

40. Bundeswettbewerb für Informatik - Runde 2 -

Aufgabe 2 (Zahlenrätsel)

Aufgabenstellung

Schreibe ein Programm, das Rätsel nach Gwendolines Schema erstellt. Die Rätsel sollten interessant und unterschiedlich sein (also z.B. nicht $2 \cdot 2 \cdot 2 \cdot 2 = 16$). Das Programm sollte für eine gegebene Anzahl von Operatoren ein Rätsel so erzeugen, dass

- a) das Rätsel eindeutig lösbar ist (also nicht $3 \cdot 4 \cdot 3 = 15$, welches zwei Lösungen hat),
- b) die Operanden einzelne Ziffern sind,
- c) eine positive ganze Zahl als Ergebnis herauskommt,
- d) Punkt- vor Strich-Rechnung angewandt wird,
- e) gleichrangige Operatoren linksassoziativ angewandt werden (also z.B. $6 : 3 * 2 = 4$ oder $5 - 2 + 1 = 4$) und
- f) alle Zwischenergebnisse ganze Zahlen sind (also nicht $3 : 2 * 4 = 6$).

Lasse dein Programm für verschiedene Operatorenanzahlen laufen und zeige uns mehrere deiner Ergebnisse. Schaffst du es, Rätsel mit 10, 15 oder sogar noch mehr Operatoren zu generieren? Hier ist so ein Rätsel:

$$4 \cdot 3 \cdot 2 \cdot 6 \cdot 3 \cdot 9 \cdot 7 \cdot 8 \cdot 2 \cdot 9 \cdot 4 \cdot 4 \cdot 6 \cdot 4 \cdot 4 \cdot 5 = 4792$$

Lösungsidee

Das Hauptziel des Algorithmuses ist, dass ein Zahlenrätsel mit einer vorgegebenen Länge generiert wird, welches die in der Aufgabenstellung gegebenen Regeln erfüllt.

Verwendung des Programmes

Das Programm zur Lösung dieses Problemes befindet sich in der Datei "Aufgabe_2.jar". Das Programm kann mit der Befehlszeile (CMD auf Windows, bzw. Terminal auf MacOS) ausgeführt werden. Dafür navigiert man zuerst in den Ordner der JAR-Datei (hier Aufgabe 2). Anschließend führt man den Befehl "java -jar Aufgabe_2.jar <Anzahl Operatoren> <Ausgabedatei>" aus. Die Ausgabedatei ist optional. Wenn keine Ausgabedatei angegeben ist, dann wird das Ergebnis als "output.txt" in dem Ordner der JAR-Datei gespeichert.

Implementierung

Die Implementierung des Algorithmus besteht im Kern aus den folgenden drei Klassen, welche unterschiedliche Aufgaben übernehmen:

- EquationGenerator (Generierung der Gleichung)
- EquationCalculator (Berechnung von Zwischenergebnissen)
- EquationVerifier (Überprüfung der Einzigartigkeit der Gleichungen)

Darüber hinaus werden die folgenden Hilfsklassen verwendet:

- NumberGenerator (Zufallsgenerator für die einzelnen Ziffern)
- Operators (Variablen und Funktionen für die verschiedenen Operatoren)
- Utils (Hilfsfunktionen, beispielsweise Bestimmung von geraden/ungeraden Zahlen)

Generieren der Gleichungen

Grundannahmen und Regeln

Die mathematischen Gleichungen, welche durch den Algorithmus generiert werden, besitzen die folgende Form:

$$x_1 X x_2 Y x_3 Z x_4 \dots x_n = y$$

- $x_1 \dots x_n \in \mathbb{N} \wedge 0 < x_1 \dots x_n < 10$
- $x_1 X x_2 \in \mathbb{N} \wedge x_1 X x_2 > 0$
- $y \in \mathbb{N} \wedge y > 0$

Hierbei repräsentieren die Variablen $x_1 \dots x_n$ die Ziffern und die Variablen $X \dots Z$ die Operatoren der Gleichungen.

Zusätzlich zu dieser Definition gelten die folgenden Regeln:

- es dürfen keine Operatoren **mehrfach** hintereinander vorkommen (*max. 1*)
- es dürfen keine Ziffern **mehrfach** hintereinander vorkommen (*max. 2*)
- es wird die **Punkt- vor Strichrechnung** berechnet/angewandt
- gleichrangige Operatoren werden **linksassoziativ** berechnet
- die Gleichungen müssen **eindeutig lösbar** sein (eine Lösung)

Diese Einschränkungen sollen gewährleisten, dass die generierten Gleichungen sowohl abwechslungsreich, als auch einzigartig sind und die Chance, eine Gleichung mehrfach zu erhalten somit möglichst gering gehalten wird.