**1a** filmpje dijkstra: <https://www.youtube.com/watch?v=WN3Rb9wVYDY>

Code dijkstra: <https://stackoverflow.com/questions/22897209/dijkstras-algorithm-in-python>

1e reactie op website: dijkstra code

2e reactie op website: van begin station route naar gevraagd station

**AANPAK:**

Handmatig max treinen + max minuten instellen

1 traject per keer:

* Opstellen: kortste ongebruikte kritieke pad
* Zo ver mogelijk uitbreiden met kortste ongebruikte kritieke paden

Alle trajecten gemaakt:

* Alle ongebruikte, kritieke paden proberen te koppelen aan 1 van de trajecten mbv dijkstra

|  |  |  |  |
| --- | --- | --- | --- |
| **AANTAL TREINEN** | **MAX MINUTEN PER TREIN** | **DOELFUNCTIE** | **AANTAL KRITIEKE CONNECTIES NIET INGEDEELD** |
| 7 | 120 | 9829,1 | 0 |
| 7 | 100 | 9829,2 | 0 |
| 7 | 90 | 9828,6 | 0 |
| 7 | 79 | 9334,9 | 1 |
|  |  |  |  |
| 6 | 120 | 9847,7 | 0 |
| 6 | 115 | 9847,8 | 0 |
| 6 | 90 | 9847,2 | 0 |
| 6 | 85 | 9846,8 | 0 |
| 6 | 79 | 9353,1 | 1 |
|  |  |  |  |
| 5 | 120 | 9867,2 | 0 |
| 5 | 115 | 9867,3 | 0 |
| 5 | 107 | 9866,9 | 0 |
| 5 | 90 | 9866,3 | 0 |
| 5 | 79 | 9372,6 | 1 |
|  |  |  |  |
| 4 | 116 | 9886,1 | 0 |
| 4 | 115 | 9881,4 | 0 |
| 4 | 107 | 9388,2 | 1 |
|  |  |  |  |
| 3 | 120 | 9406,7 | 1 |

RANDOM BEGINCONNECTIE

*Sub play()*

*i = 0*

*y = 10000*

*For Each Row In ActiveSheet.Rows*

*If Cells(Row.Row, 1).Value Like "\*Doelwaarde:\*" Then*

*u = CDbl(Mid(Cells(Row.Row + 1, 1).Value, 1, 6)) / 10*

*If u >= 9888 Then*

*Cells(Row.Row + 1, 1).Interior.Color = RGB(0, 255, 0)*

*ElseIf u < 9700 Then*

*Cells(Row.Row + 1, 1).Interior.Color = RGB(255, 0, 0)*

*End If*

*If u > i Then*

*i = u*

*End If*

*If u < y Then*

*y = u*

*End If*

*End If*

*Next*

*Cells(1, 10).Value = i*

*Cells(1, 11).Value = y*

*End Sub*

**4, 120** met random beginstation **50.000x** gerunt, geexporteerd naar excel, met vba analyseren.

**Best:**

|  |
| --- |
| 4 120 |
|  |
| ['Gouda', 'Alphen a/d Rijn'] 19 |
| ['Alkmaar', 'Den Helder'] 36 |
| ['Delft', 'Den Haag Centraal'] 13 |
| ['Alkmaar', 'Hoorn'] 24 |
|  |
| ['Dordrecht', 'Rotterdam Centraal', 'Schiedam Centrum', 'Rotterdam Centraal', 'Rotterdam Alexander', 'Gouda', 'Alphen a/d Rijn'] 64 |
| ['Amsterdam Amstel', 'Amsterdam Centraal', 'Amsterdam Sloterdijk', 'Zaandam', 'Castricum', 'Alkmaar', 'Den Helder'] 77 |
| ['Delft', 'Den Haag Centraal', 'Leiden Centraal', 'Den Haag Centraal', 'Gouda'] 55 |
| ['Alkmaar', 'Hoorn', 'Zaandam', 'Beverwijk', 'Haarlem', 'Heemstede-Aerdenhout', 'Haarlem', 'Amsterdam Sloterdijk'] 114 |
|  |
| Doelwaarde: |
| 9889.0 |
|  |

**Slechtst:**

|  |
| --- |
| ['Gouda', 'Alphen a/d Rijn'] 19 |
| ['Den Haag Centraal', 'Gouda'] 18 |
| ['Delft', 'Den Haag Centraal'] 13 |
| ['Dordrecht', 'Rotterdam Centraal'] 17 |
| Haarlem - Beverwijk |
| Heemstede-Aerdenhout - Haarlem |
| Zaandam - Castricum |
| Zaandam - Beverwijk |
| Zaandam - Hoorn |
|  |
| ['Schiedam Centrum', 'Rotterdam Centraal', 'Rotterdam Alexander', 'Gouda', 'Alphen a/d Rijn', 'Leiden Centraal', 'Schiphol Airport', 'Amsterdam Zuid', 'Amsterdam Amstel', 'Amsterdam Centraal', 'Amsterdam Sloterdijk', 'Amsterdam Centraal', 'Amsterdam Sloterdijk', 'Zaandam'] 119 |
| ['Hoorn', 'Alkmaar', 'Castricum', 'Beverwijk', 'Haarlem', 'Heemstede-Aerdenhout', 'Leiden Centraal', 'Den Haag Centraal', 'Gouda'] 111 |
| ['Delft', 'Den Haag Centraal', 'Leiden Centraal', 'Heemstede-Aerdenhout', 'Haarlem', 'Beverwijk', 'Castricum', 'Alkmaar', 'Den Helder'] 118 |
| ['Dordrecht', 'Rotterdam Centraal', 'Schiedam Centrum', 'Delft', 'Den Haag Centraal', 'Leiden Centraal', 'Heemstede-Aerdenhout', 'Haarlem', 'Amsterdam Sloterdijk', 'Zaandam', 'Castricum', 'Alkmaar'] 111  7374.1  **4 120 Random min:** 9889,0. 7374,1 5 niet ingedeeld (3x zaandam, 2x haarlem)  **3 120 Random min:** 9409,4 1 niet ingedeeld. 5905,2  **5 120 Random min:** 9870,2. 9344,6  **4 120 Random max:** 9888,3. 7374,1 5 niet ingedeeld (3x zaandam, 2x haarlem)  **3 120 Random max:** 8908,7 2 niet ingedeeld (rdam en den haag). 5405,2 9.  **5 120 Random max:** 9869,5. 9344,6 heemstede-haarlem.  **4 120 min random:** 9888,6. 8877,3 (2x zaandam)  **3 120 min random:** 9903,8. 5908,3 8:(3x den haag, 3x gouda, 2x zaandam, rdam)  **5 120 min random:** 9869,6. 9855,1  10.000x  **4 120 max random:** 9888,9. 9374,6. (zaandam-castricum)  **3 120 max random:** 9406,8 (dordrecht-rdam). 6906,1. 6:(haarlem, castricum, rdam)  **5 120 max random:** 9870,0. 9858,4.  10.000x  **4 120 random random:** 9889,0. 7376,1 (3x zaandam, haarlem, den helder) |
|  |

**1b** bij 1a alleen gebruik gemaakt van minimum bij new\_traject en new\_connection

**AANPAK:**

Loops voor max treinen + max minuten, geeft de beste combinatie terug

1 traject per keer:

* Opstellen: kortste ongebruikte kritieke pad
* Zo ver mogelijk uitbreiden met kortste ongebruikte kritieke paden

Alle trajecten gemaakt:

* Alle ongebruikte, kritieke paden proberen te koppelen aan 1 van de trajecten mbv dijkstra

**New\_traject:** maximum

**New\_connection**: minimum 4 114 9888,2 0

**New\_traject**: maximum

**New\_connection**: maximum 4 98 9886,4 0

**New\_traject**: minimum

**New\_connection:** maximum 4 108 9882,3 0

**LOOP random random**

100x per max\_trein, max\_min

for treinen in range(1, 6):

for minuutjes in range(50, 121):

**4 120: 9889,0**

**1c** alles kritiek maken

**AANPAK:**

Loops voor max treinen + max minuten, geeft de beste combinatie terug

1 traject per keer:

* Opstellen: kortste ongebruikte kritieke pad
* Zo ver mogelijk uitbreiden met kortste ongebruikte kritieke paden

Alle trajecten gemaakt:

* Alle ongebruikte, kritieke paden proberen te koppelen aan 1 van de trajecten mbv dijkstra

class CONNECTION():

Init:

self.critic = False 🡪 self.critic = True

setCritic:

self.critic = critic 🡪 self.critic = True

class STATION():

init:

self.critic = False 🡪 self.critic = True

**Met new\_traject() op minimum en new\_connection op minimum**

loop for treinen: van 1 t/m 6 (7 duurt extreem lang)

loop for minuten: van 1 t/m 120

beste resultaat: 4 118 9878,5 0

**Met new\_traject() op maximum en new\_connection op minimum**

loop for treinen: van 1 t/m 6

loop for minuten: van 1 t/m 120

beste resultaat: 4 118 9875,5 0

**Met new\_traject() op maximum en new\_connection op maximum**

loop for treinen: van 1 t/m 6

loop for minuten: van 1 t/m 120

beste resultaat: 4 118 9875,9 0

**Met new\_traject() op minimum en new\_connection op maximum**

loop for treinen: van 1 t/m 6

loop for minuten: van 1 t/m 120

beste resultaat: 5 114 9858,7 0

(1-6, 1-120 100x)

**RANDOM, MIN:** 4120 9879,5 0

**RANDOM, MAX:** 4 116 9879,0 0

**MIN, RANDOM:** 4 112 9878,0 0

**MAX, RANDOM:** 5 118 9858,1 0

**RANDOM, RANDOM:** 4 105 9880,1 0

Later: nog eens random random: 4 100 9880.6 0

**4**

**100**

**9881.0**

**0**

**1D**

13, 180, random, random, 500x loopen: 9629,6

12, 180, random, random, 500x: 9644,2

11, 180, random, random, 500x: 9660,5

10, 180, random, random, 500x: 9676,3

9, 180, random, random, 500x: 9684,1 (langste traject : 176)

Bij traject uitbreiden per spoor & 5000x: 9693,5

Bij trajecten uitbreiden heel traject & 5000x: 9699,5

8, 180, random, random, 500x: 9208,1 (3 over)

9, 175, random, random, 500x: 9683,9

**BIJ LINKEN EERST KIJKEN OF DOELWAARDE TOENEEMT ->**

9, 180, random, random: 9700,9

Linken wordt nu chronologisch gedaan, maakt het verschil als we dit anders doen?

**1E**

20, 180, random, random, 500x: 9440,6

19, 180, random, random, 500x: 9461,3

18, 180, random, random, 500x: 9482,9

17, 180, random, random, 500x: 9497,3

16, 180, random, random, 500x: 9514,6

12, 180, random, random, 500x: 9569,9

Bij traject uitbreiden per spoor & 5000x: 9588,0

Bij per traject helemaal uitbreiden & 5000x: 9593,9

**GRENZEN:**

**1a & 1b**

Lowerbound: 0 \* 10.000 – 7 \* 20 – 120\*7 /10 = **-224**

Upperbound: kritieke sporen zijn samen 287 minuten. 287/120 -> minimaal 3 treinen nodig.

1 \* 10.000 – 3 \* 20 – 287/10 = **9911,3**

**1c**

Lowerbound: 0 \* 10.000 – 7 \* 20 – 120\*7 /10 = **-224**

Upperbound: kritieke sporen zijn samen 381 minuten. 381/120 -> minimaal 4 treinen nodig.

1 \* 10.000 – 4 \* 20 – 381/10 = **9881,9**

**1d**

Lowerbound = 0 \* 10.000 – 20 \* 20 – 180\*20/10 = -**760**

Upperbound = 1 \* 10.000 – 6 \* 20 - 1011/10 = **9778,9**

**1e**

Lowerbound = 1/89 \* 10.000 – 20 \* 20 – 180\*20/10 = -**647,6**

Upperbound = 1 \* 10.000 – 9 \* 20 - 1551/10 = **9664,9**

**STATESPACE**

**Bij 1a & 1b & 1c:**

Aantal trajecten binnen 120 min (zo lang mogelijk): 2737

2737 boven 7: 2,265 \* 10^20

2737 boven 4: 2,333 \* 10^12

**Bij 1d & 1e:**

Aantal trajecten binnen 180 min (zo lang mogelijk): 3761642

3761642 boven 20: 1,253 \* 10^53

3761642 boven 12: 1,643 \* 10^34

3761642 boven 9: 4,109 \* 10^26

Bij 1a/b/c is het aantal trajecten van zo lang mogelijke lengte en maximaal 120 minuten dat gemaakt kan worden 2737. Omdat er 7 trajecten gebruikt mogen worden is de statespace 7 boven 2737 = 2,265 \* 10^20.

Bij 1d/e is het aantal trajecten van zo lang mogelijke lengte en maximaal 180 minuten dat gemaakt kan worden, is 3761642. Omdat er 20 trajecten gebruikt mogen worden is de statespace 20 boven 3761642 = 1,253\* 10^53.

<https://www.calculatorsoup.com/calculators/discretemathematics/combinations.php>