Übung 8

Algorithmen in JAVA, Rekursive Funktionen (Methoden), Komplexitätsklassen

Aufgabe 1: Fakultät (1 Punkt)

Schreiben Sie eine Java-Funktion, welche **n!** für eine als Parameter übergebene Zahl **n** berechnet und zurückgibt. Verwenden Sie diesmal zur Berechnung keine Schleife, sondern Rekursion! Hinweis: Die Funktion sollte als **`static`** deklariert werden. Die Eingabezahl können Sie in **`main(...)`** von der Konsole einlesen und das Ergebnis dort auch ausgeben lassen.

Aufgabe 2: Multiplikation mal anders (1 Punkt)

In Übung 4 haben wir anhand eines Flussdiagramms gezeigt, dass die Multiplikation zweier Zahlen auch ohne Multiplikationsoperator möglich ist, in einer iterativen Variante. Schreiben Sie das gleiche Programm nun in JAVA, diesmal aber als **rekursive Funktion**!

Aufgabe 3: Komplexität (2 Punkte)

Bestimmen Sie die Komplexitätsklassen der folgenden Funktionen in O-Notation.

1. $12.5 * n^5 - 2.5 * n^3 - n^2$
2. $-2 * n^3 + 0.001 * n^4 + 100 - 10^{512} * n + n^4 * \log 2n$
3. $\frac{5*n^2*\log n+4*n}{n^2}$

Aufgabe 4: Komplexität (2 Punkte)

Ein Rechner kann pro Sekunde 10^6 Operationen durchführen. Berechnen Sie wie lange es dauern würde die folgenden Probleme mit diesem Computer zu lösen:

1. **Bubble-Sort:** Um ein Array zu sortieren (1, 2, 3 ...) kann man auf den sog. Bubble-Sort zurückgreifen. Dieser hat im durchschnittlichen Fall eine Komplexität von $O(n^2)$. Es soll nun eine Array mit 275.000 (zweihundertfünfundsiebzigtausend) Elementen sortiert werden. Wie lange benötigt der Rechner hierfür?

2. **Die Türme von Hanoi:** Das Spiel besteht aus drei gleich großen Stäben A, B und C, auf die mehrere gelochte Scheiben gelegt werden, alle verschieden groß. Zu Beginn liegen alle Scheiben auf Stab A, der Größe nach geordnet, mit der größten Scheibe unten und der kleinsten oben. Ziel des Spiels ist es, den kompletten Scheiben-Stapel von A nach C zu versetzen.

   Bei jedem Zug darf die oberste Scheibe eines beliebigen Stabes auf einen der beiden anderen Stäbe gelegt werden, vorausgesetzt, dort liegt nicht schon eine kleinere Scheibe. Folglich sind zu jedem Zeitpunkt des Spieles die Scheiben auf jedem Feld der Größe nach geordnet.

   Auf Stapel A liegen nun 64 Scheiben. Wie lange benötigt ihr Computer, um den Stapel mit dem in der Vorlesung genannten Algorithmus zu versetzen?
   Tipp: Sie müssen zunächst die Komplexitätsklasse des Spiels ermitteln, die sicherlich größer als $\mathcal{O}(n)$ sein wird, um die Anzahl der Spielzüge zu ermitteln!