

Magisches Quadrat

2	7	6	→	15
9	5	1	→	15
4	3	8	→	15
↙	↓	↓	↓	↘
15	15	15	15	15

Das magische Quadrat ist ein mathematisches Rätsel. Gegeben ist ein Quadrat der Seitenlänge n . Die Aufgabe besteht darin, die Zahlen 1 bis n^2 so zu verteilen, dass die Summe der Zahlen jeder Zeile, jeder Spalte und der beiden großen Diagonalen identisch ist. Diese Summe wird auch als magische Zahl bezeichnet.

Da die Verteilung der Zahlen bei gegebener Seitenlänge immer feststeht, ist auch immer die magische Zahl berechenbar. Es gilt, dass die magische Zahl $1/n$ mal der Summe der Zahlen von 1 bis n^2 sein muss:

$$\text{magischeZahl} = \frac{1}{n} \sum_{k=1}^{n^2} k = \frac{n^3 + n}{2}$$

Algorithmus



Lies hier erst weiter, nachdem du dir selbst Gedanken gemacht hast, wie du vorgehen könntest.

Die Vorgehensweise ist letztlich sehr ähnlich zum 8-Damen-Problem. Auch hier versuchen wir systematisch das nächste Feld zu besetzen. Unterschiede sind im Detail:

- Beim 8-Damen-Problem bestand der rekursive Aufruf im Besetzen der nächsten Reihe, da immer nur eine Dame pro Reihe platziert werden darf. Dafür wurde pro rekursivem Aufruf jede Spalte durchprobiert.
- Beim magischen Quadrat besteht der rekursive Aufruf im Besetzen des nächsten Felds, da jedes Feld eine Zahl enthalten muss. Pro rekursivem Aufruf wird dafür jede noch nicht vergebene Zahl durchprobiert.

Aus dieser Überlegung ergibt sich der folgende Pseudocode:

```
fuelleFeld(f):
  wenn f < n*n:
    für jede Zahl i = 1 bis n*n:
```

```
wenn i noch nicht genutzt wurde:  
    markiere i als genutzt  
    platziere i im quadrat im feld f  
  
wenn das Quadrat fehlerfrei ist:  
    fuelleFeld(f+1)  
  
entferne i aus dem quadrat aus feld f  
markiere i als ungenutzt  
sonst:  
    druckeQuadrat()
```

Gestartet wird der Code mit dem Aufruf `fuelleFeld(0)`

Hinweise:

- Der rekursive Parameter `f` gibt an, das wievielte Feld des Quadrats gefüllt werden soll. Dabei wird links oben begonnen zu zählen (mit 0) und reihenweise vorgegangen. Bei einem 3x3 Quadrat sieht die Nummerierung also wie folgt aus:

0	1	2
3	4	5
6	7	8

- Markierung von genutzten Zahlen: Damit keine Zahl doppelt genutzt wird, muss man sich merken, welche bereits im Quadrat platziert sind. Dafür bietet sich eine Instanzvariable vom Typ boolean-Array der Länge $n*n$ an. Die Zahl 1 wird dem Index 0 zugeordnet, die Zahl 2 dem Index 1, usw. Nennt man das Array `zahlGenutzt`, so bedeutet `true`, dass die Zahl bereits platziert ist.
- Wenn du die Zahl `i` im `f`-ten Feld des 2-dimensionalen Quadrats platzieren willst, dann musst du die fortlaufend Zahl in Koordinaten umrechnen. Dies gelingt dir mit diesem Code:

```
quadrat[f / n][f % n] = i
```



(A1)

1. Lade die

Vorlage

herunter und mache dich mit dem bereits vorgegebenen Code vertraut.

2. Vervollständige den Konstruktor.
3. Implementiere den Pseudocode im gegebenen Methodengerüst.
 1. Die Methode `quadratFehlerfrei()` ist bereits vorhanden und darf genutzt werden. Sie überprüft, ob es in einer Reihe, Spalte oder Diagonalen zu einem Fehler kommt. Bei einer enthaltenen 0 wird diese Reihe/Spalte/Diagonale nicht als fehlerhaft gewertet.
 2. Die Methode `druckeQuadrat()` ist bereits implementiert und darf genutzt werden.

From:
<https://info-bw.de/> -

Permanent link:
<https://info-bw.de/faecher:informatik:oberstufe:algorithmen:rekursion:backtracking:magisches-quadrat:start>

Last update: **29.06.2024 09:14**

