

Acht-Damen-Problem

Problemstellung

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 8 | | | | | | | | |
| 7 | | | | | | | | |
| 6 | | | | | | | | |
| 5 | | | | | | | | |
| 4 | | | | | | | | |
| 3 | | | | | | D | | |
| 2 | | | | | | | | |
| 1 | | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Wir betrachten ein Spielfeld für Schach. Wir nummerieren aber zur Vereinfachung nur mit Zahlen. Im dargestellten Spielfeld befindet sich auf der Position (6|3) eine Dame, d.h. in der Spalte 6 und in der Zeile 3.

Auf diesem Spielfeld sollen acht Damen so aufgestellt werden, dass sie sich nach den Regeln des Schachspiels gegenseitig nicht bedrohen.

Idee zum Algorithmus

Man setzt die einzelnen Damen beginnend in Spalte 1 und dann in den Zeilen bei 1 beginnend.

1. Man beginnt mit der 1. Dame auf (1|1).
2. Die 2. Dame kommt auf (2|3).
3. Die 3. Dame kommt auf (3|5).
4. Die 4. Dame kommt auf (4|2).
5. Die 5. Dame kommt auf (5|4).
6. Die 6. Dame hat in der 6. Spalte keinen unbedrohten Platz. Also nimmt man sie weg vom Feld und geht zurück zur 5. Dame.
7. Die 5. Dame kommt auf (5|8).
8. Die 6. Dame hat in der 6. Spalte keinen unbedrohten Platz. Also nimmt man sie weg vom Feld und geht zurück zur 5. Dame.
9. Die 5. Dame hat in der 5. Spalte keinen unbedrohten Platz. Also nimmt man sie weg vom Feld und geht zurück zur 4. Dame.
10. Die 4. Dame kommt auf (4|7).

11. Und so weiter, bis man eine Lösung gefunden hat.

Dem Algorithmus liegt das gleiche Prinzip wie beim „Haus des Nikolaus“ zugrunde.

Datenstrukturen

1. Man könnte wieder ein zweidimensionales boolesches Array nehmen und für jede Position festhalten, ob sie bedroht ist oder nicht. Die zweite Möglichkeit besteht darin, in drei eindimensionalen Arrays für die Zeilen, die ansteigenden Diagonalen und die fallenden Diagonalen fest zu halten, ob dort eine Dame steht oder nicht. Der zweite Ansatz wird hier verfolgt.
2. Der Index 0 wird in keinem Array benutzt, damit die Berechnungen einfacher werden. Die Variable `damen` soll die Anzahl der zu setzenden Damen angeben.
3. In `zeileFrei = new boolean[damen+1]`; wird festgehalten, ob in einer bestimmten Zeile schon eine Dame steht.
4. In `steigDiagFrei = new boolean[2*damen]`; wird festgehalten, ob in einer bestimmten steigenden Diagonale schon eine Dame steht.
Für alle Positionen einer solchen Diagonale gilt, dass `zeile-spalte` konstant ist. Betrachtet man den Term `damen+zeile-spalte`, dann bewegt sich der Index zwischen 1 und `2*damen-1`.
5. In `fallDiagFrei = new boolean[2*damen+1]`; wird festgehalten, ob in einer bestimmten fallenden Diagonale schon eine Dame steht.
Für alle Positionen einer solchen Diagonale gilt, dass `zeile+spalte` konstant ist. Der Index bewegt sich dann zwischen 2 und `2*damen`.
6. Für das Festhalten einer gefundenen Lösung ist noch ein geeignetes Array einzusetzen.

Methoden

Es bieten sich folgende Methoden an:

1. `void setzeDame(int x, int y)` setzt eine Dame in Spalte `x` und Zeile `y`.
2. `void loescheDame(int x, int y)` löscht eine Dame in Spalte `x` und Zeile `y`.
3. `boolean nichtBedroht(int x, int y)` liefert nach Untersuchung einen booleschen Wert zurück.
4. Das Hauptprogramm sollte so aussehen:

```
public static void main(String[] arg)
{ Out.println("Bestimmt werden alle Loesungen des N-Damen-Problems.");
  Out.print("Wie viele Damen sollen gesetzt werden? ");
  damen = In.readInt();
  initArrays();
  probiereDameZuSetzen(1); // 1. Dame in 1. Spalte setzen
```

```
    Out.println();
    Out.println("Es ergaben sich "+loesungsAnzahl+" Loesungen.");
} // main
```

Aufgaben

Formuliere ein komplettes Programm zur Ermittlung aller Lösungen für ein beliebiges quadratisches Spielfeld.

Zusatzaufgaben:

1. Gibt es Spielfelder ohne eine Lösung?
2. Stelle die Anzahl $q(n)$ der Lösungen in Abhängigkeit von der Damenanzahl n in einer Tabelle dar.
Finde eine Formel für $q(n)$ (bis jetzt ist keine geschlossene Formel bekannt). ;-)
3. Stelle die Damen bzw. den Ablauf des Algorithmus mit JavaKara durch Ablegen von Kleeblättern auf einem Kara-Feld dar. Dazu müssen bestimmte Methoden zum direkten Positionieren von Kara benutzt werden (siehe Dokumentation zu JavaKara).
4. **Schwierig!** Suche nur nach „echten“ Lösungen. Lösungen, die durch Drehen oder Spiegeln des Spielfeldes ineinander über geführt werden können, sind äquivalent und sollen durch einen Repräsentant der Lösung dargestellt werden (Hinweis: In den 92 Lösungen für 8 Damen sind nur echte 12 Lösungen enthalten).

Geschichte des Problems

Das Problem wurde zuerst von MAX BEZZEL in einer Schachzeitung gestellt und blieb zunächst unbeachtet. Dann wurde es von DR. NAUCK in der populären *Illustrierte Zeitung* gestellt (1.6.1850) und erregte großes Interesse. GAUSS hat das Problem in dieser Zeitung gelesen und beschäftigte sich viel damit. Am 21.9.1850 gab der **blinde** DR. NAUCK alle Lösungen an. Bis dahin hatte GAUSS erst 72 Lösungen gefunden. Er benutzte den oben dargestellten Algorithmus, hatte aber keinen Computer.