Übungen zur Vorlesung „Mathematik I"

Aufgabe 1. Vollständige Induktion 2 *Oft hat man Aussagen, die man für unendlich viele Beispiele nachprüfen müsste. Wenn man dies nicht tun will, kann man mit dieser "Domino-Effekt"-Beweismethode mit wenigen Schritten unendlich viele Aussagen auf einmal nachweisen, ohne den Rechner anschmeißen zu müssen (und sich damit eine Menge wertvolle Rechenzeit sparen). Zum Beispiel kann man so lange Summen mit einfachen Formeln ausdrücken:*
Aufgabe: Zeigen Sie mit Hilfe vollständiger Induktion die Richtigkeit der foglenden Aussagen für alle natürlichen Zahlen n:

(a) $2^n > n^2$ für $n \geq 5$.

(b) $n^3 + 11 \cdot n$ ist durch 6 teilbar.

Aufgabe 2. Zahlsysteme *Computer funktionieren im Binärsystem bzw. Hexademizalsystem (für effektives Speichern, weil, wie wir hier sehen werden, vier Binär-Stellen ersetzt werden können durch eine Hexadezimalstelle), daher brauche ich hierzu wohl nichts weiter zu motivieren.*

(a) Wandeln Sie folgende Zahlen vom Binärsystem ins Dezimalsystem um:

- 1111
- 101010

(b) Wandeln Sie folgende Zahlen vom Dezimalsystem ins Binärsystem um:

- 127
- 1024

(c) Wandeln Sie die Hexadezimalzahl $(ABC012)_{16}$ in eine Binärzahl um, indem Sie die einzelnen Ziffern $A, B, C, 0, 1, 2$ des Hexadezimalsystems jeweils als viertellige Binärzahl schreiben.

Aufgabe 3. Euklidischer Algorithmus, ggT *Wenn man den ggT von zwei kleinen Zahlen bestimmen will, geht das oft noch im Kopf: Man geht im Kopf die möglichen Teiler der Zahlen durch und bestimmt so die tatsächlichen. Bei sehr großen Zahlen wird das jedoch sehr sehr aufwändig. Besser ist es, einen Algorithmus zu benutzen, der garantiert zum Ziel führt. Noch besser: dieser Algorithmus liefert* *uns eine Methode, wie man den $ggT(a, b)$ als Linearkombination (also die Summe von Vielfachen) von a und b schreiben kann. Wofür man das braucht? Ein Beispiel beim Rechnen in Restklassen sehen wir in der nächsten Übung.*

(a) Bestimmen Sie mit Hilfe des Euklidischen Algorithmus den größen gemeinsamen Teiler $d = \mathrm{ggT}(a, b)$ von $a = 3192$ und $b = 2079$, und geben Sie ganze Zahlen $s, t \in \mathbb{Z}$ an, so dass $s \cdot a + t \cdot b = d$.

(b) Verwenden Sie den Euklidischen Algorithmus, um Zahlen $s, t \in \mathbb{Z}$ mit $s \cdot 49 + t \cdot 21 = -14$ zu finden! Hinweis: berechnen Sie erst s', t' mit $s' \cdot 49 + t' \cdot 21 = ggT(49, 21)$!

(c) Gibt es s, t, so dass $s \cdot 49 + t \cdot 21 = 15$?

Aufgabe 4. Rechnen mit Restklassen *Jeden Tag rechnet man beim Bestimmen der Uhrzeit modulo 12. Restklassen werden aber auch in zahlreichen Informatik-Anwendungen verwendet, zum Beispiel bei Hash-Algorithmen (siehe nächstes Übungsblatt) und in Verschlüsselungsverfahren.* Berechnen Sie folgende Werte:

(a) $-7 \bmod 6$, $49 \bmod 6$, $-1 \bmod 6$, $-16 \bmod 6$, $-16 \bmod 7$, $-34 \bmod 5$

(b) $\overline{7 \cdot 13 \cdot 19 \cdot 40}$ in $\mathbb{Z}/6\mathbb{Z}$

(c) $\overline{4} \odot (\overline{11} \odot \overline{3} \oplus \overline{1})$ in $\mathbb{Z}/27\mathbb{Z}$

(d) $(\overline{-21} \odot \overline{2}) \oplus \overline{16}$in $\mathbb{Z}/27\mathbb{Z}$

(e) $\overline{7^{123}}$ in $\mathbb{Z}/6\mathbb{Z}$

Aufgabe 5. Hashing mit Restklassen *Wir haben in der Vorlesung besprochen, warum Hashing im realen Leben oft zum Einsatz kommt: wenn Sie zum Beispiel einen Online-Handel betreiben, wollen Sie Ihre Kunden mit zehnstelligen Kundennummern ungern in einer Liste mit* 10^{10} *Zeilen abspeichern. Für diesen Fall verwendet man Hashing, um die Kunden mit wenig Speicherbedarf und kurzen Suchzeiten mit System in einer Liste abspeichern zu können.* Gegeben sei der folgende Datensatz bestehend aus Name und zugehöriger Identnummer:

$$\begin{aligned}D :=\{&(\text{Anton}, 5929), (\text{Berta}, 13495), (\text{Carla}, 10269), (\text{Friedrich}, 9541), (\text{Dora}, 7558),\\ &(\text{Emil}, 7560), (\text{Gerda}, 11080), (\text{Hans}, 5871), (\text{Ida}, 5872), (\text{Xaver}, 9438),\\ &(\text{Bernd}, 9447), (\text{Max}, 2078)\}\end{aligned}$$

Wir betrachten folgende Hashing-Funktion

$$\begin{aligned} h : D &\mapsto \{0, \dots, 12\} \\ (x, y) &\mapsto y \mod 13 \end{aligned}$$

(a) Ist h kollisionsfrei?

(b) Verwenden Sie lineare Sondierung, um die Einträge des Datensatzes in einer Liste (der Länge 13) einzusortieren.

(c) Ist es auch möglich mittels quadratischer Sondierung eine solche Liste zu erstellen?