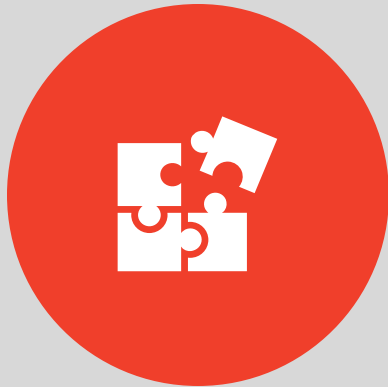




ONLY RUSH!!!

Door Pepijn, Cécile en Sjors

Casus introductie: Rush Hour



PUZZELSPEL
KRIJG DE RODE AUTO UIT DE FILE,
ONDANKS DE BLOKKERENDE AUTO'S



MET MEERDERE AUTO'S DIE JE KUNT
BEWEGEN, WEET JE NIET WELKE DE
JUISTE ROUTE IS



IN HOEVEEL ZETTEN KRIJG JE DE RODE
AUTO UIT DE FILE?

Bereken een set moves zodat de rode auto het bord kan verlaten



Regels van het spel:

- Iedere uitdaging heeft een eigen InitialBoard (links game #1)
- Schuif de voertuigen heen en weer tot de weg vrij is
 - Ze kunnen niet van richting veranderen
 - De lengte varieert: auto's 2 en vrachtwagens 3 blokjes
 - door te schuiven ontstaan nieuwe gamestates
- Win-condition:
Alle grids rechts v d rode auto zijn leeg!

Objective Function

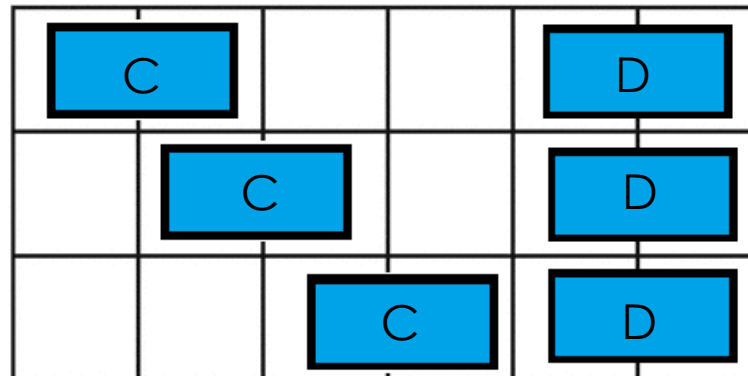
- $Z = x$
- x is het aantal zetten benodigd om een oplossing te vinden.



State Space:

$$2 \cdot 2 \cdot 3 \cdot 5 \cdot 3 \cdot 3 \cdot 3 \cdot 5 \cdot 5 \cdot 3 \cdot 4 \cdot 3 = 1.458.000$$

Posities auto C:



State space

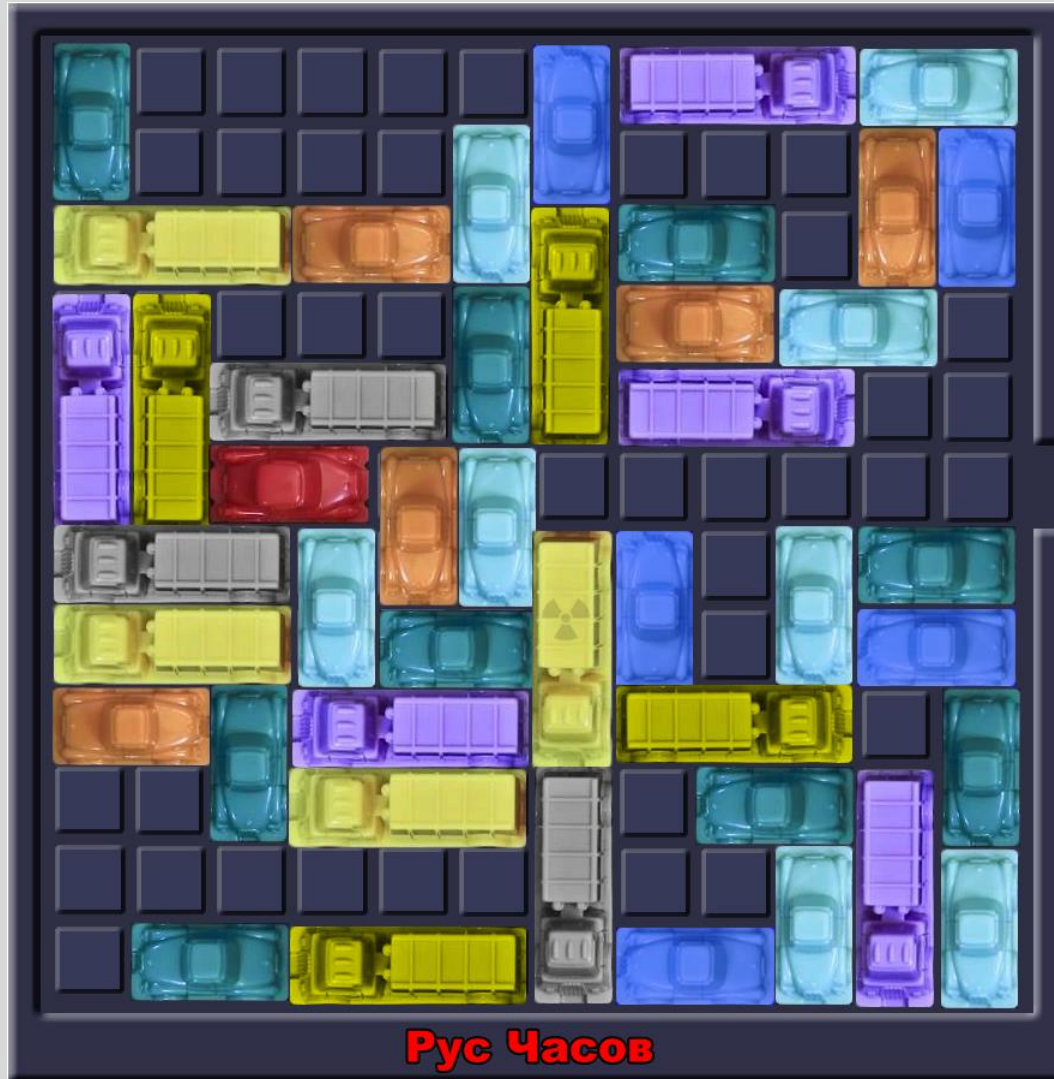
-Aantal mogelijke posities hangt af van het aantal auto's in dezelfde rij/kolom met dezelfde oriëntatie.

-Auto C: 3 posities

-Auto D: 3 posities

-Hun state space is $3 \cdot 3 = 9$

-Pas dit toe op alle rijen en kolommen en vermenigvuldig de state space.



State space:

$$8*7*2*8*8*8* \dots = 1,08 * 10^{36}$$

Column (j)

Row (i)

1-1	1-2	1-3	1-4	1-5	1-6
2-1	2-2	2-3	2-4	2-5	2-6
3-1	3-2	3-3	3-4	3-5	3-6
4-1	4-2	4-3	4-4	4-5	4-6
5-1	5-2	5-3	4-5	5-5	5-6
6-1	6-2	6-3	4-6	6-5	6-6

Game#1

- A,H,2,6,2
- B,H,4,6,3
- C,H,2,5,2
- D,H,5,5,2
- E,V,4,4,2
- F,H,1,3,2
- G,V,3,3,2
- H,H,4,3,2
- I,V,6,3,2
- J,H,5,2,2
- K,V,1,1,2
- L,V,3,1,2
- **X,H,1,4,2**

Rushhour12x12_7

A,H,8,12,3
 B,H,11,12,2
 C,V,1,11,2
 D,V,7,11,2
 E,H,1,10,3
 F,H,4,10,2
 G,V,6,10,2
 H,H,8,10,2
 I,V,11,10,2
 J,V,12,10,2
 K,H,8,9,2
 L,H,10,9,2
 M,H,3,8,3
 N,V,6,8,2
 O,V,7,8,3
 P,H,8,8,3
 Q,V,1,7,3
 R,V,2,7,3
 S,H,1,6,3
 T,V,5,6,2
 U,V,6,6,2
 V,H,1,5,3
 W,V,4,5,2
 Y,H,5,5,2
 AA,V,8,5,2
 AB,V,10,5,2
 AC,H,11,5,2
 AD,H,1,4,2
 AE,H,4,4,3
 AF,V,7,4,3
 AG,H,8,4,3
 AH,V,3,3,2
 AI,H,4,3,3
 AJ,H,9,3,2
 AK,V,12,3,2
 AL,H,2,1,2
 AM,H,4,1,3
 AN,V,7,1,3
 AP,H,8,1,2
 AO,V,10,1,2
 AP,V,11,1,3
 AQ,V,12,1,2
 X,H,3,7,2

Column

1-6	2-6	3-6	4-6	5-6	6-6
1-5	2-5	3-5	4-5	5-5	6-5
1-4	2-4	3-4	4-4	4-5	4-6
1-3	2-3	3-3	4-3	5-3	6-3
1-2	2-2	3-2	4-2	5-2	6-2
1-1	2-1	3-1	4-1	5-1	6-1

Row

Rushhour6x6_1.csv

Tweeledige info:

1. Filename:

- **Maat** vh bord 6x6
- **Nummer** vh spel

2. File inhoud:

- Voertuig **ID**: 1 of 2 letters
- **Horizontale** of **Verticale** oriëntatie
- **Row** locatie vd linker gridbox
- **Column** locatie vd onderste gridbox
- **Maat** van het voertuig

Oriëntatie + Maat:

→ positie voertuig

→ 90 graden draai:

board_size +1 – column
 wordt row(i) && Row wordt
 column(j)

#1



12 auto's + 1 truck

#2



12 auto's + 1 truck

#3

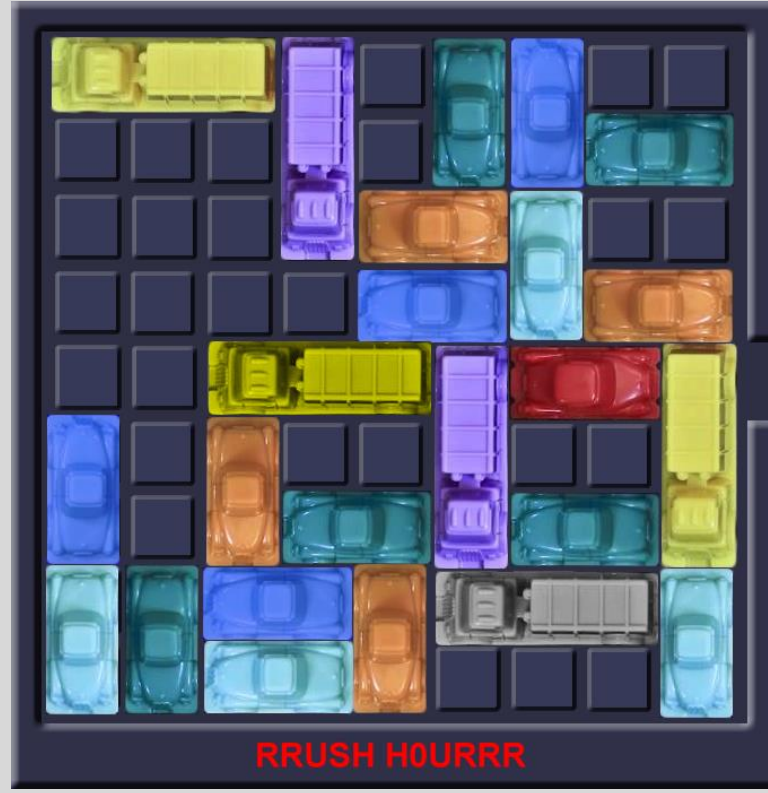


6 auto's + 3 trucks

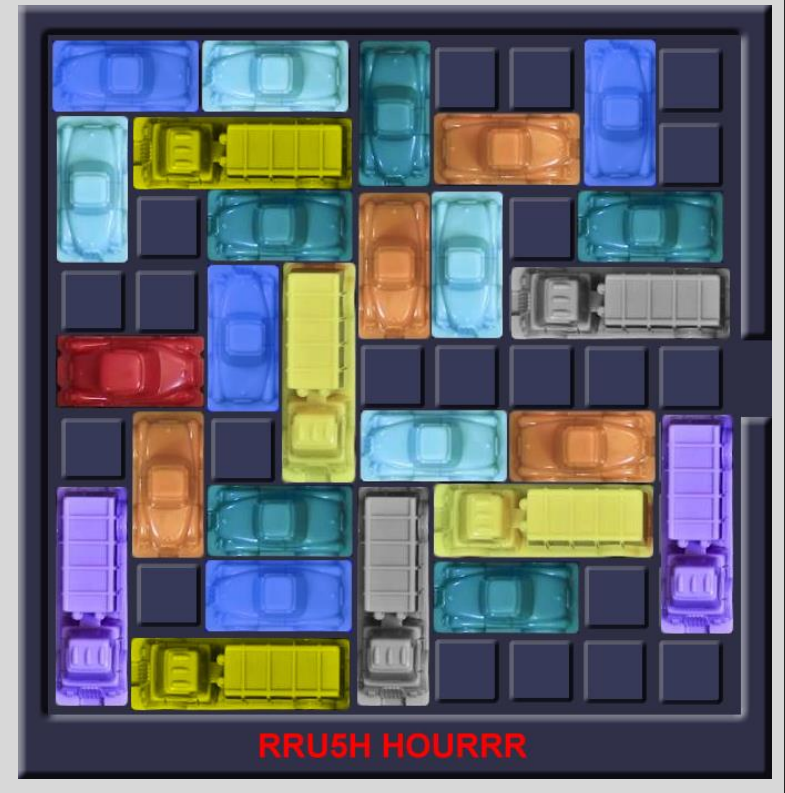
#4



#5



#6



Random Algoritme: resultaten

	#1	#2	#3	#4	#5	#6
Runs	17776	12015	3364	1062	1302	1051
Avg	3521	1258	7005	2740	3201	2556
Least	89	61	1069	546	314	1067
Solved %	2.5%	56%	4.9%			

Hard constraints:

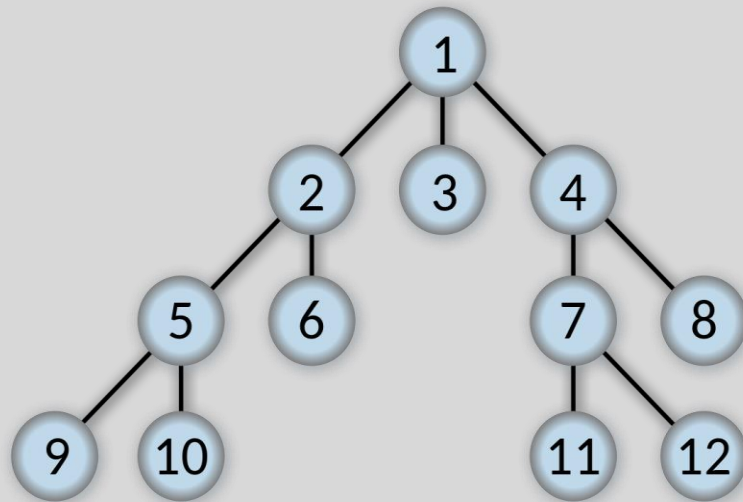
- Geen collisions met andere auto's
- Geen collisions met de grenzen

← Heuristiek:

- Unieke states

Breadth First Search

Standaard BFS



Node bestaat uit:

- 1) Staat van het spel.
- 2) Zetten die gedaan zijn om op die staat uit te komen.

Heuristieken:

- Als de rode auto naar de exit kan worden bewogen, doe dat dan.
- Ga alleen unieke states af.

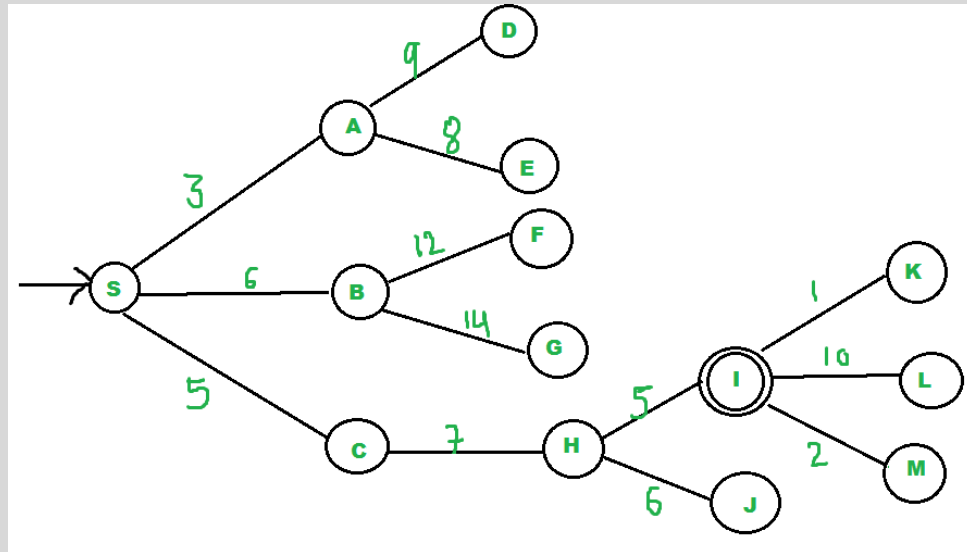
Breadth First Resultaten

Game #	1	2	3	4	5	6	7
Grootte	6x6	6x6	6x6	9x9	9x9	9x9	12x12
Zetten	21	15	33	27	N/A	N/A	N/A
Tijd (s)	3	16	132	54000	>54000	N/A	N/A

NB: Zonder het gebruik van heuristieken ontstaat er een memory error.

Best first

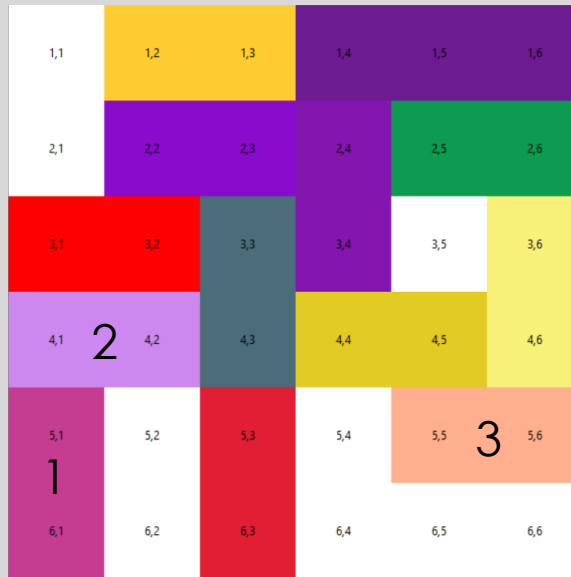
Standaard Best First



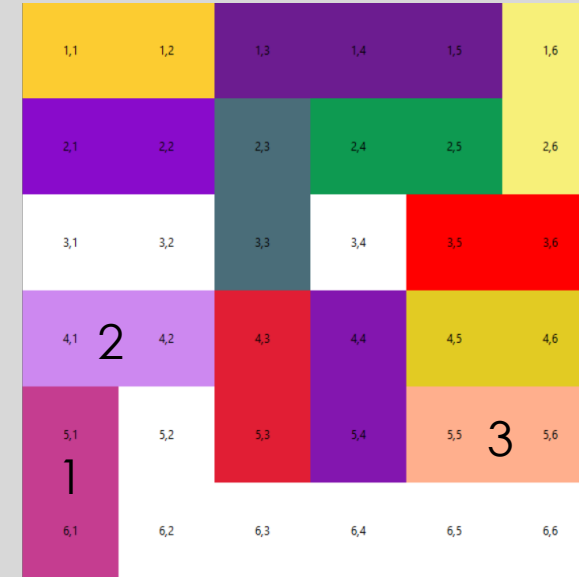
Heuristiek NR3: Ga de beste states eerst af!

Best First Heuristieken: beste state

Game 1 begin state



Game 1 final state



heuristiek = aantal voertuigen – aantal voertuigen op dezelfde positie als in de final state
= 13 - 3
= 10

Best First Resultaten

Game	1	2	3	4	5	6	7
Grootte	6x6	6x6	6x6	9x9	9x9	9x9	12x12
Iteraties	1000	1000	1000	1000	N/A	N/A	N/A
Minste zetten	34	23	64	32	N/A	N/A	N/A
Tijd (s)	6	44	222	101	N/A	N/A	N/A

NB: Zonder een heuristiek die checkt voor unieke states duurt het te lang.

Depth First Search

class Depthfirst: Resultaten

#1

- Moves: 482
- vs 21 BFS

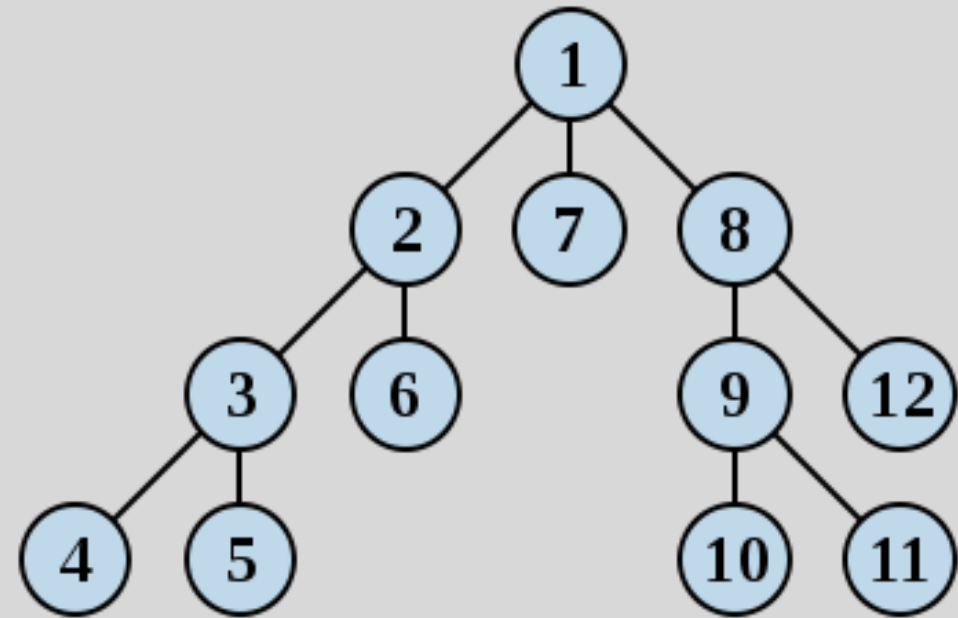
#2

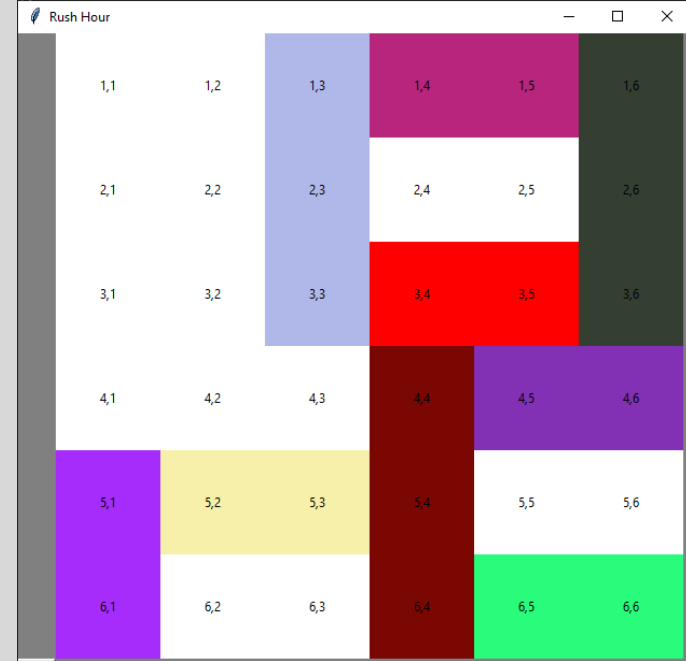
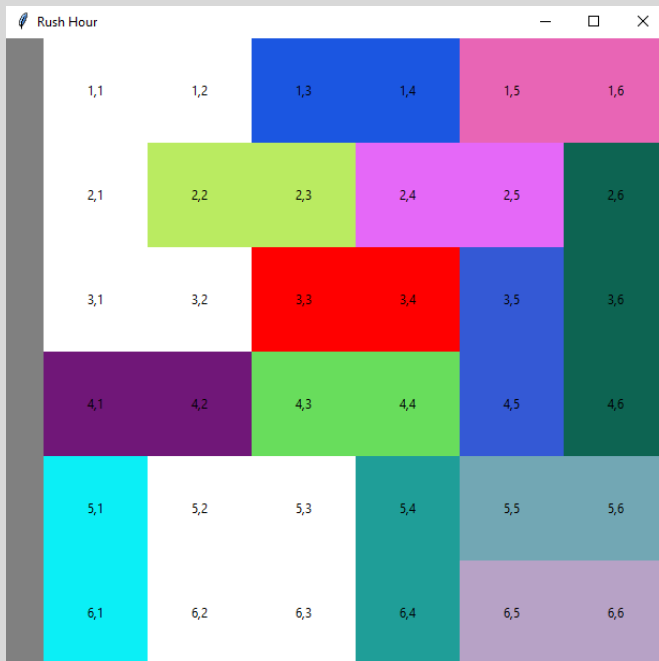
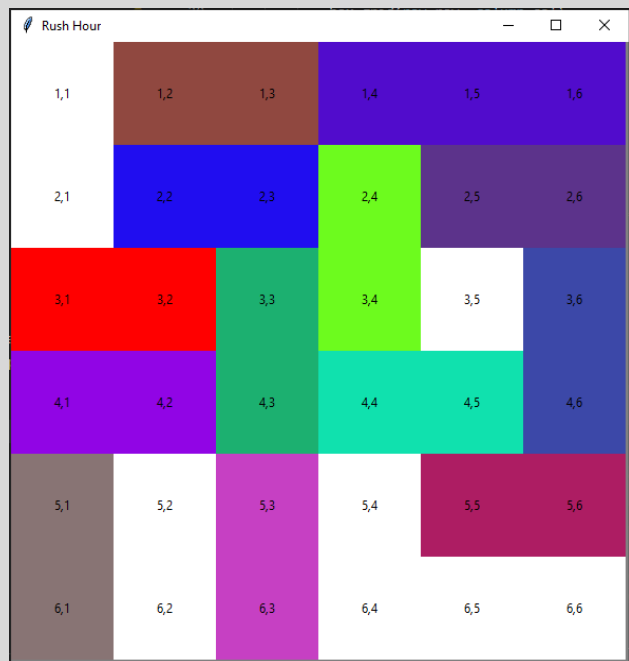
- Moves: 1464
- vs 15 BFS

#3

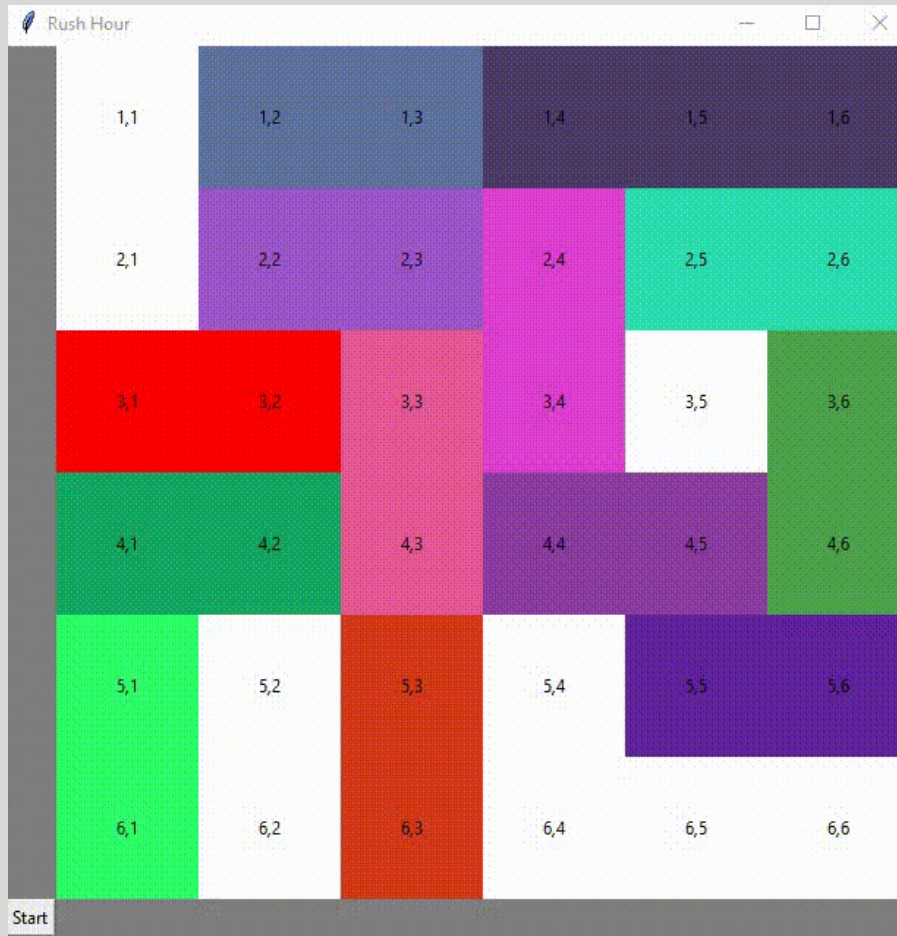
- Moves: 3079
- vs 35 BFS

Volgorde: door tot het doodloopt en dan backtracken om verder te zoeken.





Visualisatie



Met blik op de toekomst

**** van toepassing op welke code ****

- Random + heuristiek: unieke states

**** met welk doeleinde/of visie ****

- verband tussen het aantal zetten en het aantal mogelijkheden vast te komen te zitten.

Conclusie

De heuristiek “is state uniek” heeft gewenste en ongewenste gevolgen.



RANDOM:
KAN TOT EEN SNELLERE
OPLOSSING LEIDEN, MAAR
KOMT HEEL VAAK VAST TE
ZITTEN.



BREADTHFIRST
HEEFT HET ALLEEN GEWENSTE
GEVOLGEN. HET MAAKT
BREADTHFIRST ERG SNEL
ZONDER BELANGRIJKE
OPLOSSINGEN WEG TE
GOOIEN.



BESTFIRST
MAAKT HET SNEL MAAR GOOIT
BELANGRIJKE OPLOSSINGEN
WEG.



DEPTHFIRST
MAAKT HET SNEL MAAR GOOIT
BELANGRIJKE OPLOSSINGEN
WEG.

Breadthfirst is met alle gewenste gevolgen óók de **langzaamste oplossing** van Rush Hour.

Einde van de presentatie

Koffiepauze \(\^^)/



Of zijn er nog vragen?