

Algorithmen und Programmierung

11. Übung

1. Aufgabe:

Schreiben Sie eine rekursive Methode zur Berechnung eines Polynoms n -ten Grades

$$p(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \dots + a_1 \cdot x + a_0$$

mittels Horner Schema:

$$p(x) = ((\dots((a_n \cdot x + a_{n-1}) \cdot x + a_{n-2}) \cdot x \dots) \cdot x + a_1) \cdot x + a_0$$

Die Koeffizienten a_i , $0 \leq i \leq n$ sollen in der Reihenfolge a_0, a_1, \dots, a_n von der Tastatur eingelesen werden. Der Grad des Polynoms und der Wert x sollen der Methode als Parameter übergeben werden. Verwenden Sie dabei kein Feld zur Speicherung der Koeffizienten!

2. Aufgabe:

Gegeben sei folgende rekursive Methode:

```
public static int f(int n, int a, int b) {
    if (n==0) {
        return a;
    }
    else {
        return f(n-1, b, a+b);
    }
}
```

Geben Sie eine induktive Definition der Werte $f(n, 0, 1)$ für $n \geq 0$ an!

3. Aufgabe:

Ein Palindrom ist eine Zeichenkette, die vorwärts und rückwärts gelesen gleich lautet (Bsp.: Anna, Rentner). Entwickeln Sie eine rekursive Methode, die prüft, ob eine eingegebene Zeichenkette ein Palindrom ist. Die einzelnen Zeichen sollen dabei nacheinander eingelesen und nicht in einem Feld gespeichert werden! Die Länge der Zeichenkette soll der Methode als Parameter übergeben werden.

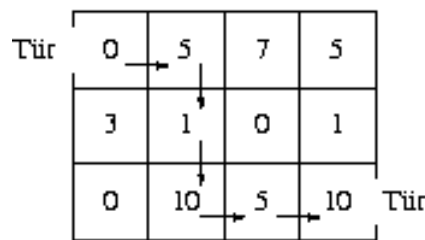
Hinweis: Groß- und Kleinschreibung können Sie ignorieren. Gehen Sie davon aus, dass entweder immer nur große oder immer nur kleine Buchstaben eingegeben werden.

4. Aufgabe:

Es war einmal ein armer Bauer, der hatte einem König einen großen Gefallen getan. Zur Belohnung sollte er einen Schatz von Goldmünzen erhalten. Da der König jedoch nicht zu viel Geld ausgeben wollte und außerdem ein Faible für knifflige Rätsel hatte, erdachte er sich ein Spiel zur Übergabe der Goldmünzen. Den größten Festsaal seines Schlosses erklärte er zu einem Spielfeld, das in gleich große quadratische Felder unterteilt wurde. Auf jedes Feld ließ er eine bestimmte Anzahl von Goldmünzen legen, die jedoch von Feld zu Feld unterschiedlich war. Es gab auch Felder völlig ohne Goldmünzen.

Der Saal mit dem Spielfeld hatte zwei Türen in gegenüberliegenden Ecken. Der Bauer wurde nun zu der einen Tür geführt (Veranschaulichung: in der Draufsicht in der linken oberen Ecke). Er sollte den Raum über das Spielfeld zur anderen Tür (rechte untere Ecke) hin durchqueren. Die Goldmünzen auf jedem Feld, das er betrat, durfte er mitnehmen. Er durfte dabei jedoch auf jedem Feld ausschließlich nach „unten“ oder nach rechts zum nächsten Feld weitergehen. Eine Wendung nach „oben“ oder ein Abzweigen nach links waren nicht erlaubt.

In folgendem Beispiel

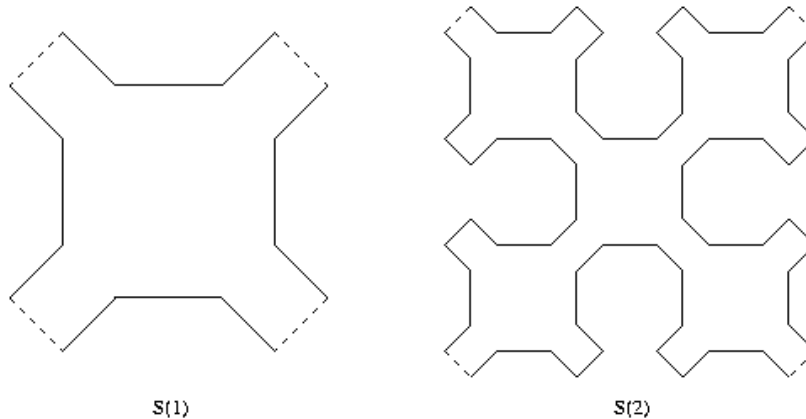


kann der Bauer maximal 31 Goldmünzen einsammeln, nämlich genau dann, wenn er dem eingezeichneten Weg folgt.

Schreiben Sie eine rekursive Methode, die für ein vorgegebenes Spielfeld berechnet, wieviele Goldmünzen der Bauer im besten Fall mitnehmen kann. Das Spielfeld sei dabei als zweidimensionale Matrix ($m \times n$) gegeben, die für jedes Feld die Anzahl der dort befindlichen Goldmünzen enthält und die der Methode als Parameter übergeben wird.


5. Aufgabe:

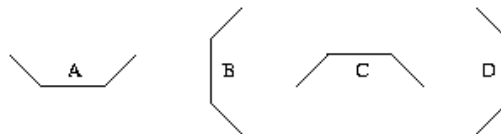
Die Sierpinski-Kurve $S(i)$ ist eine rekursiv definierte Kurve (mit i als Rekursionstiefe).



- a) Bestimmen Sie das Rekursionsmuster dieser Kurve! Orientieren Sie sich dabei an den Erläuterungen zur Hilbertkurve aus der Vorlesung.

Hilfestellung:

Die Kurve besteht aus dem Grundelement . Die vier auftretenden Orientierungen dieses Elements bezeichnen wir mit Buchstaben:



Beachten Sie, dass die Verbindungslinien zwischen den Elementen bei einer Rekursionstiefe größer 1 unterschiedliche Längen haben!

Die in der Abbildung oben gestrichelt dargestellten Linien gehören nicht zum Rekursionsmuster, sondern werden genau einmal pro Figur dargestellt. Ihre Länge ist jedoch von der Rekursionstiefe abhängig.

- b) Schreiben Sie ein Java-Applet, das diese Kurve für eine feste Rekursionstiefe darstellt. Die oben zur Veranschaulichung gestrichelt gezeichneten Linien sollen natürlich durchgezogen dargestellt werden.