

## Wegberechnung im Gitter

Betrachten Sie das folgende Gitter mit  $m + 1$  Zeilen und  $n + 1$  Spalten ( $m \geq 1$  und  $n \geq 1$ ):<sup>1</sup> [geeksforgeeks](https://www.geeksforgeeks.org/count-possible-paths-top-left-bottom-right-nxm-matrix/)<sup>2</sup>

Angenommen, Sie befinden sich zu Beginn am Punkt  $(0, 0)$  und wollen zum Punkt  $(m, n)$ .

Für die Anzahl  $A(i, j)$  aller verschiedenen Wege vom Punkt  $(0, 0)$  zum Punkt  $(i, j)$  lassen sich folgende drei Fälle unterscheiden (es geht jeweils um die kürzesten Wege ohne Umweg!):

-  $1 \leq i \leq m$  und  $j = 0$ :

Es gibt genau einen Weg von  $(0, 0)$  nach  $(i, 0)$  für  $1 \leq i \leq m$ .

-  $i = 0$  und  $1 \leq j \leq n$ :

Es gibt genau einen Weg von  $(0, 0)$  nach  $(0, j)$  für  $1 \leq j \leq n$ .

-  $1 \leq i \leq m$  und  $1 \leq j \leq n$ :

auf dem Weg zu  $(i, j)$  muss als vorletzter Punkt entweder  $(i - 1, j)$  oder  $(i, j - 1)$  besucht worden sein.

Daraus ergibt sich folgende Rekursionsgleichung:

$$A(i, j) = \begin{cases} 1 & \text{falls } (1 \leq i \leq m \text{ und } j = 0) \text{ oder } (i = 0 \text{ und } 1 \leq j \leq n) \\ A(i - 1, j) + A(i, j - 1) & \text{falls } 1 \leq i \leq m \text{ und } 1 \leq j \leq n \end{cases}$$

Implementieren Sie die Java-Klasse `Gitter` mit der Methode

```
public int berechneAnzahlWege(),
```

die ausgehend von der Rekursionsgleichung durch dynamische Programmierung die Anzahl aller Wege vom Punkt  $(0, 0)$  zum Punkt  $(m, n)$  berechnet. Die Überprüfung, ob  $m \leq 1$  und  $n \leq 1$  gilt, können Sie der Einfachheit halber weglassen.

```
43 public int berechneAnzahlWege() {
44     int i, j;
45     for (i = 1; i <= m; i++) {
46         anzahlWege[i][0] = 1;
47     }
48     for (j = 1; j <= n; j++) {
49         anzahlWege[0][j] = 1;
50     }
51     for (i = 1; i <= m; i++) {
52         for (j = 1; j <= n; j++) {
53             anzahlWege[i][j] = anzahlWege[i - 1][j] + anzahlWege[i][j - 1];
54         }
55     }
56     return anzahlWege[m][n];
}
```

<sup>1</sup>Quelle möglicherweise von <https://www.yumpu.com/de/document/read/17936760/ubungen-zum-prasenzmodul-algorithmen-und-datenstrukturen>

<sup>2</sup><https://www.geeksforgeeks.org/count-possible-paths-top-left-bottom-right-nxm-matrix/>

57

}

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/muster/dp/Gitter.java](https://github.com/src/main/java/org/bschlangaul/aufgaben/aud/muster/dp/Gitter.java)**Additum Die komplette Java-Klasse**

```

3 import org.bschlangaul.helfer.Farbe;
4 import org.bschlangaul.helfer.Konsole;
5
6 /**
7  * <a href="https://www.studon.fau.de/file2521908_download.html">Angabe:
  ↳ AB_3
8  * Greedy_DP_Backtracking.pdf</a>
9  * <a href="https://www.studon.fau.de/file2521907_download.html">Lösung:
  ↳ AB_3
10 * Greedy_DP_Backtracking_Lsg.pdf</a>
11 *
12 * Qualifizierungsmaßnahme Informatik: Algorithmen und Datenstrukturen:
13 * Aufgabenblatt 3: Algorithmenmuster.
14 *
15 * <a href="https://www.studon.fau.de/file2521908_download.html">Angabe:
  ↳ AB_3
16 * Greedy_DP_Backtracking.pdf</a>
17 * <a href="https://www.studon.fau.de/file2521907_download.html">Lösung:
  ↳ AB_3
18 * Greedy_DP_Backtracking_Lsg.pdf</a>
19 */
20 public class Gitter {
21
22     /**
23      * m + 1: Anzahl der Zeilen
24      */
25     private int m;
26
27     /**
28      * n + 1: Anzahl der Spalten
29      */
30     private int n;
31
32     /**
33      * anzahlWege[i][j]: Anzahl der Wege vom Punkt (0,0) zum Punkt (i,j)
34      */
35     private int anzahlWege[][];
36
37     public Gitter(int m, int n) {
38         this.m = m;
39         this.n = n;
40         anzahlWege = new int[m + 1][n + 1];
41     }
42
43     public int berechneAnzahlWege() {
44         int i, j;
45         for (i = 1; i <= m; i++) {
46             anzahlWege[i][0] = 1;
47         }
48         for (j = 1; j <= n; j++) {
49             anzahlWege[0][j] = 1;
50         }

```

```

51     for (i = 1; i <= m; i++) {
52         for (j = 1; j <= n; j++) {
53             anzahlWege[i][j] = anzahlWege[i - 1][j] + anzahlWege[i][j - 1];
54         }
55     }
56     return anzahlWege[m][n];
57 }
58
59 /**
60  * Zeige die Lösung in der Konsole.
61  */
62 public void zeigeLoesung() {
63     System.out.println(
64         String.format("Anzahl der Wege von %sx%s: %s", Farbe.gelb(m),
65             ↳ Farbe.gelb(n), Farbe.grün(berechneAnzahlWege())));
66     System.out.println(Farbe.rot("Gitter:"));
67     Konsole.zeige2DIntFeld(anzahlWege);
68     System.out.println();
69 }
70
71 public static void main(String args[]) {
72     new Gitter(2, 2).zeigeLoesung();
73     new Gitter(3, 3).zeigeLoesung();
74     new Gitter(4, 4).zeigeLoesung();
75     new Gitter(5, 5).zeigeLoesung();
76 }

```

Code-Beispiel auf Github ansehen: [src/main/java/org/bschlangaul/aufgaben/aud/muster/dp/Gitter.java](https://github.com/bschlangaul/aufgaben/aud/muster/dp/Gitter.java)

## Text-Ausgabe

```

1  Anzahl der Wege von 2x2: 6
2  Gitter:
3      x 0 1 2
4      0 0 1 1
5      1 1 2 3
6      2 1 3 6
7
8  Anzahl der Wege von 3x3: 20
9  Gitter:
10     x 0 1 2 3
11     0 0 1 1 1
12     1 1 2 3 4
13     2 1 3 6 10
14     3 1 4 10 20
15
16 Anzahl der Wege von 4x4: 70
17 Gitter:
18     x 0 1 2 3 4
19     0 0 1 1 1 1
20     1 1 2 3 4 5
21     2 1 3 6 10 15
22     3 1 4 10 20 35
23     4 1 5 15 35 70
24
25 Anzahl der Wege von 5x5: 252
26 Gitter:
27     x 0 1 2 3 4 5
28     0 0 1 1 1 1 1

```

```
29      1  1  2  3  4  5  6
30      2  1  3  6 10 15 21
31      3  1  4 10 20 35 56
32      4  1  5 15 35 70 126
33      5  1  6 21 56 126 252
```

### Test-Datei

```
3  import static org.junit.Assert.*;
4
5  import org.junit.Test;
6
7  public class GitterTest {
8      @Test
9      public void zweiMailZwei() {
10         Gitter gitter = new Gitter(2, 2);
11         assertEquals(6, gitter.berechneAnzahlWege());
12     }
13
14     @Test
15     public void zehnMalZwanzig() {
16         Gitter gitter = new Gitter(10, 20);
17         assertEquals(30045015, gitter.berechneAnzahlWege());
18     }
19 }
```

Code-Beispiel auf Github ansehen: [src/test/java/org/bschlangaul/aufgaben/aud/muster/dp/GitterTest.java](https://github.com/bschlangaul/aufgaben/aud/muster/dp/GitterTest.java)