

Übungen zu Informatik II (Sommersemester 2008) Musterlösung zu Aufgabenblatt 1

Hinweis: Auf dem Aufgabenblatt war in der ersten Fassung ein Fehler – die Einsicht aus Aufgabe 1 muss korrekt lauten: Eine Zahl $p \in \mathbb{N}$ mit $p > 1$ ist eine Primzahl, falls $b \nmid p$ für alle $b \in \{2, \dots, p-1\}$ gilt.

Aufgabe 1 (Präsenzaufgabe)

Voraussetzung: Seien $p, c \in \mathbb{N}$ mit $p > 1$, $c^2 > p$ und $c < p$.

Behauptung: Die Zahl p ist eine Primzahl genau dann, wenn $b \nmid p$ für alle $b \in \{2, \dots, c\}$.

Beweis: Es ist ausreichend, die beiden folgenden Behauptungen zu zeigen:

1. Wenn p eine Primzahl ist, dann gilt $b \nmid p$ für alle $b \in \{2, \dots, c\}$.
2. Wenn p keine Primzahl ist, dann gilt $b \nmid p$ nicht für alle $b \in \{2, \dots, c\}$.

Wir zeigen diese beiden Teile nacheinander. Zunächst zu 1. Sei p eine Primzahl. Nach Definition wissen wir: Die Zahl p hat außer 1 und p keinen weiteren Teiler, also gilt $b \nmid p$ für alle $b \in \{2, \dots, p\}$. Insbesondere gilt wegen $c < p$ auch $b \nmid p$ für alle $b \in \{2, \dots, c\}$.

Zu 2. Nun sei p keine Primzahl. Dann müssen wir zeigen, dass $b \nmid p$ nicht für alle $b \in \{2, \dots, c\}$ gilt, also dass wir mindestens ein $b \in \{2, \dots, c\}$ mit $b \mid p$ finden können.

Da p keine Primzahl ist, gibt es einen Teiler $t \in \{2, \dots, p-1\}$. Dann gibt es auch eine natürliche Zahl $t' \in \mathbb{N}$ mit $p = t \cdot t'$, d.h. auch t' ist ein Teiler von p .

Ohne Beschränkung der Allgemeinheit nehmen wir an¹, t sei kleiner oder gleich t' . Dann gilt $a = t \cdot t' \geq t \cdot t = t^2$. Aus $c^2 > a$ und $a \geq t^2$ folgt nun $0 < c^2 - t^2 = (c-t)(c+t)$, also $c > t$. Damit haben wir $t \in \{2, \dots, c\}$ gefunden wie gewünscht.

Damit haben wir die Behauptung bewiesen.

Aufgabe 2 (Präsenzaufgabe)

Eine einfache Variante des Algorithmus':

Algorithmus isPrime1(a)

Vorbedingung: $a \in \mathbb{N}$.

```
falls  $a \leq 1$ 
    gib 0 zurück
setze  $b = 2$ 
solange  $b^2 \leq a$ 
    falls  $b \mid a$ 
        gib 0 zurück
    setze  $b = b + 1$ 
gib 1 zurück
```

Nachbedingung: $a = \bar{a}$ und return = 1 gdw. \bar{a} ist Primzahl.

Dabei steht return für den Rückgabewert des Algorithmus'.

¹Wir können dies annehmen, da der andere Fall, in dem t' kleiner oder gleich t ist, völlig analog behandelt werden kann, nur mit vertauschten Rollen für die Zahlen t und t' .

Eine Variante, die ohne Multiplikation auskommt, indem sie laufend b^2 aus $(b-1)^2$ bestimmt (und dabei ausnutzt, dass für $c \in \mathbf{N}$ die Gleichung $(c+1)^2 = c^2 + 2c + 1$ gilt):

Algorithmus isPrime2(a)

Vorbedingung: $a \in \mathbf{N}$.

```

falls  $a \leq 1$ 
    gib 0 zurück
setze  $b = 2$ 
setze  $bq = 4$ 
solange  $bq \leq a$ 
    falls  $b \mid a$ 
        gib 0 zurück
    setze  $bq = bq + b + b + 1$ 
    setze  $b = b + 1$ 
gib 1 zurück
    
```

Nachbedingung: $a = \bar{a}$ und return = 1 gdw. \bar{a} ist Primzahl.

Aufgabe 3 (Hausaufgabe)

Voraussetzung: Seien $a, b \in \mathbf{N}$ mit $b > 0$.

Behauptung:

1. Es gilt $a \bmod b = a$, falls $a < b$.
2. Es gilt $a \bmod b = (a - b) \bmod b$, falls $a \geq b$.

Beweis: Zunächst zeigen wir, dass generell $a \bmod b < b$ gilt für alle $a, b \in \mathbf{N}$ mit $b > 0$.

Angenommen, dies wäre nicht so. Sei also $c = a \bmod b$ mit $c \geq b$. Dann existiert nach Definition ein $d \in \mathbf{N}$ mit $a = bd + c$. Nun können wir aber $c' = c - b$ und $d' = d + 1$ wählen, womit $a = bd' + c'$ gilt. Somit haben wir eine Zahl c' gefunden, die *kleiner* ist als c . Das widerspricht der Definition der modularen Reduktion, also kann c nicht gleich $a \bmod b$ sein.

1. Sei $a < b$. Laut Modulo-Definition ist $a \bmod b$ die kleinste Zahl $c \in \mathbf{N}$, so dass ein $d \in \mathbf{N}$ existiert mit $a = bd + c$.

Für $c = a$ und $d = 0$ die Gleichung erfüllt. Für $d' > 0$ gilt aber schon $a < bd' + c'$ für beliebige $c' \in \mathbf{N}$. Daher ist $c = a$ die einzige und damit auch kleinste Zahl, die obige Gleichung erfüllt.

2. Sei $a \geq b$. Seien $d, d' \in \mathbf{N}$ derart, dass $a = bd + (a \bmod b)$ und $a - b = bd' + (a - b) \bmod b$, diese Zahlen gibt es nach Definition der modularen Reduktion. Dann erhalten wir $(a \bmod b) - (a - b \bmod b) = (d' - d + 1)b$.

Aus $a \bmod b < b$ folgt:

$$-b < (a \bmod b) - (a - b \bmod b) < b.$$

Also gilt auch $-b < (d' - d + 1)b < b$. Damit gilt $-1 < d' - d + 1 < 1$, also $d - d' + 1 = 0$, woraus $a \bmod b = a - b \bmod b$ folgt.

Aufgabe 4 (Hausaufgabe)

Algorithmus mod(a, b)

Vorbedingung: $a, b \in \mathbf{N}$ und $b > 0$.

```

solange  $a \geq b$ 
    setze  $a = a - b$ 
    
```

Nachbedingung: $a = \bar{a} \bmod \bar{b} \wedge b = \bar{b}$.