

Trainingsblatt 7 zu Algorithmische Mathematik

Dieses Blatt bezieht sich auf die Videos 212–304.

Aufgabe 1: Gegeben sind die Zahlen $-12, 3, 7, 4, 51, 33, -45, 7, 7, -73, 0, 2$. Nutzen Sie das Bucket-sort-Verfahren aus Video 212, um diese zu sortieren. Gehen Sie hierzu wie folgt vor:

- Definieren Sie eine geeignete monotone Funktion, die für die gegebenen Zahlen innerhalb des Verfahrens verwendet werden kann. (*Tipp: Suchen Sie eine lineare Funktion, die die gegebenen Zahlen in den positiven Bereich bringt und gleichzeitig alle Werte auf das Intervall $[0, 1)$ zusammenstaucht.*) (1 Punkt)
- Gehen Sie von $k = 5$ Buckets aus und sortieren Sie die gegebenen Zahlen entsprechend der im Verfahren genannten Regel und basierend auf Ihrer Funktion händisch in die Buckets. (2 Punkte)
- Sortieren Sie alle Buckets einzeln (hier dürfen Sie z.B. „per Hand“ sortieren) und reihen Sie die Inhalte der Buckets aneinander. (1 Punkt)
- Wie lautet eine allgemeine Funktion $f : \mathbb{R} \rightarrow [0, 1)$, die für das Bucketsort-Verfahren verwendet werden kann? (*Tipp: Gesucht ist eine Funktion, die unabhängig von den hier konkret gegebenen Werten funktioniert. Gehen Sie davon aus, dass die größte Zahl m und die kleinste Zahl n heißt.*) (2 Punkte)

Aufgabe 2: Bestimmen Sie Paper-and-Pencil-basiert $\text{ggT}(391, 544)$ mithilfe der in Video 301 gezeigten rekursiven Variante des modernen Euklidischen Algorithmus. Gehen Sie hierbei wie im Diagramm auf Folie 5 des Videos vor, so dass sich die einzelnen Aufrufe nachvollziehen lassen. (4 Punkte)

Aufgabe 3 ★: Die Fakultät $n! := 1 \cdot 2 \cdot \dots \cdot (n-1) \cdot n$ definiert man i.d.R. wie eben geschehen. Sie lässt sich aber auch durch eine rekursive Definition erklären, nämlich:

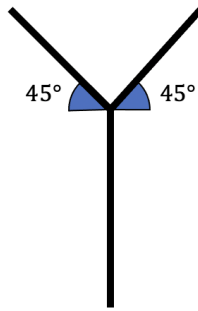
$$n! := \begin{cases} 1 & n = 1 \\ (n-1)! \cdot n, & n > 1 \end{cases}$$

Schreiben Sie auf Basis dieser Definition ein Scratch-Programm, das rekursiv programmiert ist. (*Tipp: Gesucht ist also ein Programm, das nicht iterativ, d.h. z.B. über eine Schleife, arbeitet. Gehen Sie stattdessen so vor, dass Sie einen Eigenen Block nutzen, der in seinem eigenen Programmcode auf sich selbst zugreift.*) (5 Punkte)

Aufgabe 4 ★: Ziel dieser Aufgabe ist das Erstellen einer rekursiven Grafik, wie in Video 304 vorgestellt. Hierzu ist das folgende Regelsystem vorgegeben:

- Wenn du eine Linie zeichnest, zeichne zwei weitere von zwei Dritteln Länge am Ende der Linie. Die beiden Linien sollen um 45° nach links bzw. rechts vom Ende der vorherigen Linie abknicken.
- Zeichne eine Linie.

Die untenstehende Skizze illustriert diese Regeln nach einmaliger (noch nicht rekursiver) Ausführung.



Erstellen Sie ein Scratch-Programm, das diese Regeln bis zu einer vorgegebenen Grundlänge der Linie sowie einer vorgegebenen Rekursionstiefe ausführt. Beide Größen sollen hierbei über Schieberegler frei einstellbar sein. (*Tip: Versuchen Sie die Figur zunächst wie dargestellt und noch nicht rekursiv fortgesetzt mit Scratch als Turtle-Grafik zu erstellen. Erst danach fügen Sie die Rekursion ein. Sollte es nicht auf Anhieb funktionieren, betrachten Sie die Situation für eine kleine Rekursionstiefe Schritt für Schritt (z.B. mit Warte-Blöcken ausgebremsst). Schieberegler erzeugen Sie außerdem, indem Sie auf eine auf der Bühne dargestellte Variable rechts klicken und „Schieberegler“ wählen.*) (10 Punkte)