

# **Sudoku**

## Algorithmus zum Lösen von Sudokus

B.Daiber, A.Kaißer  
Technische Hochschule Ulm  
Ulm, 19.07.2023

Programmieren

# Inhaltsverzeichnis

Grundlagen des Spiels

Konzept

Implementierung

Ergebnisse

Fazit

Ausblick

# Aufgabenstellung

Entwicklung eines Algorithmus zum Lösen von Sudokus in C++

# Geschichte

- Wurde 1979 in den USA erfunden (unter anderem Namen)
- Von Howard Garns

# Geschichte

- Wurde 1979 in den USA erfunden (unter anderem Namen)
- Von Howard Garns
- 1984 erhält es in Japan seinen Namen
- Sudoku = ‘Ziffern dürfen nur einmal vorkommen’

# Geschichte

- Wurde 1979 in den USA erfunden (unter anderem Namen)
- Von Howard Garns
- 1984 erhält es in Japan seinen Namen
- Sudoku = 'Ziffern dürfen nur einmal vorkommen'
- Seit 2005 in Deutschland regelmäßig in Zeitungen

## Aufbau

- 9 Blöcke mit je 9 Feldern
- Zahlen von 1 bis 9
- Insgesamt 81 Felder

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

Abbildung: Aufbau eines Sudokus

# Regeln

Jede Zahl darf nur einmal...

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

Abbildung: Aufbau eines Sudokus



# Regeln

Jede Zahl darf nur einmal...

- in jedem Block vorkommen

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

Abbildung: Aufbau eines Sudokus

# Regeln

Jede Zahl darf nur einmal...

- in jedem Block vorkommen
- in jeder Zeile vorkommen

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

Abbildung: Aufbau eines Sudokus

# Regeln

Jede Zahl darf nur einmal...

- in jedem Block vorkommen
- in jeder Zeile vorkommen
- in jeder Spalte vorkommen

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

Abbildung: Aufbau eines Sudokus

# Regeln

Jede Zahl darf nur einmal...

- in jedem Block vorkommen
- in jeder Zeile vorkommen
- in jeder Spalte vorkommen

Ziel des Spiels:

Jedes Feld mit einer Zahl füllen

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

Abbildung: Aufbau eines Sudokus

# Konzept

- Klasse "Sudoku" für das Spielfeld
  - Lesen
  - Schreiben

# Konzept

- Klasse "Sudoku" für das Spielfeld
  - Lesen
  - Schreiben
- Analytischer Algorithmus als Basis

# Konzept

- Klasse "Sudoku" für das Spielfeld
  - Lesen
  - Schreiben
- Analytischer Algorithmus als Basis
- Funktion zur Validierung

# Konzept

- Klasse "Sudoku" für das Spielfeld
  - Lesen
  - Schreiben
- Analytischer Algorithmus als Basis
- Funktion zur Validierung
- Funktionsfähigkeit überprüfen mit Tests



# Konzept

- Klasse "Sudoku" für das Spielfeld
  - Lesen
  - Schreiben
- Analytischer Algorithmus als Basis
- Funktion zur Validierung
- Funktionsfähigkeit überprüfen mit Tests
- Backtracking Algorithmus für schwere Probleme

## Sudoku Klasse

- Spielfeld erstellen (`int field[9][9]`)

## Sudoku Klasse

- Spielfeld erstellen (`int field[9][9]`)
- Ausgabe auf der Konsole

## Sudoku Klasse

- Spielfeld erstellen (int field[9][9])
- Ausgabe auf der Konsole
- Einlesen über den Konstruktor
  - Aus 2D-Array
  - Aus CSV-Datei

## Sudoku Klasse

- Spielfeld erstellen (int field[9][9])
- Ausgabe auf der Konsole
- Einlesen über den Konstruktor
  - Aus 2D-Array
  - Aus CSV-Datei
- Analytischer Algorithmus

## Sudoku Klasse

- Spielfeld erstellen (`int field[9][9]`)
- Ausgabe auf der Konsole
- Einlesen über den Konstruktor
  - Aus 2D-Array
  - Aus CSV-Datei
- Analytischer Algorithmus
- Validierung des Spielfeldes

## Analytischer Algorithmus

Überprüft das Feld auf eindeutige (triviale) Lösungen

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

Abbildung: Analytischer Ansatz

## Analytischer Algorithmus

Überprüft das Feld auf eindeutige (triviale) Lösungen

- Betrachtung eines Feldes

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

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

Abbildung: Analytischer Ansatz



## Analytischer Algorithmus

Überprüft das Feld auf eindeutige (triviale) Lösungen

- Betrachtung eines Feldes
  - Block

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

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

Abbildung: Analytischer Ansatz

## Analytischer Algorithmus

Überprüft das Feld auf eindeutige (triviale) Lösungen

- Betrachtung eines Feldes
  - Block
  - Zeile

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

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

Abbildung: Analytischer Ansatz

## Analytischer Algorithmus

Überprüft das Feld auf eindeutige (triviale) Lösungen

- Betrachtung eines Feldes
  - Block
  - Zeile
  - Spalte

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

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

Abbildung: Analytischer Ansatz

## Analytischer Algorithmus

Überprüft das Feld auf eindeutige (triviale) Lösungen

- Betrachtung eines Feldes
  - Block
  - Zeile
  - Spalte
- Übriges Element eintragen

1

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

Abbildung: Analytischer Ansatz

## Analytischer Algorithmus

Überprüft das Feld auf eindeutige (triviale) Lösungen

- Betrachtung eines Feldes
  - Block
  - Zeile
  - Spalte
- Übriges Element eintragen
- Für alle leeren Felder

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

Abbildung: Analytischer Ansatz

## Analytischer Algorithmus

Überprüft das Feld auf eindeutige (triviale) Lösungen

- Betrachtung eines Feldes
  - Block
  - Zeile
  - Spalte
- Übriges Element eintragen
- Für alle leeren Felder
- So lange Einträge statt finden

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

Abbildung: Analytischer Ansatz

## Analytischer Algorithmus

Überprüft das Feld auf eindeutige (triviale) Lösungen

- Betrachtung eines Feldes
  - Block
  - Zeile
  - Spalte
- Übriges Element eintragen
- Für alle leeren Felder
- So lange Einträge statt finden
- Nicht lösbar bei übrigen Nullen

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

Abbildung: Analytischer Ansatz

## Backtracking Algorithmus - Grundlagen

- Suchstrategie nach dem Trial and Error Prinzip
- Implementierung mit Rekursion

Lösungsschritte:



## Backtracking Algorithmus - Grundlagen

- Suchstrategie nach dem Trial and Error Prinzip
- Implementierung mit Rekursion

Lösungsschritte:

- Teillösung wird ausgewählt
- Überprüfen, ob diese stimmt
  - Falls ja, wird mit dem nächsten Teilschritt weitergemacht
  - Falls nein, geht man einen Schritt zurück, neue mögliche Lösung
- Alle Teillösungen ermittelt - return true
- Eine Teillösung existiert nicht - return false

## Backtracking Algorithmus - Umsetzung

- Auslagerung in "SudokuBacktrack" Klasse
- Hilfsfunktion "empty" überprüft, ob noch Felder leer sind
- Hilfsfunktion "correctPart" überprüft Korrektheit in Zeile, Spalte und Block
- Funktion "solveBacktracking" setzt Algorithmus um

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

Abbildung: Aufbau eines Sudokus

## Backtracking Algorithmus - Umsetzung

Funktion "solveBacktracking"

- Abfrage "empty"
- Auswahl eines leeren Felds
- Auswahl einer Zahl (9,8,7,6,5,4,3,2,1,0) und Überprüfen mit "correctPart"

## Backtracking Algorithmus - Umsetzung

Funktion "solveBacktracking"

- Abfrage "empty"
- Auswahl eines leeren Felds
- Auswahl einer Zahl (9,8,7,6,5,4,3,2,1,0) und Überprüfen mit "correctPart"
- Einsetzen der Zahl, wenn korrekt
- Ein Schritt zurück, wenn falsch

## Backtracking Algorithmus - Umsetzung

Funktion "solveBacktracking"

- Abfrage "empty"
- Auswahl eines leeren Felds
- Auswahl einer Zahl (9,8,7,6,5,4,3,2,1,0) und Überprüfen mit "correctPart"
- Einsetzen der Zahl, wenn korrekt
- Ein Schritt zurück, wenn falsch
- Rekursiver Aufruf der Funktion

## Einlesen der Sudokus

- Sudokus einzeln in CSV Dateien abgespeichert
  - SudokuLeicht
  - SudokuMittel
  - SudokuSchwer
  - SudokuLeer
  - SudokuFehler
- Einlesen beim Erzeugen eines Objekts der Klasse Sudoku

## Validierung der Ergebnisse

Überprüfung der Ergebnisse durch eine Methode

## Validierung der Ergebnisse

Überprüfung der Ergebnisse durch eine Methode

- Nullen im Feld
- Werte außerhalb  $[0,9]$



## Validierung der Ergebnisse

### Überprüfung der Ergebnisse durch eine Methode

- Nullen im Feld
- Werte außerhalb  $[0,9]$
- Prüfen nach den Spielregeln
  - Alle Zahlen von 1 bis 9 in allen ...
    - 9 Zeilen
    - 9 Spalten
    - 9 Blöcke
  - Prüfung durch Summe der Quadrate aller Zahlen
  - 27 Teillösungen

# Ergebnisse

- Oben: lösbares Sudoku

```

Test: sudokuMittel

0 4 0   2 9 8   5 0 7
0 5 2   1 0 6   9 8 4
0 0 0   0 4 5   0 6 2

9 0 3   6 1 4   8 7 0
0 8 6   5 3 0   4 2 9
5 7 4   0 8 2   6 0 3

8 3 0   7 6 9   2 4 1
4 1 9   8 0 3   7 5 6
2 0 7   4 5 1   3 0 8

Sudoku solved correctly:
27 correct partial sol.

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

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

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

```

**Abbildung:** Ausgabe gelöstes Sudoku

# Ergebnisse

- Oben: lösbares Sudoku
- Validierung: alle 27  
Teillösungen sind korrekt
- Unten: gelöstes Sudoku

```

Test: sudokuMittel

0 4 0   2 9 8   5 0 7
0 5 2   1 0 6   9 8 4
0 0 0   0 4 5   0 6 2

9 0 3   6 1 4   8 7 0
0 8 6   5 3 0   4 2 9
5 7 4   0 8 2   6 0 3

8 3 0   7 6 9   2 4 1
4 1 9   8 0 3   7 5 6
2 0 7   4 5 1   3 0 8

Sudoku solved correctly:
27 correct partial sol.

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

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

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

```

**Abbildung:** Ausgabe gelöstes Sudoku

# Ergebnisse

- Fehlerhaftes Sudoku

Test: sudokuFehler

0 4 0	2 9 8	5 0 7
0 5 2	1 0 6	9 8 4
0 0 0	0 4 5	0 6 2

9 0 3	6 1 4	8 7 0
0 8 6	5 3 0	4 2 9
5 7 4	0 8 2	6 0 3

8 3 0	7 6 9	2 4 1
4 1 9	8 0 3	7 5 6
2 0 7	4 5 1	3 0 3

Sudoku is not solved:  
24 correct partial sol.

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

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

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

**Abbildung:** Ausgabe fehlerhaftes Sudoku

# Ergebnisse

- Fehlerhaftes Sudoku
- Validierung: nur 24 Teillösungen sind korrekt
- Sudoku ist nicht lösbar

Test: sudokuFehler

0 4 0	2 9 8	5 0 7
0 5 2	1 0 6	9 8 4
0 0 0	0 4 5	0 6 2

9 0 3	6 1 4	8 7 0
0 8 6	5 3 0	4 2 9
5 7 4	0 8 2	6 0 3

8 3 0	7 6 9	2 4 1
4 1 9	8 0 3	7 5 6
2 0 7	4 5 1	3 0 3

Sudoku is not solved:  
24 correct partial sol.

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

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

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

**Abbildung:** Ausgabe fehlerhaftes Sudoku

## Fazit

- Analytischer Algorithmus kann einfache Sudokus lösen
- Backtracking Algorithmus kann alle Sudokus lösen
- Löst in weniger als 0,1 Sekunde

## Ausblick

### Möglichkeiten der Weiterentwicklung

- Effizienz des Algorithmus steigern
- Eingabe der Sudokus in die Konsole
- Schreiben der Sudokus in eine CSV-Datei
- Eigene Sudokus generieren

Vielen Dank  
für Ihre Aufmerksamkeit