Aufgabe 2: Geburtstag

Richard Wohlbold
Team-ID: 00487

30. Dezember 2019

Inhaltsverzeichnis

1 Lösungsidee

1.1 Generierung der Tabelle

Meine Lösungsidee für das Problem besteht darin, alle möglichen Terme, die durch die gegebenen Rechenoperationen aus der gegebenen Ziffer erhalten werden können, in einer Tabelle zu speichern. Dabei wird nach der Anzahl der vorkommenden Ziffern verfahren: Angefangen wird bei $n = 1$ Ziffern. Für $n = 1$ Ziffern lässt sich ohne Berücksichtigung der Fakultätsfunktion nur ein Term finden, nämlich die Ziffer selbst. Für $n = [2, \infty)$ Ziffern werden Terme mit einer geringeren Anzahlen an Ziffern i,j über die gegebenen Grundrechenarten kombiniert, sodass $n = i + j$. Dabei sollen die Terme bei den kommutativen Grundrechenarten ($+$, $*$) nicht vertauscht werden, da dort dasselbe Ergebnis entsteht, bei den nicht-kommutativen Grundrechenarten jedoch schon, da oft verschiedene Ergebnisse auftreten.

Um beispielsweise alle Terme für $n = 5$ zu finden, werden alle Terme mit $i = 1$ und $j = 4$ und mit $i = 2$ und $j = 3$ über das kartesische Produkt kombiniert und für alle Rechenarten und Reihenfolgen gespeichert.

Es gibt vier Grundrechenarten, von denen zwei nicht-kommutativ sind. Dadurch ergeben sich sechs mögliche neue Terme durch jedes Termpaar, davon ausgehend, dass keine Dopplungen auftreten und alle Terme valide sind (z.B. keine Division durch 0):

$$N(1) = 1$$
$$N(n) = \sum_{i=1}^{n//2} 6i(n-i)$$

Es ergeben sich die N, die in Abbildung 1 zu sehen sind.

n	$N(n)$
1	1
2	6
3	36
4	432
5	3.888
6	46.656
7	513.216
8	6.718.464
9	78.941.952
10	1.038.002.688
11	12.939.761.664
12	174.505.383.936

Abbildung 1: Anzahl der möglichen Terme N nach der Anzahl der vorkommenden Ziffern n

1.2 Scannen

Um die Laufzeit des Programms zu verbessern, wartet der Algorithmus nicht darauf, bis die Zahl in der Tabelle auftaucht, sondern überprüft, ob man zwei Terme aus der Tabelle durch eine Grundrechenart kombinieren kann, um auf das gewünschte Ergebnis zu kommen. Dieses Verfahren nenne ich *Scan*. Dazu wird für jede Zahl j, für die ein Term in der Tabelle steht, ein Partner p berechnet, mit dem das gewünschte Ergebnis e erhalten werden kann. Dafür gibt es verschiedene Möglichkeiten:

$$
\begin{aligned}
e = j + p &\quad\Leftrightarrow\quad p = e - j \\
e = j - p &\quad\Leftrightarrow\quad p = j - e \\
e = p - j &\quad\Leftrightarrow\quad p = j + e \\
e = p * j &\quad\Leftrightarrow\quad p = \frac{e}{j} \\
e = p / j &\quad\Leftrightarrow\quad p = e * j \\
e = j / p &\quad\Leftrightarrow\quad p = \frac{j}{e}
\end{aligned}
$$

Für jedes j werden nun alle möglichen Partner p und es wird geschaut, ob p Teil der Tabelle ist. Ist dies der Fall, wird es einer Menge hinzugefügt. Der Term mit der geringsten Anzahl an Ziffern der Menge ist ein mögliches Ergebnis.

1.3 Zusammenfügen

Es wird immer abwechselnd eine Tabelle mit einer Ziffer mehr als zuvor generiert und ein *Scan* ausgeführt. Dies wird solange wiederholt, bis ein Scan ein Ergebnis mit m Ziffern zurückgibt und eine Tabelle mit $m - 2$ Ziffern vorliegt und ein Scan ausgeführt wurde.

An diesem Punkt wird die beste Lösung verwendet, da eine Lösung mit $n = m-2$ Ziffern in der Tabelle stehen würde, der Scan eine Lösung mit $n = m - 1$ Ziffern erkannt hätte und eine Lösung mit $n = m$ Ziffern bereits vorliegt.

Der Algorithmus kann nicht schon bei der Tabelle mit $m - 3$ Ziffern terminieren, da eine Lösung, die sich aus Termen mit einer Ziffer und $m - 2$ Ziffern zusammensetzt, durch einen *Scan* nicht erkannt werden würde, sodass der Algorithmus nicht die beste Lösung liefern würde. Algorithmus 1 zeigt die Kombination der Tabellengeneration und des Scans.

1.4 Fakultät und Potenzieren

hinzufügen

2 Umsetzung

hinzufügen

Algorithm 1 Kombinieren von GENERATE und SCAN, um den Term mit der geringsten Anzahl an Ziffern zu finden

```
1: i ← 1
2: n ← ∞
3: r ← nil
4: while i ≤ n − 2 do
5:     GENERATE(i)
6:     r ← SCAN
7:     if r ≠ nil then
8:         n ← DIGITS(r)
9:     end if
10:    i ← i + 1
11: end while
12: return r
```

2.1 Laufzeitanalyse

hinzufügen

3 Beispiele

hinzufügen

4 Quellcode (ausschnittsweise)

hinzufügen

Liste der noch zu erledigenden Punkte