Elementare Zahlentheorie mit Maple (01202) **SoSe 2010**

Klausur am 25.09.2010:

Musterlösungen

Aufgabe 1

Seien $m = 2a + 1$ und $n = 2b + 1$ mit $a, b \in \mathbb{N}_0$. Dann gilt

$$\begin{aligned} m^2 - n^2 &= 4a^2 + 4a + 1 - 4b^2 - 4b - 1 = 4a^2 + 4a - 4b^2 - 4b \\ &= 4(a^2 + a - b^2 - b). \end{aligned}$$

Wir zeigen, dass $a^2 + a - b^2 - b$ gerade ist. Es ist $a^2 + a - b^2 - b = a(a + 1) - b(b + 1)$. Unabhängig davon, ob a oder b gerade oder ungerade sind, sind $a(a + 1)$ und $b(b + 1)$ gerade. Es folgt, dass $a^2 + a - b^2 - b$ gerade ist, dass also $m^2 - n^2 = 4(a^2 + a - b^2 - b)$ durch 8 teilbar ist.

Aufgabe 2

Sei $n \in \mathbb{N}$. Angenommen, $n^3 - 1$ ist eine Primzahl. Dann ist $n > 1$, denn $1^3 - 1 = 0$ ist keine Primzahl. Es ist $n^3 - 1 = (n - 1)(n^2 + n + 1)$. Der Faktor $n^2 + n + 1$ ist größer als 1, daher muss $n - 1 = 1$ sein, also $n = 2$. Wir haben also gezeigt: Ist $n^3 - 1$ eine Primzahl, so ist $n = 2$. Ist umgekehrt $n = 2$, so ist $n^3 - 1 = 8 - 1 = 7$ eine Primzahl. Es ist also $n^3 - 1$ genau dann eine Primzahl, wenn $n = 2$ ist.

Aufgabe 3

Die Primzahl p ist kein Teiler von $1, \ldots, p - 1$. Mit dem Kleinen Satz von Fermat gilt $k^{p-1} \equiv 1 (\mathrm{mod} p)$ für alle $1 \leq k \leq p - 1$. Es folgt

$$1^{p-1} + 2^{p-1} + \cdots + (p-1)^{p-1} \equiv \underbrace{1 + 1 + \cdots 1}_{p-1 \text{ Mal}} \equiv p - 1 \equiv -1 (\mathrm{mod} p).$$

Aufgabe 4

Sei $n = p_1^{e_1} \cdots p_r^{e_r}$ die kanonische Primfaktorzerlegung von n. Angenommen, es gibt einen Primteiler p_i mit $p_i > 2$. Dann gilt $\varphi(p_i^{e_i}) = p_i^{e_i - 1}(p_i - 1)$. Da p_i ungerade ist, ist $p_i - 1$ gerade und $\varphi(n) = \varphi(p_1^{e_1}) \cdots \varphi(p_r^{e_r})$ ist gerade.

Angenommen, es gibt kein p_i mit $p_i > 2$. Dann ist $n = 2^r$, und da $n > 2$ ist, folgt $r \geq 2$. Dann gilt $\varphi(2^r) = 2^{r-1}$, und da $r \geq 2$ ist, folgt, dass $\varphi(n)$ gerade ist.

Aufgabe 5

Das Tripel $(a, a + 2, c)$ kann kein primitives pythagoreisches Tripel sein, denn die Zahlen a und $a + 2$ sind beide gerade oder beide ungerade. In einem primitiven pythagoreischen Tripel muss aber eine der Zahlen gerade, die andere ungerade sein. Das Tripel $(6, 8, 10)$ ist von der Form $(a, a + 2, c)$ und erfüllt $6^2 + 8^2 = 100 = 10^2$.

Aufgabe 6

Wir zerlegen 2010 in Primfaktoren. Es ist $2010 : 2 = 1005$, $1005 : 5 = 201$ und $201 : 3 = 67$. Die Primfaktorzerlegung von 2010 ist also

$$2010 = 2 \cdot 3 \cdot 5 \cdot 67.$$

Der Primfaktor 3 ist kongruent zu 3 modulo 4, und er tritt mit ungeradem Exponenten in der Primfaktorzerlegung auf. Es folgt, dass 2010 nicht Summe von zwei Quadraten ist.

Aufgabe 7

Sei $z \in \mathbb{Z}[i]$ eine Gauß'sche Primzahl. Dann gibt es genau eine Primzahl $p \in \mathbb{N}$ mit $z \mid p$. Ist $p \equiv 2(\mathrm{mod}4)$, also $p = 2$, so ist $z = e(1 + i)$, und e ist eine Einheit in $\mathbb{Z}[i]$. Dann gilt $N(z) = N(e)N(1 + i) = 2$, und 2 ist eine Primzahl. Ist $p \equiv 3(\mathrm{mod}4)$, so ist $z = ep$ für eine Einheit e, und es gilt $N(z) = N(e)N(p) = p^2$. Ist $p \equiv 1(\mathrm{mod}\,4)$, so ist $p = ez\overline{z}$ und es gilt $p^2 = N(p) = N(e)N(z)N(\overline{z})$, also $N(z) = p$.

Aufgabe 8

1. Die Variable p, die in Zeile 5 auf 1 gesetzt wird, ist in Zeile 2 nicht als lokale Variable deklariert worden. Richtig (beziehungsweise besser) wäre also

```
>  #Die Zeilennummerierung ist zu Ihrer Orientierung
1.  klausur:=proc(a::integer)
2.     local i,n,l,p;
3.     l:=ifactors(a);
4.     n:=nops(l[2]);
5.     p:=1;
6.     for i from 1 to n do
7.      p:=p*l[2][i][1]
8.     od;
9.     print(p);
10.  end:
```

2. Für $a = 112$ wird in Zeile 3 die Liste l auf `ifactors`$(112) = [1, [[2, 4], [7, 1]]]$ gesetzt, denn es ist ja $112 = 2^4 \cdot 7^1$. In Zeile 4 ist n die Anzahl der Einträge in der zweiten Liste in l, also die Anzahl der Einträge in $[[2, 4], [7, 1]]$, also $n = 2$. In Zeile 5 wird $p = 1$ gesetzt. Im ersten Durchlauf der Schleife in den Zeilen 6 bis 8 ist $i = 1$, und es wird p auf $p = 1 \cdot l[2][1][1] = 1 \cdot 2 = 2$ gesetzt. Der zweite und letzte Durchlauf der Schleife für $i = 2$ setzt $p = 2 \cdot l[2][2][1] = 2 \cdot 7 = 14$. Die Prozedur endet also mit $p = 14$, und dieser Wert wird in Zeile 9 auf den Bildschirm ausgegeben.

3. Bei Eingabe einer Zahl $a = \pm p_1^{e_1} \cdot \ldots \cdot p_k^{e_k}$ ist `ifactors`$(a) = [\pm 1, [[p_1, e_1], \ldots, [p_k, e_k]]]$. In der Schleife in den Zeilen 6 bis 8 werden die p_i's aus dieser Liste multipliziert. Ausgabe der Prozedur ist also $p_1 \cdot \ldots \cdot p_k$.

4. Eine Prozedur, die bei Eingabe von $a = \pm p_1^{e_1} \cdot \ldots \cdot p_k^{e_k}$ die Ausgabe $e_1 + \ldots + e_k$ liefert, ist die folgende:

```
> #Die Zeilennummerierung ist zu Ihrer Orientierung
1.  klausur:=proc(a::integer)
2.     local i,n,l,p;
3.     l:=ifactors(a);
4.     n:=nops(l[2]);
5.     p:=0;
6.     for i from 1 to n do
7.      p:=p+l[2][i][2]
8.     od;
9.     print(p);
10.  end:


> klausur(112);
```