



Informatik

Übungsstunde

Typ: string

Matrizen

Erfordert: `#include<vector>`

Wichtige Befehle:

Definition: `std::vector<std::vector<int> >
my_vec (n_rows, std::vector<int>(n_cols,
init_value))`

Zugriff: `my_arr.at(1).at(1) = 8 * my_arr.at(0).at(2);`

Rekursion

- Base Case: Wir müssen das Problem für ein „simples“ n lösen
- Sei $(n-1)$ gegeben. Inwiefern kann ich diese Lösung benutzen um n zu erhalten

Collatz

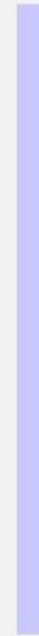
$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ 3n + 1 & \text{if } n \text{ is odd} \end{cases}$$



Links



Mitte

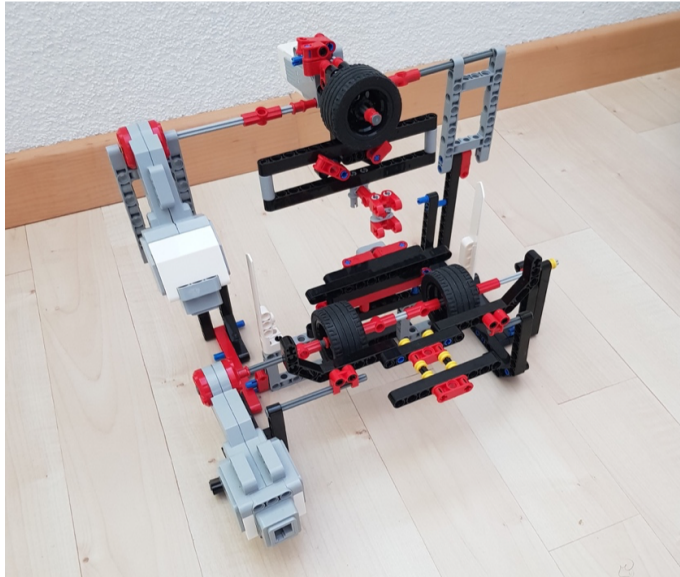


Rechts

Backtracking

- Setze validen Wert in erstes Feld. Gehe zum nächsten Feld und wiederhole solange ein valider Wert verfügbar ist.
- Kann kein valider Wert in ein Feld eingetragen werden, ist das Problem bei einem vorherigen Feld. Also geht man ein Feld zurück und versucht einen anderen validen Wert. Man wiederhole diesen Schritt so oft, bis man wieder einen neuen Wert für ein Feld gefunden hat
- Ist man beim ersten Feld wieder angekommen und hat keinen Wert zur Verfügung existiert keine Lösung.

Backtracking in Sudoku



5	3	1	2	7	6	8	9	4
6	2	4	1	9	5	2		
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

