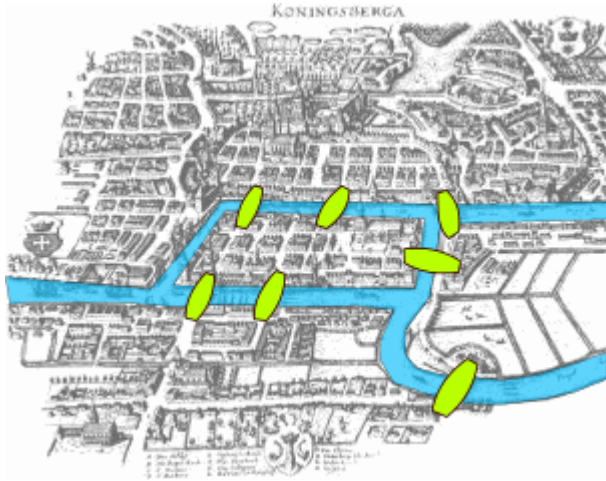
Welke dozen neem je volgens Dantzig mee?



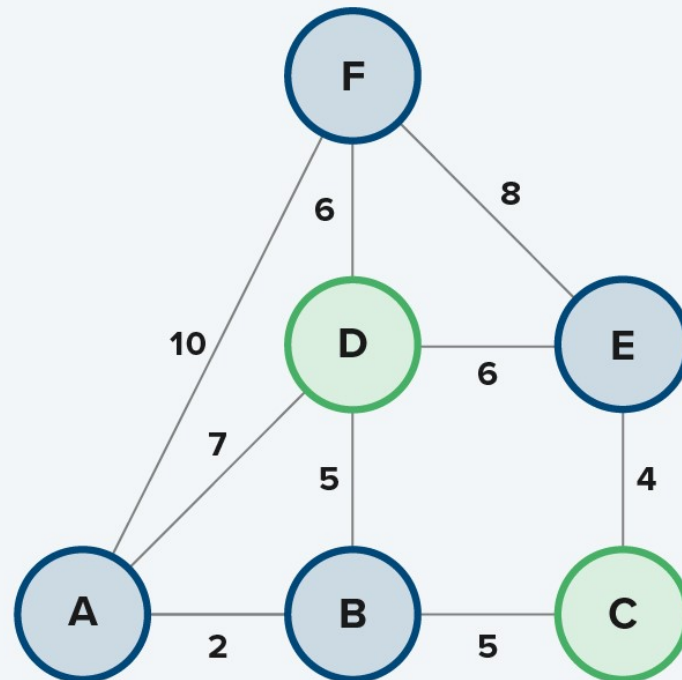
Handelsreizigersprobleem



“Zeven bruggen van Koningsbergen”



Chinese postbodeprobleem

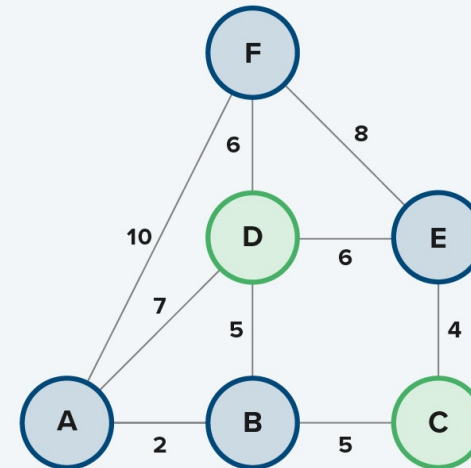
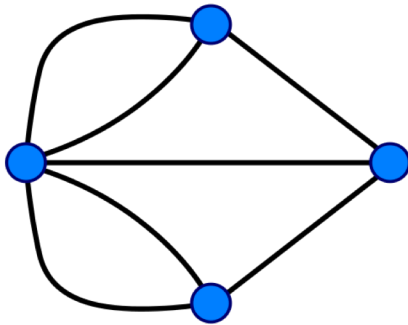


Stelling

“Als er meer dan twee kruispunten met een oneven aantal aangesloten straten zijn, moeten sommige straten meerdere keren worden doorlopen.”

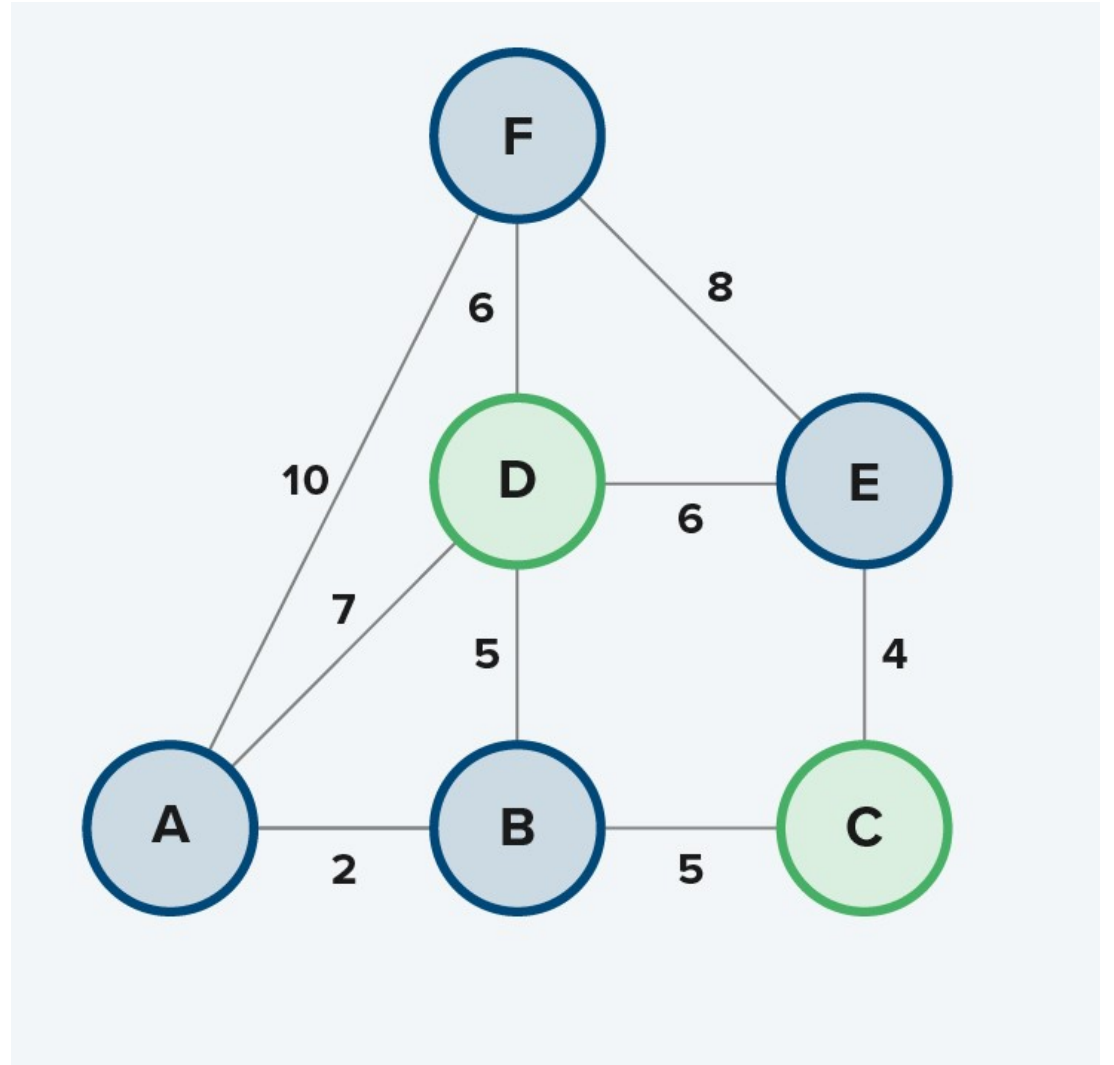
Stelling

“Als er meer dan twee kruispunten met een oneven aantal aangesloten straten zijn, moeten sommige straten meerdere keren worden doorlopen.”



Vraag

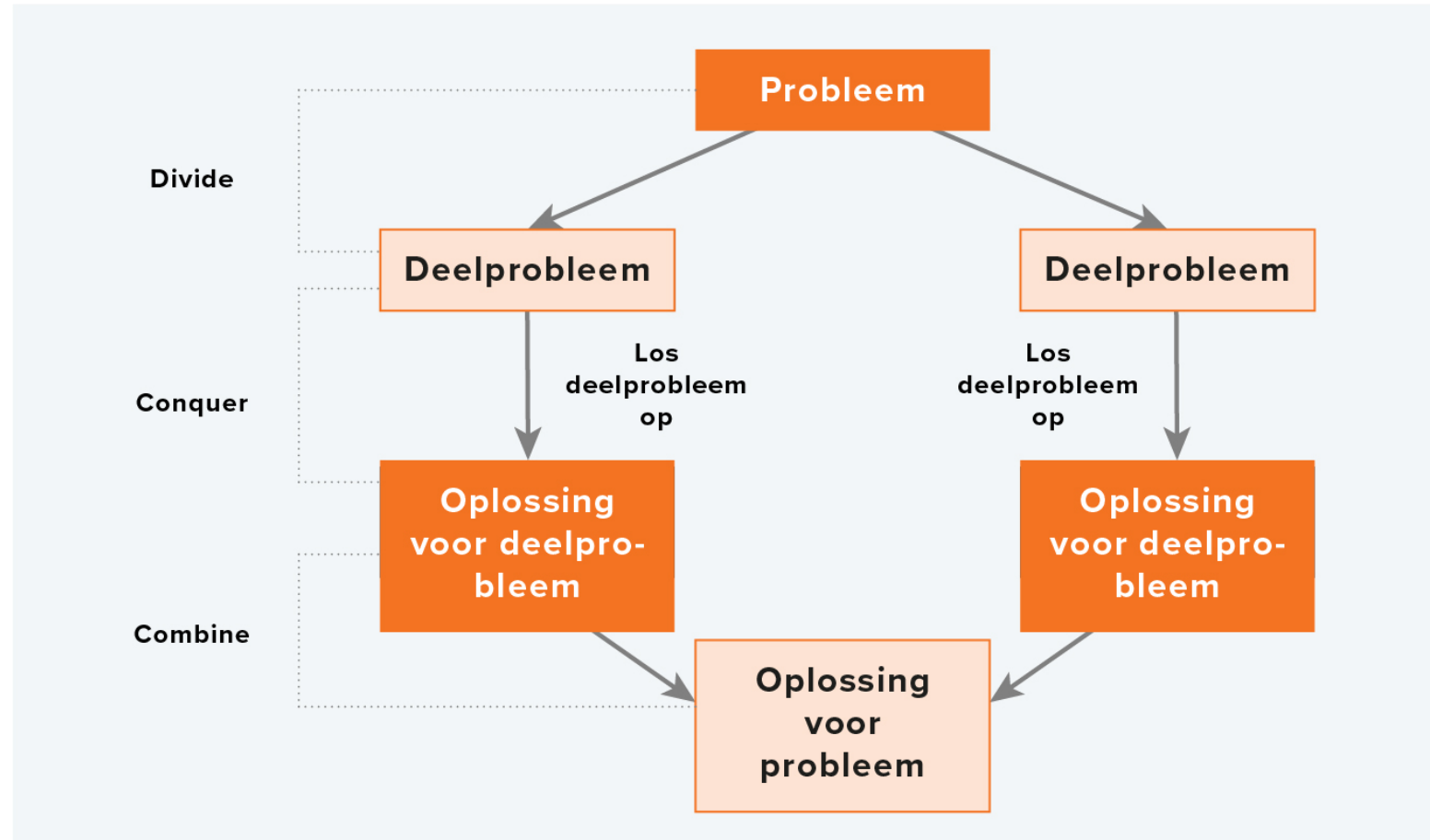
Hoe doorloopt de postbode het snelste deze straten?



PAUZE

15 minuten

Divide-and-conquer

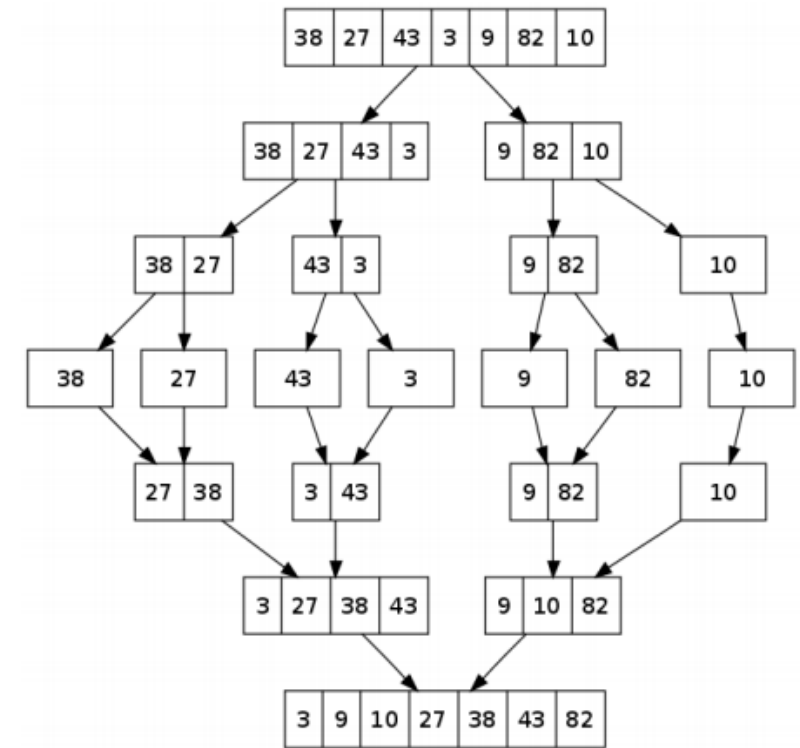


Divide-and-conquer

- MergeSort
- QuickSort
- Wat is de orde?

Orde van efficiëntie van divide-and-conquer

- $T(n) = T(n/2) + T(n/2) + O(n)$
- $T(n) = 2 \cdot T(n/2) + O(n)$
- $T(n) = T(n/3) + T(2n/3) + O(n)$
- $T(n) = 3 \cdot T(n/3) + O(n)$



Orde van efficiëntie van divide-and-conquer

- Recursieve formule: $T(n) = 2 \cdot T(n/2) + O(n)$
- $T(n) = O(n \cdot \log n)$

Ordes van efficiëntie – van snel naar langzaam

- Constant: $T(n) = O(1)$ Minimum van gesorteerde lijst
- Logaritmisch: $T(n) = O(\log n)$ Binary Search
- Lineair: $T(n) = O(n)$ Minimum van ongesorteerde lijst
- Loglineair: $T(n) = O(n \cdot \log n)$ Meeste divide-and-conquer alg.
- Kwadratisch: $T(n) = O(n^2)$ BubbleSort
- Exponentieel: $T(n) = O(c^n)$ Handelsreizigersprobleem

Implementatie van sorteeralgoritmen

- BubbleSort
 - Geschikt voor HAVO
 - Goede reden om functies te gebruiken
- QuickSort
 - Recursie nodig
 - Geschikt voor VWO

Recursie

- Functie met twee eigenschappen:
 - Roept zichzelf aan
 - Heeft een stopcriterium

QuickSort programmeren

- Verdeel elke lijst over drie lijsten:
- Functie met twee eigenschappen:
 - Roept zichzelf aan
 - Heeft een stopcriterium

QuickSort programmeren

- Verdeel elke lijst over drie lijsten:

3 7 1 4 8 3

- Linkerlijst
- Pivotlijst
- Rechterlijst

QuickSort programmeren

- Verdeel elke lijst over drie lijsten:
 - Linkerlijst
 - Pivotlijst
 - Rechterlijst
- Maak eerst pseudocode of woorden
 - Denk aan het stopcriterium

QuickSort in pseudocode

```
QuickSort(lijst):  
    is lengte(lijst) < 2? return lijst  
    pivot = lijst[0]  
    voor elke element in de lijst:  
        is element < pivot?  
            toevoegen aan linkerlijst  
        is element == pivot?  
            toevoegen aan pivotlijst  
    anders:  
        toevoegen aan rechterlijst  
    voer quicksort uit op linkerlijst  
    voer quicksort uit op rechterlijst  
    voeg linker-, pivot- en rechterlijst samen  
    return het resultaat
```

EINDE

Bedankt!