Computer algebra:

Primality testing and integer factorisation

Friedrich-Schiller-Universität Jena, WS 2014

Jürgen Müller

Contents

1 Integer arithmetic

(1.1) Landau symbols. Let $\mathcal{F}$ be the set of all functions $f\colon D_f \to \mathbb{R}_{>0}$, where $D_f \subseteq \mathbb{N}_0$ is a co-finite subset, that is all functions defined for almost all non-negative integers and taking positive real values. Then for $f \in \mathcal{F}$ the **Landau symbols** $O(f)$ and $o(f)$ are the sets of all functions $g \in \mathcal{F}$ such that the sequence $[\frac{g(n)}{f(n)} \in \mathbb{R}_{>0}; n \in D_f \cap D_g]$ is bounded, and such that $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$, respectively; hence we have $o(f) \subseteq O(f)$.

Functions $f, g \in \mathcal{F}$ are called **asymptotically equivalent** if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 1$; we write $f \sim g$, and in this case we have $f \in O(g)$ and $g \in O(f)$. We use a similar notation for functions in several variables, and for functions defined on right unbounded subsets of $\mathbb{R}$.

For example, Stirling's formula $\lim_{n\to\infty} \frac{n!\cdot e^n}{n^n\cdot\sqrt{2\pi n}} = 1$ says that $n! \sim (\frac{n}{e})^n \cdot \sqrt{2\pi n}$, hence $\ln(n!) \sim n(\ln(n) - 1) + \frac{1}{2} \cdot \ln(n) + \ln(\sqrt{2\pi}) \sim n\ln(n)$. Moreover, letting $\mathcal{P} := \{p \in \mathbb{N}; p \text{ prime}\}$, and for $x \in \mathbb{R}$ letting $\mathcal{P}_{\leq x} := \{p \in \mathcal{P}; p \leq x\}$ and $\pi(x) := |\mathcal{P}_{\leq x}| \in \mathbb{N}_0$, the Prime Number Theorem says that $\pi(x) \sim \frac{x}{\ln(x)}$.

(1.2) Bit lengths and bit operations. The number of digits to the base $1 \neq z \in \mathbb{N}$ necessary to represent $n = \sum_{i=0}^{b-1} n_i z^i \in \mathbb{N}$, where $n_i \in \{0, \ldots, z-1\}$, is given as the **bit length** $b = b_z(n) = 1 + \lfloor \log_z(n) \rfloor = 1 + \lfloor \frac{\ln(n)}{\ln(z)} \rfloor \in \mathbb{N}$, where $\lfloor \cdot \rfloor$ denotes lower Gaussian brackets. For $n \in \mathbb{Z}$ we need an additional sign, hence the input length of $0 \neq n \in \mathbb{Z}$ into a Turing machine is $1 + b_z(|n|) \sim \frac{1}{\ln(z)} \cdot \ln(|n|)$, the representation of $|n|$ being the sequence $[n_0, \ldots, n_b]$.

The computational complexity of integer arithmetic is counted in **bit operations** on bit strings, hence with respect to the base $z = 2$: These are the operations 'and', 'or', 'exclusive or', 'not', and 'shift', Generalised bit operations are **Byte operations**, **word operations** and **long word operations**, with respect to the bases $z = 2^8$, $z = 2^{32}$ and $z = 2^{64}$, respectively. The time needed for these operations indeed is polynomial in the input length.

We treat bit operations as oracles. An algorithm, whose input up to sign is $n \in \mathbb{N}$, is called an $L_{\alpha,c}$**-time algorithm**, where $0 \leq \alpha \leq 1$ and $c > 0$, if it needs $O(L_{\alpha,c}(n))$ bit operations, where $L_{\alpha,c}(n) := \exp(c \cdot (\ln(n))^\alpha \cdot (\ln(\ln(n)))^{1-\alpha})$. Hence for $\alpha = 0$ we have $L_{0,c}(n) = \ln^c(n)$, thus runs in **polynomial time**; for $\alpha = 1$ we have $L_{1,c}(n) = \exp(c\ln(n)) = n^c$, thus runs in **exponential time**; and for $0 < \alpha < 1$ we have $cx^\alpha \ln(x)^{1-\alpha} \in o(x)$, see Exercise (6.2), thus runs in **subexponential time** $O(\exp(h(\ln(n))))$, for some function $h(x) \in o(x)$.

(1.3) Ring operations. These in general are addition, subtraction and multiplication, as well as division by units; the latter do not play a role for integers. The following algorithms are essentially the conventional techniques:

a) Addition. Let $n \geq m \in \mathbb{N}$ and $b := b_z(n) = \max\{b_z(n), b_z(m)\}$, for some $1 \neq z \in \mathbb{N}$. Hence we have $n = \sum_{i=0}^{b-1} n_i z^i$, where $n_i \in \{0, \ldots, z-1\}$, and we may assume $m = \sum_{j=0}^{b-1} m_j z^j$, where $m_j \in \{0, \ldots, z-1\}$, by letting $m_j := 0$ for $j \in \{b_z(m), \ldots, b-1\}$.

```
• $\delta \leftarrow 0$    # carry
• for $k \in [0, \ldots, b-1]$ do
    • $s_k \leftarrow n_k + m_k + \delta$    # $s_k \in \{0, \ldots, 2z-1\}$
    • if $s_k \geq z$ then
        • $s_k \leftarrow s_k - z$
        • $\delta \leftarrow 1$
    • else
        • $\delta \leftarrow 0$
• $s_b \leftarrow \delta$
• return $[s_0, \ldots, s_b]$
```

Hence we have $n + m = \sum_{k=0}^{b} s_k z^k$. For each $k \in \{0, \ldots, b\}$ this needs a fixed number of bit operations, and hence needs $O(b) = O(\ln(n))$ bit operations in total, that is runs in **linear time**. A similar analysis, see Exercise (6.4), shows that subtraction also needs $O(\ln(n))$ bit operations.

b) Multiplication. Let $b_n := b_z(n)$ and $b_m := b_z(m)$. Hence we have

$$nm = \sum_{i=0}^{b_n-1} \sum_{j=0}^{b_m-1} n_i m_j z^{i+j} = \sum_{k=0}^{b_n+b_m-2} \left(\sum_{l=\max\{0,k-b_m+1\}}^{\min\{b_n-1,k\}} n_l m_{k-l} \right) z^k.$$

```
• for $k \in [0, \ldots, b_n + b_m - 1]$ do
    • $s_k \leftarrow 0$
• for $i \in [0, \ldots, b_n - 1]$ do
    • $\delta \leftarrow 0$    # carry
    • for $j \in [0, \ldots, b_m - 1]$ do
        • $s \leftarrow s_{i+j} + n_i m_j + \delta$    # $s \in \{0, \ldots, z^2 - 1\}$
        • $s_{i+j} \leftarrow s \bmod z$    # $s = (s \bmod z) + \lfloor \frac{s}{z} \rfloor \cdot z$
        • $\delta \leftarrow \lfloor \frac{s}{z} \rfloor$    # $\delta \in \{0, \ldots, z-1\}$
    • $s_{i+b_m} \leftarrow \delta$
• return $[s_0, \ldots, s_{b_n+b_m-1}]$
```

Hence we have $nm = \sum_{k=0}^{b_n+b_m-1} s_k z^k$. For each i and j this needs a fixed number of bit operations, thus in total needs $O(b_z(n) b_z(m)) = O(\ln(n)\ln(m))$ bit operations. Hence for $n \geq m$ this amounts to $O(\ln^2(n))$ bit operations, that is runs in **quadratic time**.

(1.4) Fast multiplication [Karatsuba, 1962]. As for multiplication, we can do better. To this end, let $k \in \mathbb{N}_0$ and $b := 2^k$, and let $1 \neq z \in \mathbb{N}$ and $m, n \in \mathbb{N}$ such that $m, n < z^b$; hence we have $b_z(m), b_z(n) \leq b$.

For $k \geq 1$, let $m = m' \cdot z^{\frac{b}{2}} + m''$ and $n = n' \cdot z^{\frac{b}{2}} + n''$, for some $0 \leq m', m'', n', n'' < z^{\frac{b}{2}}$. Hence we have $0 \leq |m' - m''|, |n' - n''| < z^{\frac{b}{2}}$, and $m \cdot n = m'n'z^b + (m'n'' + m''n') \cdot z^{\frac{b}{2}} + m''n''$, where $m'n'' + m''n' = m'n' + m''n'' + (m' - m'')(n'' - n')$. Thus let $K(m, n, k)$ be the following algorithm:

```
• if $k = 0$ then
    • return $mn$
• else
    • $r \leftarrow K(m', n', k-1)$
    • $s \leftarrow K(m'', n'', k-1)$
    • $t \leftarrow K(|m' - m''|, |n' - n''|, k-1)$
    • $b \leftarrow 2^k$
    • return $rz^b + (r + s \pm t) \cdot z^{\frac{b}{2}} + s$
```

Hence by induction with respect to $k \in \mathbb{N}_0$ we have $K(m, n, k) = mn$. We show that this **divide-and-conquer** technique needs $O(b^{\log_2(3)})$ bit operations; assuming that $n \geq m$ and $\frac{b}{2} < b_z(n) \leq b$ this amounts to $O(\ln^{\log_2(3)}(n))$ bit operations, where $1 < \frac{158}{100} < \log_2(3) < \frac{159}{100} < 2$, thus runs in time strictly between quadratic and linear:

Let $\kappa(k) \in \mathbb{N}$ be the number of bit operations needed to compute $K(\cdot, \cdot, k)$. Then we may assume that $\kappa(0) = 1$, and for $k > 0$ we have three calls of $K(\cdot, \cdot, k-1)$ as well as additions and shifts, thus $\kappa(k) = 3 \cdot \kappa(k-1) + \gamma \cdot 2^k$, for some $\gamma > 0$. Thus by induction we get

$$\kappa(k) = 3^k \cdot \kappa(0) + \gamma \cdot \sum_{i=0}^{k-1} (3^i \cdot 2^{k-i}) = 3^k + 2^k \cdot \gamma \cdot \frac{(\frac{3}{2})^k - 1}{\frac{3}{2} - 1} = (2\gamma + 1) \cdot 3^k - \gamma \cdot 2^{k+1}.$$

Hence we have $\kappa(k) \in O(3^k) = O(3^{\log_2(b)}) = O((2^{\log_2(3)})^{\log_2(b)}) = O(b^{\log_2(3)})$. ♯

The best known integer multiplication algorithm, the **Schönhage-Strassen Algorithm [1971]** using **Fast Fourier Transform**, see [2, Ch.8.3], runs in **nearly-linear** time $O(\ln(n) \cdot \ln(\ln(n)) \cdot \ln(\ln(\ln(n)))) \subseteq O(\ln^{1+\epsilon}(n))$, for $\epsilon > 0$.

(1.5) Quotient and remainder. Let $m \geq n \in \mathbb{N}$, and let $q, r \in \mathbb{N}_0$ be the unique elements such that $r < n$ and $m = qn + r$; they are called the associated **quotient** and **remainder**, respectively, and we have $q = \lfloor \frac{n}{m} \rfloor$. Note that in particular to compute in $\mathbb{Z}_n$ the computation of remainders is needed. Again, the following is derived from the conventional technique to compute quotients:

Let $b := b_z(n)$ and $b' := b_z(m)$, for some $1 \neq z \in \mathbb{N}$, and $n = \sum_{i=0}^{b-1} n_i z^i$ and $m = \sum_{j=0}^{b'-1} m_j z^j$, where $n_i, m_j \in \{0, \ldots, z-1\}$. For later use, replacing $[m, n]$ by $[km, kn]$ where $k := \lfloor \frac{z}{n_{b-1}+1} \rfloor$, we may assume that $n_{b-1} \geq \lfloor \frac{z}{2} \rfloor$; this does not change b and increases b' by at most 1. We consider the leading bits of n and m in order to find the leading bit of the quotient: To this end, replacing n by $z^l n$ for some $l \in \{0, \ldots, b' - b\}$, we may assume that we have $b \leq b_z(m) \leq b + 1$, and that $q \in \{0, \ldots, z-1\}$ is the leading bit we are looking for. Now, letting $q' := \min\{\lfloor \frac{m_b z + m_{b-1}}{n_{b-1}} \rfloor, z - 1\}$, we have $q' - 2 \leq q \leq q'$:

To show that $q \leq q'$ we may assume that $q' = \lfloor \frac{m_b z + m_{b-1}}{n_{b-1}} \rfloor$. Then we have $n_{b-1} q' \geq m_b z + m_{b-1} - (n_{b-1} - 1)$. Hence $m - q'n \leq m - q' n_{b-1} z^{b-1} \leq m - (m_b z + m_{b-1}) z^{b-1} + (n_{b-1} - 1) z^{b-1} = (n_{b-1} - 1) z^{b-1} + \sum_{j=0}^{b-2} m_j z^j < n_{b-1} z^{b-1} \leq n$

implies that $q \leq q'$. Now assume that $q' \geq q+3$. Then from $q' \leq \frac{m}{n_{b-1}z^{b-1}} < \frac{m}{n-z^{b-1}}$ and $q = \lfloor \frac{m}{n} \rfloor > \frac{m}{n} - 1$ we get $3 \leq q'-q < \frac{m}{n-z^{b-1}} - (\frac{m}{n} - 1) = \frac{m \cdot z^{b-1}}{n(n-z^{b-1})} + 1$, thus $\frac{m}{n} > 2(n_{b-1}-1)$, and hence $z-4 \geq q'-3 \geq q = \lfloor \frac{m}{n} \rfloor \geq 2(n_{b-1}-1) \geq z-3$, a contradiction. Thus we have $q' \leq q+2$. $\sharp$

Having found the first bit q of the quotient, we replace m by $m - qz^l n$, and iterate. Hence we need at most $b'-b+1$ iteration steps. In each step, to compute q at most 3 trials are necessary, and the multiplication to compute $qz^l n$ needs $O(b)$ bit operations; note that the shift is not performed in practice. Moreover, since $m \geq qz^l n$ and $b' \leq b_z(qz^l n) + 1$, the subtraction $m - qz^l n$ also needs $O(b)$ bit operations only, instead of $O(b')$. This amounts to $O(b)$ bit operations per iteration step, and thus to a total of $O(b(b'-b)) = O(\ln(n)\ln(\frac{m}{n})) \subseteq O(\ln(n)\ln(m)) \subseteq O(\ln^2(m))$ bit operations, that is runs in quadratic time.

(1.6) Extended Euclidean algorithm. Let $m, n \in \mathbb{N}$. Then the greatest common divisor $r = \gcd(m,n) \in \mathbb{N}$, and **Bézout coefficients** $s, t \in \mathbb{Z}$ such that $r = sm + tn \in \mathbb{Z}$, can be computed as follows, without determining the factorisation of m and n. Leaving out the steps indicated by $\circ$, only needed to compute s and t, just yields the greatest common divisor r, the remaining algorithm being called the **Euclidean algorithm**:

```
• $r_0 \leftarrow m$
• $r_1 \leftarrow n$
∘ $s_0 \leftarrow 1$
∘ $t_0 \leftarrow 0$
∘ $s_1 \leftarrow 0$
∘ $t_1 \leftarrow 1$
• $i \leftarrow 1$
• while $r_i > 0$ do
    • $r_{i+1} \leftarrow r_{i-1} \bmod r_i$      # quotient and remainder $r_{i-1} = q_i r_i + r_{i+1}$
    ∘ $q_i \leftarrow \lfloor \frac{r_{i-1}}{r_i} \rfloor$
    ∘ $s_{i+1} \leftarrow s_{i-1} - q_i s_i$
    ∘ $t_{i+1} \leftarrow t_{i-1} - q_i t_i$
    • $i \leftarrow i+1$
• return $[r; s, t] \leftarrow [r_{i-1}; s_{i-1}, t_{i-1}]$      # resp. $r \leftarrow r_{i-1}$
```

We have $r_0 = s_0 m + t_0 n$ and $r_1 = s_1 m + t_1 n$, and by induction on $i \geq 1$ we have $r_{i+1} = r_{i-1} - q_i r_i = (s_{i-1}m + t_{i-1}n) - q_i \cdot (s_i m + t_i n) = s_{i+1}m + t_{i+1}n$. As we have $r_i < r_{i-1}$ for all $i \geq 1$, there is $l \in \mathbb{N}$ such that $r_l > 0$ and $r_{l+1} = 0$. Hence the algorithm terminates, after l executions of the while loop, returning $[r, s, t] := [r_l, s_l, t_l]$ such that $r = sa + tb$. Moreover, from $r_{i+1} = r_{i-1} - q_i r_i$, for all $i \in \{1, \ldots, l\}$, we indeed get $r = r_l = \gcd(r_l, 0) = \gcd(r_l, r_{l+1}) = \gcd(r_i, r_{i+1}) = \gcd(r_{i-1}, r_i) = \gcd(r_0, r_1) = \gcd(m, n)$.

Let $1 \neq z \in \mathbb{N}$. For $i \in \{1, \ldots, l\}$ we need $O(b_z(r_i)b_z(q_i))$ bit operations to compute $[q_i, r_{i+1}]$. Since $b_z(q_i) = 1 + \lfloor \log_z(q_i) \rfloor$, we have $O(\sum_{i=1}^{l} b_z(q_i)) = O(b_z(\prod_{i=1}^{l} q_i)) \subseteq O(b_z(r_0))$. Hence computing the quotients and remainders needs $O(\sum_{i=1}^{l} b_z(r_i)b_z(q_i)) \subseteq O(b_z(r_1) \cdot \sum_{i=1}^{l} b_z(q_i)) \subseteq O(b_z(r_1)b_z(r_0))$ bit operations. To compute the linear combinations needs $O(\sum_{i=1}^{l} b_z(q_i)b_z(s_i))$ bit operations, where in turn $b_z(s_i) \in O(b_z(s_{i-1})+b_z(q_{i-1}))$, hence we have $b_z(s_i) \in O(\sum_{j=1}^{i-1} b_z(q_j))$, yielding $O(\sum_{i=1}^{l} \sum_{j=1}^{i-1} b_z(q_i)b_z(q_j))$ bit operations. As above we from this obtain $O(\sum_{j=1}^{l-1} \sum_{i=j+1}^{l} b_z(q_j)b_z(q_i)) \subseteq O(\sum_{j=1}^{l-1} b_z(q_j)b_z(r_j)) \subseteq O(b_z(r_1) \cdot \sum_{j=1}^{l-1} b_z(q_j)) \subseteq O(b_z(r_1)b_z(r_0))$ bit operations. Thus this needs $O(b_z(r_1)b_z(r_0)) = O(b_z(m)b_z(n))$ bit operations; if $m \geq n$ this hence needs $O(\ln(m)\ln(n)) \subseteq O(\ln(m)^2)$ bit operations, that is runs in quadratic time.

Note that the number l of steps needed is not explicitly needed nor determined in the previous argument; actually it is in $O(\ln(n))$, see Exercise (6.6). For example, for $m := 126$ and $n := 35$ we get $r = \gcd(m, n) = 7 = 2m - 7n$:

i	q_i	r_i	s_i	t_i
0		126	1	0
1	3	35	0	1
2	1	21	1	-3
3	1	14	-1	4
4	2	7	2	-7
5		0	-5	18

(1.7) Polynomial arithmetic. Let R be a commutative ring and let $R[X]$ be the polynomial ring over R in the indeterminate X. For $0 \neq f = \sum_{i=0}^{d} f_i X^i \in R[X]$, where $f_i \in R$, having **leading coefficient** $\mathrm{lc}(f) := f_d \neq 0$, let $\deg(f) := d \in \mathbb{N}_0$ denote its **degree**.

The computational complexity of polynomial arithmetic is usually measured in ring operations in R, relative to the degrees of the polynomials involved. Hence in general this is not directly related to the number of bit operations needed, since coefficient growth in R has to be taken into account, for example for $R = \mathbb{Z}$, while it directly relates to the number of bit operations needed for finite rings R, for example for residue class rings $R = \mathbb{Z}/n\mathbb{Z}$ or for finite fields $R = \mathbb{F}_q$.

The algorithms for integer arithmetic straightforwardly generalize to polynomial arithmetic by letting $z := X$, and even have a tendency to become slightly easier, see Exercise (6.10):

Let $0 \neq f, g \in R[X]$, where $\deg(f) \geq \deg(g)$. Addition $f + g$ and subtraction $f - g$ need $O(\deg(f))$ ring operations, while multiplication $f \cdot g$, using the classical technique, needs $O(\deg(f)^2)$ ring operations. The Karatsuba algorithm generalizes to multiplication $f \cdot g$, needing $O(\deg(f)^{\log_2(3)})$ ring operations. Moreover, the best known polynomial multiplication algorithm again is the **Schönhage-Strassen Algorithm [1971]**, see [2, Ch.8.3], whenever $\deg(f) + \deg(g) \leq n$ needing $O(n \cdot \ln(n) \cdot \ln(\ln(n)))$ ring operations.

Let $0 \neq g \in R[X]$ such that its leading coefficient $\mathrm{lc}(g) \in R$ is a unit in R. Hence for $f \in R[X]$ there exist unique $q, r \in R[X]$ such that $r = 0$ or $\deg(r) < \deg(g)$, fulfilling $f = qg + r$. We may assume $\deg(f) \geq \deg(g)$, hence to compute $[q, r]$

needs $O(\deg(f)\cdot(\deg(f)-\deg(g))) \subseteq O(\deg(f)^2)$ ring operations in R; note that only $\mathrm{lc}(g) \in R^*$ has to be inverted, and that the quotient q can be computed without guessing. Finally, $R[X]$ is Euclidean if and only if R is a field; in this case the extended Euclidean algorithm generalizes to $0 \neq f, g \in R[X]$, and needs $O(\deg(f)\cdot\deg(g))$ ring operations.

2 Modular arithmetic

(2.1) Groups. a) A set G together with a **multiplication** $\cdot\colon G \times G \to G$ fulfilling the following conditions is called a **commutative group**: We have **commutativity** $ab = ba$ for all $a, b \in G$; we have **associativity** $(ab)c = a(bc)$ for all $a, b, c \in G$; there is a **neutral element** $1 \in G$ such that $a \cdot 1 = a$ for all $a \in G$; and for any $a \in G$ there is an **inverse** $a^{-1} \in G$ such that $a \cdot a^{-1} = 1$.

Then $1 \in G$ is the unique neutral element, and $a^{-1} \in G$ is the unique inverse of $a \in G$: Let $1' \in G$ also be a neutral element, then $1 = 1 \cdot 1' = 1'$; and letting $a' \in G$ also be an inverse of a, then $a' = a' \cdot 1 = a'aa^{-1} = 1 \cdot a^{-1} = a^{-1}$. Moreover, a subset $\emptyset \neq H \subseteq G$ is called a **subgroup**, if it is closed under products and taking inverses. Then we have $1 \in H$, and H again becomes a commutative group; we write $H \leq G$.

b) If G is finite, then the cardinality $|G| \in \mathbb{N}$ is called the **order** of G. In this case, the **order** of $a \in G$ is defined as $|a| := \min\{m \in \mathbb{N}; a^m = 1\} \in \mathbb{N}$; this indeed is well-defined: Since G is finite, there are $m, n \in \mathbb{Z}$, where $m > n$, such that $a^m = a^n$, hence we have $a^{m-n} = a^m \cdot (a^n)^{-1} = 1$ where $m - n \in \mathbb{N}$.

Moreover, letting $n := |a|$, we have $a^l = a^m$ for $l, m \in \mathbb{Z}$ if and only if $n \mid (l-m)$: If $n \mid (l-m)$ then there is $k \in \mathbb{Z}$ such that $l = m+kn$, hence $a^l = a^m(a^n)^k = a^n$. Conversely, if $a^l = a^m$, then by quotient and remainder writing $l - m = qn + r$, where $q \in \mathbb{Z}$ and $r \in \{0, \ldots, n-1\}$, yields $1 = a^{l-m} = a^r \cdot (a^n)^q = a^r$, thus by the minimality of n we conclude $r = 0$, that is $n \mid (l-m)$.

In particular, we have $a^m = 1$ if and only if $n \mid m$. This also shows that the **cyclic** subgroup $\langle a \rangle := \{a^m \in G; m \in \mathbb{Z}\} = \{a^m \in G; m \in \{0, \ldots, n-1\}\} \leq G$ has order $|\langle a \rangle| = n$. Hence it follows from Lagrange's Theorem below that $|a| \mid |G|$, thus in particular we have **Fermat's Theorem** saying that $a^{|G|} = 1 \in G$.

c) Let $H \leq G$ be a subgroup. Then we have **Lagrange's Theorem** saying that $|H| \mid |G|$: For $g \in G$ we consider the set $gH := \{gh \in G; h \in H\} \subseteq G$. We have $gh = gh'$, for $h, h' \in H$, if and only if $h = h'$. Thus we have $|gH| = |H|$. Moreover, for $g, g' \in G$ we have either $gH \cap g'H = \emptyset$ or $gH = g'H$: Let $gh = g'h'$, for $h, h' \in H$, then we have $g' = ghh'^{-1}$, and thus $g'h'' = ghh'^{-1}h''$ for all $h'' \in H$. Thus we have $g'H \subseteq gH$, and hence by symmetry $gH = g'H$. Letting $\mathcal{X} := \{gH \subseteq G; g \in G\}$, we thus have $G = \coprod_{X \in \mathcal{X}} X$, hence $|H| \mid |G|$.

(2.2) Modular arithmetic. a) For $n \in \mathbb{N}$ let $\mathbb{Z}_n := \{0, \ldots, n-1\} \subseteq \mathbb{Z}$, and for $a \in \mathbb{Z}$ let $\overline{a} := (a \bmod n) \in \mathbb{Z}_n$ be the remainder of a upon division by n. We define an addition $+\colon \mathbb{Z}_n \times \mathbb{Z}_n \to \mathbb{Z}_n$ and a multiplication $\cdot\colon \mathbb{Z}_n \times \mathbb{Z}_n \to \mathbb{Z}_n$ by

$\overline{a} + \overline{b} := \overline{a+b}$ and $\overline{a} \cdot \overline{b} := \overline{ab}$. This is well-defined, since we have independence from the choice of representatives: Let $a, a', b, b' \in \mathbb{Z}$ such that $\overline{a} = \overline{a'}$ and $\overline{b} = \overline{b'}$, that is there are $k, l \in \mathbb{Z}$ such that $a' = a + kn$ and $b' = b + ln$. Thus we have $a' + b' = (a + kn) + (b + ln) = (a + b) + (k + l)n$ and $a'b' = (a + kn)(b + ln) = ab + (al + bk + kln)n$, hence $\overline{a' + b'} = \overline{a + b}$ and $\overline{a'b'} = \overline{ab}$.

This shows that the laws of commutativity, associativity and distributivity are inherited from $\mathbb{Z}$; note that iterated arithmetic operations can be performed in $\mathbb{Z}$ before going over to remainders with respect to n, although this is usually not recommended. As for addition, we have a neutral element $0 \in \mathbb{Z}_n$, and any $a \in \mathbb{Z}_n$ has an additive inverse $\overline{-a} \in \mathbb{Z}_n$; hence $\mathbb{Z}_n$ is a commutative additive group. As for multiplication, we have a neutral element $1 \in \mathbb{Z}_n$, but a multiplicative inverse does not always exist; for example, $2 \in \mathbb{Z}_4$ does not have an inverse. Thus $\mathbb{Z}_n$ is a commutative ring, but not necessarily a field.

b) We consider the multiplicative structure of $\mathbb{Z}_n$: An element $a \in \mathbb{Z}_n$ is called **invertible**, if there is $b \in \mathbb{Z}_n$ such that $ab = 1 \in \mathbb{Z}_n$. Let $\mathbb{Z}_n^* := \{a \in \mathbb{Z}_n; a \text{ invertible}\}$, then $\mathbb{Z}_n^*$ is a commutative multiplicative group with neutral element $1 \in \mathbb{Z}_n^*$, called the **group of units** of $\mathbb{Z}_n$. Note that $\mathbb{Z}_1^* = \mathbb{Z}_1 = \{0\}$, while for $n \geq 2$ we have $0 \neq 1$ and $0 \notin \mathbb{Z}_n^*$. Thus $\mathbb{Z}_n$ is a field if and only if $\mathbb{Z}_n^* = \mathbb{Z}_n \setminus \{0\}$. Let $\varphi \colon \mathbb{N} \rightarrow \mathbb{N} \colon n \mapsto |\mathbb{Z}_n^*|$ be **Euler's totient function**, thus $\mathbb{Z}_n$ is a field if and only if $\varphi(n) = n - 1$.

In general, we have $\mathbb{Z}_n^* = \{\overline{a} \in \mathbb{Z}_n; \gcd(a, n) = 1\}$, being also called the **group of prime residues** modulo n: Note first that, whenever $a, a' \in \mathbb{Z}$ such that $\overline{a} = \overline{a'}$, we have $\gcd(a, n) = \gcd(a', n)$. If $a \in \mathbb{Z}_n^*$, having inverse $b \in \mathbb{Z}_n^*$, then $\overline{ab} = 1 \in \mathbb{Z}_n$ says that there is $k \in \mathbb{Z}$ such that $ab + kn = 1 \in \mathbb{Z}$, implying that $\gcd(a, n) = 1$. Conversely, if $a \in \mathbb{Z}_n$ such that $\gcd(a, n) = 1$, then there are Bézout coefficients $s, t \in \mathbb{Z}$ such that $sa + tn = 1 \in \mathbb{Z}$, hence we have $1 = \overline{sa + tn} = \overline{sa} \in \mathbb{Z}_n$, saying that $\overline{s} \in \mathbb{Z}_n$ is an inverse of $a \in \mathbb{Z}_n$. Note that hence inverses can be computed using the extended Euclidean algorithm, thus needing $O(\ln^2(n))$ bit operations.

From that we conclude that $\mathbb{Z}_n$ is a field if and only if $n \in \mathbb{N}$ is a prime: We have $\mathbb{Z}_1 = \{0\}$; if $n \geq 2$ is composite then there are $a, b \in \mathbb{Z}_n \setminus \{0, 1\}$ such that $n = ab \in \mathbb{Z}$, hence we have $\gcd(a, n) = a \neq 1$, that is $a \notin \mathbb{Z}_n^* \,\dot\cup\, \{0\}$; if n is a prime, then we have $\gcd(a, n) = 1$ for all $0 \neq a \in \mathbb{Z}_n$, hence $\mathbb{Z}_n^* = \mathbb{Z}_n \setminus \{0\}$.

c) We consider the additive structure of $\mathbb{Z}_n$: The additive group $\mathbb{Z}_n = \langle \overline{1} \rangle$ is cyclic of order n. We have $\mathbb{Z}_n = \coprod_{d \mid n} \{a \in \mathbb{Z}_n; \gcd(a, n) = d\}$, where in turn $\{a \in \mathbb{Z}_n; \gcd(a, n) = d\} = \{kd \in \mathbb{Z}_n; k \in \mathbb{Z}_{\frac{n}{d}}; \gcd(k, \frac{n}{d}) = 1\} = d\mathbb{Z}_{\frac{n}{d}}^*$. Moreover, given $a \in d\mathbb{Z}_{\frac{n}{d}}^*$, that is $\gcd(a, n) = d$, and $k \in \mathbb{Z}$, we have $\overline{ka} = 0 \in \mathbb{Z}_n$ if and only if $n \mid ka$, which holds if and only if $\frac{n}{d} \mid k$. Hence $d\mathbb{Z}_{\frac{n}{d}}^* \subseteq \mathbb{Z}_n$ is the subset of all elements having additive order $\frac{n}{d}$. Thus $\mathbb{Z}_n$ has precisely $\varphi(d)$ elements of additive order d, for all $d \mid n$, and hence we have $\sum_{d \mid n} \varphi(d) = n$. Moreover, from that we conclude that $\mathbb{Z}_n$ has precisely d elements of additive order dividing d, that is precisely one cyclic subgroup of order d, for all $d \mid n$.

d) Here is a couple of examples: If $n := p^e$ where $p \in \mathbb{N}$ is a prime and $e \in \mathbb{N}$, then we have $\mathbb{Z}_n \setminus \mathbb{Z}_n^* = \{a \in \mathbb{Z}_n; \gcd(a, n) > 1\} = \{a \in \mathbb{Z}_n; p \mid a\} = p\mathbb{Z}_{p^{e-1}}$, thus $\varphi(p^e) = |\mathbb{Z}_{p^e}^*| = p^e - p^{e-1} = p^{e-1}(p-1)$. Alternatively, in terms φ alone, we proceed by induction on e, where for $e = 1$ we have $\varphi(p) = p - 1$; letting $e \geq 2$, we get $\varphi(p^e) = p^e - \sum_{i=0}^{e-1} \varphi(p^i) = p^e - 1 - \sum_{i=1}^{e-1}(p^i - p^{i-1}) = p^e - p^{e-1}$.

If $n := pq$, where $p \neq q \in \mathbb{N}$ are primes, then we have $\mathbb{Z}_n \setminus \mathbb{Z}_n^* = \{a \in \mathbb{Z}_n; \gcd(a, n) > 1\} = \{a \in \mathbb{Z}_n; p \mid a\} \cup \{a \in \mathbb{Z}_n; q \mid a\} = p\mathbb{Z}_q \cup q\mathbb{Z}_p$, thus $\varphi(pq) = |\mathbb{Z}_{pq}^*| = n - (q + p - 1) = (p-1)(q-1)$. Alternatively, in terms φ alone, we have $\varphi(n) = n - \varphi(p) - \varphi(q) - \varphi(1) = pq - (p-1) - (q-1) - 1 = (p-1)(q-1)$.

(2.3) Modular exponentiation. Let $e, n \in \mathbb{N}$ and $m \in \mathbb{Z}_n = \{0, \ldots, n-1\}$. Then the modular e-th power $m^e \in \mathbb{Z}_n$ can be computed by **repeated squaring** as follows: To this end, let $e = \sum_{j=0}^{k-1} e_j 2^j$ be the binary representation of e, where $e_i \in \{0, 1\}$ and $k := b_2(e)$.

```
• $b_0 \leftarrow 1$
• $a_0 \leftarrow m$
• for $i \in [0, \ldots, k-1]$ do
    • $b_{i+1} \leftarrow b_i$
    • if $e_i = 1$ then
        • $b_{i+1} \leftarrow a_i b_{i+1} \bmod n$
    • $a_{i+1} \leftarrow a_i^2 \bmod n$
• return $b_k$
```

By induction on $i \in \{0, \ldots, k\}$ we have $a_i = m^{2^i} \in \mathbb{Z}_n$. Thus by induction on $i \in \{-1, \ldots, k-1\}$ we get $b_{i+1} = m^{\sum_{j=0}^{i} e_j 2^j} \in \mathbb{Z}_n$: The case $i = -1$ being clear, let $i \geq 0$; if $e_i = 0$ then $b_{i+1} = b_i = m^{\sum_{j=0}^{i-1} e_j 2^j} = m^{\sum_{j=0}^{i} e_j 2^j}$, if $e_i = 1$ then $b_{i+1} = a_i b_i = m^{2^i + \sum_{j=0}^{i-1} e_j 2^j} = m^{\sum_{j=0}^{i} e_j 2^j}$. This yields $b_k = m^{\sum_{j=0}^{k-1} e_j 2^j} = m^e$.

All integers occurring are bounded above by n^2, hence have bit lengths in $O(\ln(n))$, and the number of multiplications needed is in $O(\ln(e))$. Hence this needs $O(\ln(e)\ln^2(n))$ bit operations, that is runs in polynomial time. Note that 'classical' exponentiation by computing $m^e \in \mathbb{Z}$ first produces integers of bit lengths in $O(e \cdot \ln(n))$, and even by taking remainders modulo n needs $O(e)$ multiplications, hence the latter run in exponential time.

(2.4) Cryptosystems. A **cryptosystem** $[\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}]$ is a tuple, where the **plaintexts** $\mathcal{P}$, the **ciphertexts** $\mathcal{C}$ and the **keys** $\mathcal{K}$ are finite sets, and where $\mathcal{E} = \{E_e \colon \mathcal{P} \to \mathcal{C}; e \in \mathcal{K}\}$ and $\mathcal{D} = \{D_d \colon \mathcal{C} \to \mathcal{P}; d \in \mathcal{K}\}$ are **encryption** and **decryption functions**, respectively, such that for all $e \in \mathcal{K}$ there is some, but not necessarily unique, $d \in \mathcal{K}$ such that $D_d \circ E_e = \mathrm{id}_{\mathcal{P}}$.

To encrypt and decrypt plain texts, letters from the **Latin alphabet** are first **encoded** into and **decoded** from $\mathbb{Z}_{26} = \{0, \ldots, 25\}$, or electronic text data, using the **ASCII alphabet** of length 128, is encoded into and decoded from $\mathbb{Z}_{128} = \{0, \ldots, 127\}$. Hence a **word** of length $l \in \mathbb{N}$ is identified with an element in $\mathbb{Z}_z^l$, for some $z \geq 2$, and the latter is interpreted as the representation of a non-negative integer with respect to the base z. So, typically we have $\mathcal{P}, \mathcal{C} \subseteq \mathbb{Z}_z^l$.

The idea now is to keep information private to communication partners, Alice and Bob say, who communicate through an insecure channel, where data might be caught by an opponent, Oscar say. Hence plaintexts are first **encrypted** by Bob, then sent through the channel, and are **decrypted** again by Alice. Thus in practice, given the keys, encryption and decryption functions should be efficiently computable. Moreover, it should be difficult for Oscar to determine plaintexts from ciphertexts without knowing the keys used, and it should also be difficult for Oscar to determine the keys employed.

If given an encryption key $e \in \mathcal{K}$ a suitable decryption key $d \in \mathcal{K}$ can be assumed to be equal to e, or if d can be easily computed from e, then the cryptosystem is called **symmetric** or a **private-key cryptosystem**. In this case Alice and Bob first have to exchange the keys securely.

If a suitable decryption key $d \in \mathcal{K}$ cannot be computed easily from e, then the cryptosystem is called **asymmetric** and can be used as a **public-key cryptosystem**: To receive messages Alice publishes $e \in \mathcal{K}$, which Bob uses to encrypt messages, but Alice keeps the suitable decryption keys $d \in \mathcal{K}$ private. In this case no secure key exchange is necessary.

(2.5) The Rivest-Shamir-Adleman (RSA) cryptosystem [1978]. a) Let $p \neq q \in \mathbb{N}$ be primes and let $n := pq \in \mathbb{N}$ be the associated **modulus**. Let $\mathcal{P} = \mathcal{C} := \mathbb{Z}_n^*$ and $\mathcal{K} := \mathbb{Z}_{\varphi(n)}^*$, where $\varphi(n) = (p-1)(q-1)$, Moreover, for $e \in \mathbb{Z}_{\varphi(n)}^*$ let $E_e \colon \mathbb{Z}_n^* \to \mathbb{Z}_n^* \colon a \mapsto a^e$, and let $\mathcal{E} = \mathcal{D} := \{E_e \colon \mathbb{Z}_n^* \to \mathbb{Z}_n^*; e \in \mathbb{Z}_{\varphi(n)}^*\}$. This indeed is an (unsymmetric) cryptosystem:

Given $e \in \mathbb{Z}_{\varphi(n)}^*$, we have to provide $d \in \mathbb{Z}_{\varphi(n)}^*$ such that $E_d \circ E_e = \mathrm{id}_{\mathbb{Z}_n^*}$. In order to do so, let $d \in \mathbb{Z}_{\varphi(n)}^*$ such that $ed = 1 \in \mathbb{Z}_{\varphi(n)}^*$. Then we have $ed = 1 + k\varphi(n) \in \mathbb{Z}$, for some $k \in \mathbb{Z}$, and since by Fermat's Theorem we have $a^{\varphi(n)} = 1 \in \mathbb{Z}_n^*$, we get $(a^e)^d = a^{ed} = a \cdot (a^{\varphi(n)})^k = a \in \mathbb{Z}_n^*$, for all $a \in \mathbb{Z}_n^*$. Hence we may choose $[n, e]$ as **public key**, the **private key** being $[p, q, d]$.

b) This can be used in an **RSA block cipher**, where $l := \lfloor \log_{26}(n) \rfloor \in \mathbb{N}$ is the block length: Words of length l in the Latin alphabet are first encoded into $\mathbb{Z}_{26}^l$, and then via 26-adic expansion considered as elements of $\mathbb{Z}_{26^l} \subseteq \mathbb{Z}_n$; for example, for $l = 3$ the word '`abc`' yields $0 \cdot 26^2 + 1 \cdot 26 + 2 = 28 \in \mathbb{Z}_n$.

For example, for $p := 97$ and $q := 193$ we get $n = 18721$, and since $26^3 = 17576$ we let $l := 3$. Letting $e := 43$, using $\varphi(n) = (p-1)(q-1) = 96 \cdot 192 = 18432$, the extended Euclidean algorithm yields $1 = \gcd(e, \varphi(n)) = -8573 \cdot e + 20 \cdot \varphi(n)$, hence we let $d := \overline{-8573} = 9859 \in \mathbb{Z}_{\varphi(n)}^*$. Then the plaintext '`she has sensed a change in the weather`' is encrypted as shown in Table 1; note that we add trailing letters in order to obtain a plaintext whose length is a multiple of the block length. Note that letting $l := 1$ results in a protocol failure: By encrypting all the plaintext letters we may directly read off the plaintext from a ciphertext.

Table 1: RSA block cipher.

	$\mathbb{Z}_{26}$			$\mathcal{P} = \mathbb{Z}_{26^3}$	$\mathcal{C} = \mathbb{Z}_{18721}$
`she`	18	7	4	12354	13130
`has`	7	0	18	4750	95
`sen`	18	4	13	12285	7342
`sed`	18	4	3	12275	13805
`ach`	0	2	7	59	8347
`ang`	0	13	6	344	10022
`ein`	4	8	13	2925	5164
`the`	19	7	4	13030	13434
`wea`	22	4	0	14976	18716
`the`	19	7	4	13030	13434
`rzz`	17	25	25	12167	14498

(2.6) The RSA cryptosystem and integer factorisation. Let $p \neq q \in \mathbb{N}$ be primes and let $n := pq \in \mathbb{N}$. If $\varphi(n)$ is known, then inverses in $\mathbb{Z}^*_{\varphi(n)}$ can be computed in polynomial time, by the extended Euclidean algorithm.

Computing $\varphi(n) = (p-1)(q-1)$ is **polynomial time equivalent** to factoring $n = pq$: If p and q are known, then $\varphi(n) = (p-1)(q-1)$ is easily computed as well; conversely, if $\varphi(n) = (p-1)(q-1) = (p-1)(\frac{n}{p} - 1)$ is known, then $p^2 + (\varphi(n) - n - 1)p + n = 0$ shows that $\{p, q\}$ can be determined as the roots of a quadratic equation, namely $\{p, q\} = \{\frac{n+1-\varphi(n)}{2} \pm \frac{\sqrt{(n+1-\varphi(n))^2-4n}}{2}\}$. Hence p and q must be kept private. In particular, since for $0 \neq a \in \mathbb{Z}_n \setminus \mathbb{Z}_n^*$ we have $1 < \gcd(a, n) < n$, yielding a prime divisor of n, these elements have to be excluded; but since $\frac{n-1-\varphi(n)}{n} = \frac{p+q}{pq} = \frac{1}{p} + \frac{1}{q}$, they are extremely rare anyway.

Hence breaking the RSA cryptosystem **polynomial time reduces** to factoring n, thus the RSA cryptosystem is secure only if factoring $n = pq$ is computationally difficult. It is conjectured that factoring integers of the form pq is as difficult as factoring arbitrary integers, and that integer multiplication is a **(cryptographic) one-way function**, that is factoring integers cannot be done algorithmically in (randomised) polynomial time. Conversely it is conjectured that factoring $n = pq$ polynomial time reduces to breaking the RSA cryptosystem, implying that these problems are **polynomial time equivalent**.

Given the capabilities of the known factorisation algorithms, p and q should be chosen of the same size, which should be at least $p \sim 2^{512} \sim 10^{154}$, thus $n \sim 2^{1024} \sim 10^{308}$. The PKCS#1 standard currently in use employs 1024-bit moduli; the challenges RSA-200 and RSA-640, which are of size $n \sim 2^{663} \sim 10^{200}$ and $n \sim 2^{640} \sim 10^{192}$, respectively, have been successfully factored [2004, 2005].

Since factorisation algorithms might run faster for certain choices of the prime divisors of n, for example if $p-1$ or $p+1$ has no large prime divisors, then these have to be avoided as well; and $|p-q|$ must not be too small, since otherwise the prime divisors of n can be found by trial division with integers close to $\sqrt{n}$.

Finally, it can be shown that $|\mathcal{E}| = \frac{|\mathbb{Z}^*_{\varphi(n)}|}{\gcd(p-1,q-1)} = \frac{\varphi((p-1)(q-1))}{\gcd(p-1,q-1)}$, thus p and q have to be chosen such that $\gcd(p-1,q-1)$ is not too large. Actually, given an encryption key, there are precisely $\gcd(p-1,q-1)$ admissible decryption keys; in particular, since $2 \mid \gcd(p-1,q-1)$ in any case the latter are never unique.

3 Primality testing

(3.1) The Lucas test. a) Let $1 \neq n \in \mathbb{N}$. Then n is a prime if and only if there is a **primitive root** $a \in \mathbb{Z}_n^*$ such that $|a| = n-1$:

The number n is a prime if and only if $\varphi(n) = |\mathbb{Z}_n^*| = n-1$. Thus, if there is an element $a \in \mathbb{Z}_n^*$ such that $|a| = n-1$, then by Fermat's Theorem we have $n-1 \mid \varphi(n)$, hence $\varphi(n) = n-1$. Conversely, if n is a prime, then for all $a \in \mathbb{Z}_n^*$ we have $|a| \mid n-1$, but we have to show that there actually is some $a \in \mathbb{Z}_n^*$ such that $|a| = |\mathbb{Z}_n^*|$, that is to show **Artin's Theorem** saying that $\mathbb{Z}_n^*$ is cyclic:

Let $d \mid n-1 = |\mathbb{Z}_n^*|$, and let $a \in \mathbb{Z}_n^*$ such that $d = |a| = |\langle a \rangle|$. Then, by Fermat's Theorem, all elements of the cyclic group $\langle a \rangle$ have order dividing d. Now, $\mathbb{Z}_n^*$ being a field, the polynomial $X^d - 1 \in \mathbb{Z}_n[X]$ has at most d roots in $\mathbb{Z}_n$, implying that $\mathbb{Z}_n^*$ has at most d elements having order dividing d. Thus we conclude that $\langle a \rangle$ consists of all elements of $\mathbb{Z}_n^*$ of order dividing d. Since $\langle a \rangle$ has precisely $\varphi(d)$ elements of order d, this implies that $\mathbb{Z}_n^*$ has precisely $\varphi(d)$ elements of order d. Thus, assuming that $\mathbb{Z}_n^*$ does not have an element of order $n-1$, then we have $|\mathbb{Z}_n^*| \leq \sum_{d \mid n-1, d \neq n-1} \varphi(d) < n-1$, a contradiction. $\sharp$

b) To verify primality we have the **Lucas primality test [1876]**: Let $n-1 = \prod_{i=1}^r p_i^{e_i}$, where $r \in \mathbb{N}_0$, the $p_1, \ldots, p_r \in \mathbb{N}$ are pairwise distinct primes and $e_1, \ldots, e_r \in \mathbb{N}$. Then $a \in \mathbb{Z}_n^*$ has order $n-1$ if and only if $a^{n-1} = 1 \in \mathbb{Z}_n^*$ and $a^{\frac{n-1}{p_i}} \neq 1 \in \mathbb{Z}_n^*$, for all $i \in \{1, \ldots, r\}$: We only have to show that the power conditions imply that $|a| = n-1$. Indeed, the conditions imply that $|a| \mid n-1$; and assuming that $|a| < n-1$, we conclude that $|a|$ divides a maximal proper divisor of $n-1$, contradicting the conditions.

If n is a prime, then the tuple $[a; p_1, \ldots, p_r]$ is a called a **Lucas certificate** for n, where the primitive root $a \in \mathbb{Z}_n^*$ is called a **Lucas witness**. Note that to do this we need a factorisation algorithm to find factors of $n-1$, and to apply the Lucas test recursively to verify primality of the candidate prime factors of $n-1$ found. This yields a **Pratt certificate [1975]** for n, consisting of a Lucas certificate for n, together with Pratt certificates for the prime factors of $n-1$; here no certificate is necessary for the primes found by the Sieve of Erathostenes.

(3.2) Example: Fermat numbers. For $n \in \mathbb{N}_0$ let $F_n := 2^{2^n} + 1 \in \mathbb{N}$ be the n-th **Fermat number**, where $F_0 = 3$, $F_1 = 5$, $F_2 = 17$, $F_3 = 257$, $F_4 = 65537$ are primes. It was conjectured [Fermat, 1640] that F_n always is a prime. Actually, for $n \in \{1, \ldots, 4\}$ we have $3^{\frac{F_n-1}{2}} = 3^{2^{2^n-1}} = -1 \in \mathbb{Z}_{F_n}^*$ and thus $3^{F_n-1} = 3^{2^{2^n}} = 1 \in \mathbb{Z}_{F_n}^*$, implying that $3 \in \mathbb{Z}_{F_n}^*$ is a primitive root and $[3; 2]$ is a Lucas certificate for F_n. Note that from $2^{2^n} = -1 \in \mathbb{Z}_{F_n}^*$ we get $2^{2^{n+1}} = 1 \in \mathbb{Z}_{F_n}^*$, showing that $2 \in \mathbb{Z}_{F_n}^*$ has order 2^{n+1}, which always divides $F_n - 1 = 2^{2^n}$, and implies that $2 \in \mathbb{Z}_{F_n}^*$ is a primitive root if and only if $n \leq 1$, thus in these cases $[2; 2]$ is a Lucas certificate, but no statement is made elsewise.

But [Euler, 1732] has shown that $F_5 := 2^{2^5} + 1 = 4\,294\,967\,297$ has the prime divisor 641, hence is composite: We have $641 = 640 + 1 = 5 \cdot 2^7 + 1 \in \mathbb{Z}$, thus $\overline{5 \cdot 2^7} = -1 \in \mathbb{Z}_{641}$, and $641 = 625 + 16 = 5^4 + 2^4 \in \mathbb{Z}$, thus $\overline{2}^4 = -\overline{5}^4 \in \mathbb{Z}_{641}$, hence $\overline{F_5} = \overline{2}^{32} + 1 = \overline{2}^4 \overline{2}^{28} + 1 = -\overline{5 \cdot 2^7}^4 + 1 = -(-1)^4 + 1 = -1 + 1 = 0 \in \mathbb{Z}_{641}$. Similarly, [Landry, 1880] has shown that $F_6 := 2^{2^6} + 1 = 18\,446\,744\,073\,709\,551\,617$ has the prime divisor $274\,177$, hence is composite, too.

Actually, we have $3^{F_5-1} = 3^{2^{32}} = 3\,029\,026\,160 \neq 1 \in \mathbb{Z}_{F_5}^*$, saying that the order of $3 \in \mathbb{Z}_{F_5}^*$ does not divide $F_5 - 1$, implying that F_5 is composite without exhibiting a proper divisor. Similarly, $3^{F_6-1} = 3^{2^{64}} = 8\,752\,249\,535\,465\,629\,170 \neq 1 \in \mathbb{Z}_{F_6}^*$ implies that F_6 is composite as well. Using **Pepin's test [1877]**, proved using the **quadratic reciprocity law** for **Jacobi symbols**, saying that $3 \in \mathbb{Z}_{F_n}^*$ always is a Lucas primality or Fermat compositeness witness for F_n whenever $n \geq 1$, it has been shown that all F_n for $n \in \{5, \ldots, 32\}$ are composite, while it is still an open problem whether $\{F_0, \ldots, F_4\}$ are the only Fermat primes.

(3.3) The Fermat test. Let $1 \neq n \in \mathbb{N}$. To verify primality only with a certain probability we have the **Fermat compositeness test**: If n is a prime then for all $a \in \mathbb{Z}_n^*$ we have $a^{n-1} = 1$. Hence if there is $a \in \mathbb{Z}_n^*$ such that $a^{n-1} \neq 1 \in \mathbb{Z}_n^*$, then n is composite, and a is called a **Fermat witness** for n.

If n is composite, but still $a^{n-1} = 1 \in \mathbb{Z}_n^*$ for some $1 \neq a \in \mathbb{Z}_n^*$, then n is called a **Fermat pseudo-prime** with respect to the **base** a, and a is called a **Fermat liar** for n. If n is a Fermat pseudo-prime with respect to all bases $1 \neq a \in \mathbb{Z}_n^*$, then n is called a **Carmichael number** [Korselt, 1899; Carmichael, 1910]. We have $n^{\frac{2}{7}} \leq |\{k \in \{1, \ldots, n\}; k \text{ Carmichael number}\}| \leq n^{1-(1+\epsilon)\cdot\frac{\ln(\ln(\ln(n)))}{\ln(\ln(n))}}$, for $n \to \infty$ and for all $\epsilon > 0$ [Alford-Granville-Pomerance, 1992; Pomerance-Selfridge-Wagstaff, 1980], hence there are infinitely many Carmichael numbers; those $\leq 10^4$ are $[561, 1105, 1729, 2465, 2821, 6601, 8911]$.

The set $U_n := \{a \in \mathbb{Z}_n^*; a^{n-1} = 1\} \leq \mathbb{Z}_n^*$ is a subgroup, where we have $U_n = \mathbb{Z}_n^*$ if and only if n is either a prime or a Carmichael number. If $U_n \neq \mathbb{Z}_n^*$, then by Lagrange's Theorem we have $\frac{|U_n|}{|\mathbb{Z}_n^*|} \leq \frac{1}{2}$, implying that the fraction of Fermat liars is at most $\frac{1}{2}$; actually, we cannot do better in general, see Exercise (6.21). Since modular exponentiation needs $O(\ln(n)^3)$ bit operations, we have the following polynomial time **Monte-Carlo** algorithm to decide compositeness:

Given an error bound $0 < \epsilon \leq \frac{1}{2}$, for at least $\lceil -\log_2(\epsilon) \rceil$ randomly chosen elements of $\mathbb{Z}_n^*$ we perform the Fermat test; if a Fermat compositeness witness is found then composite is returned, otherwise (probably) prime (or Carmichael, to be precise) is returned. Thus, the answer composite is correct, while the answer prime is incorrect with an error probability of at most ϵ.

(3.4) The Miller-Rabin test [Miller, 1976; Rabin, 1980]. a) Let $1 \neq n \in \mathbb{N}$ be odd, and let $n-1 = 2^l m$, where $l \in \mathbb{N}$ and $m \in \mathbb{N}$ is odd. Moreover, let $O_n := \{a \in \mathbb{Z}_n^*; a^m = 1\} \leq \mathbb{Z}_n^*$ and $B_{n,k} := \{a \in \mathbb{Z}_n^*; a^{2^k m} = -1\} \subseteq \mathbb{Z}_n^*$, for $k \in \{0, \ldots, l-1\}$; note that for the cases $k \geq l$ we have $B_{n,k} = \emptyset$ anyway, see Exercise (6.23). These sets are mutually disjoint, and we have $B_n := O_n \,\dot\cup\, \coprod_{k=0}^{l-1} B_{n,k} \subseteq U_n := \{a \in \mathbb{Z}_n^*; a^{n-1} = 1\} \leq \mathbb{Z}_n^*$; note that $B_n^{-1} = B_n$.

Then, if n is a prime we have $B_n = \mathbb{Z}_n^*$: Let $a \in \mathbb{Z}_n^* \setminus O_n$, that is $a^m \neq 1$. Then, letting $\mathcal{J} := \{j \in \{0, \ldots, l-1\}; a^{2^j m} \neq 1\}$, we have $0 \in \mathcal{J}$, hence $\mathcal{J} \neq \emptyset$, thus we may let let $1 \neq b := a^{2^k m} \in \mathbb{Z}_n^*$, where $k := \max \mathcal{J}$. Hence if $k < l-1$ we get $b^2 = a^{2^{k+1} m} = 1$, while if $k = l-1$ we get $b^2 = a^{2^l m} = a^{n-1} = 1$ anyway, thus in any case b is a root of $X^2 - 1 \in \mathbb{Z}_n[X]$, which since $\mathbb{Z}_n$ is a field implies that $b = -1$, hence $a \in B_{n,k}$.

To the contrary, if $n \neq 9$ is composite, then by (3.6) below we have $\frac{|B_n|}{|\mathbb{Z}_n^*|} \leq \frac{1}{4}$; note that for $n = 9$ we have $l = 3$ and $m = 1$, hence $O_9 = \{1\}$, as well as $B_{9,0} = \{-1\}$ and $B_{9,1} = \{a \in \mathbb{Z}_9^*; a^2 = -1\} = \emptyset$ and $B_{9,2} = \{a \in \mathbb{Z}_9^*; a^4 = -1\} = \emptyset$, thus $B_9 = \{\pm 1\}$, while $\varphi(9) = 6$. An element $a \in \mathbb{Z}_n^* \setminus B_n$ is called a **strong compositeness witness** for n; note that these for composite n always exist, thus there are no 'strong Carmichael numbers'. Since $l \in O(\ln(n))$ and modular exponentiation needs $O(\ln(n)^3)$ bit operations, a strong compositeness test needs $O(\ln(n)^4)$ bit operations, yielding a polynomial time Monte-Carlo algorithm to decide compositeness, which actually is the workhorse of modern primality (or better compositeness) testing.

If n is composite and $a \in B_n$, then n is called a **strong pseudo-prime** with respect to the base a, and a is called a **strong liar** for n; in this case, from $B_n \subseteq U_n$ we conclude that a also is a Fermat liar. Although there are composite $n \neq 9$ having a fraction of $\frac{1}{4}$ strong liars, for most n this fraction is much smaller, see Exercise (6.22). Moreover, although it is possible to construct strong pseudo-primes for any given finite set of bases [Arnault, 1995], these are extremely rare: In view of the considerations below, if n is a strong pseudoprime with respect to $a, a' \in \mathbb{Z}_n^*$, then it is likely that it is also with respect to $aa' \in \mathbb{Z}_n^*$. Hence we are tempted to only consider bases $\overline{a} \in \mathbb{Z}_n^*$ such that $a \in \mathbb{N}$ can be chosen to be prime. For example, letting $\{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, \ldots\}$ be the smallest primes, then the smallest strong pseudo-prime n with respect to the first $t \in \mathbb{N}$ smallest primes, where $t \in \{1, \ldots, 11\}$, is given in Table 2.

b) If n is composite such that $l \geq 2$, then in general $B_n \subseteq U_n$ is not a subgroup, see Exercise (6.23): More precisely, we have $O_n \leq U_n$ and $O_n B_{n,k} = B_{n,k}$ for all $k \in \{0, \ldots, l-1\}$, and for $k' \in \{0, \ldots, l-1\}$ such that $k' < k$ we have $B_{n,k'} B_{n,k} \subseteq B_{n,k}$. Moreover, we have $B_{n,0} B_{n,0} \subseteq O_n$, implying that for $l = 1$ we indeed have $B_n \leq U_n$. But for $k \geq 1$ we might have $B_{n,k} B_{n,k} \not\subseteq O_n \,\dot\cup$

Table 2: Strong pseudo-primes.

t	n	factorisation	$\sim$
1	2047	$23 \cdot 89$	$2 \cdot 10^3$
2	1373653	$829 \cdot 1657$	$1 \cdot 10^6$
3	25326001	$2251 \cdot 11251$	$3 \cdot 10^7$
4	3215031751	$151 \cdot 751 \cdot 28351$	$3 \cdot 10^9$
5	2152302898747	$6763 \cdot 10627 \cdot 29947$	$2 \cdot 10^{12}$
6	3474749660383	$1303 \cdot 16927 \cdot 157543$	$3 \cdot 10^{12}$
$7, 8$	341550071728321	$10670053 \cdot 32010157$	$3 \cdot 10^{14}$
$9, 10, 11$	3825123056546413051	$149491 \cdot 747451 \cdot 34233211$	$4 \cdot 10^{18}$

$\coprod_{i=0}^{k-1} B_{n,i}$, thus $B_{n,k}B_{n,k} \not\subseteq B_n$.

Actually, elements violating the closure property with respect to products lead to a factorisation of n: Let $a, b \in B_{n,k}$ for some $k \geq 1$, such that $ab \not\in B_n$. Then let $j := \min\{i \in \mathbb{N}_0; (ab)^{2^i m} = 1\}$. Thus from $(ab)^{2^k m} = 1$ and $ab \not\in O_n$ we infer $j \in \{1, \ldots, k\}$. Thus letting $x := (ab)^{2^{j-1}m} \in \mathbb{Z}_n^*$ we get $x^2 = 1$, where $x \neq 1$, and from $ab \not\in B_{n,j-1}$ we get $x \neq -1$. Hence we have $(x+1)(x-1) = x^2 - 1 = 0 \in \mathbb{Z}_n$, while $x \pm 1 \neq 0 \in \mathbb{Z}_n$, thus $1 < \gcd(x \pm 1, n) < n$; note that this is just a special case of the Fermat-Legendre factorisation method, see (5.1).

We now proceed to prove the Miller-Rabin criterion:

(3.5) Proposition. Let $p \in \mathbb{N}$ be an odd prime, and let $e \in \mathbb{N}$. Then $\mathbb{Z}_{p^e}^*$ is a cyclic group of order $p^{e-1}(p-1)$.

Proof. By Artin's Theorem, $\mathbb{Z}_p^*$ is a cyclic group of order $p-1$, hence using the natural map $\mathbb{Z}_{p^e} \rightarrow \mathbb{Z}_p \colon a \pmod{p^e} \mapsto a \pmod{p}$, which induces a homomorphism of groups $\mathbb{Z}_{p^e}^* \rightarrow \mathbb{Z}_p^*$, there is an element of $\mathbb{Z}_{p^e}^*$ having order divisible by $p-1$, thus there is $a \in \mathbb{Z}_{p^e}^*$ having order $p-1$. If there is $b \in \mathbb{Z}_{p^e}^*$ having order p^{e-1}, then $ab \in \mathbb{Z}_{p^e}^*$ has order $p^{e-1}(p-1) = |\mathbb{Z}_{p^e}^*|$, implying that $\mathbb{Z}_{p^e}^*$ is cyclic. Thus, it suffices to show by induction on $e \in \mathbb{N}$ that $(1+p)^{p^{e-1}} = 1 + k_e p^e \in \mathbb{Z}$, where $p \not| \, k_e \in \mathbb{N}$, implying that $b := 1 + p \in \mathbb{Z}_{p^e}^*$ has order p^{e-1}:

For $e = 1$ we have $k_e = 1$, while for $e \geq 2$ we have $(1+p)^{p^{e-1}} = (1+k_{e-1}p^{e-1})^p = 1 + k_{e-1}p^e + \frac{p(p-1)}{2}k_{e-1}^2 p^{2(e-1)} + \sum_{i=3}^p \binom{p}{i} k_{e-1}^i p^{i(e-1)}$. We have $2(e-1)+1 \geq e+1$ if and only if $e \geq 2$, and $i(e-1) \geq e+1$ if and only if $e \geq \frac{i+1}{i-1} = 1 + \frac{2}{i-1}$, which for $e \geq 2$ and $i \geq 3$ is fulfilled. Thus we have $(1+p)^{p^{e-1}} = 1 + k_{e-1}p^e + k'_{e-1}p^{e+1}$, for some $k'_{e-1} \in \mathbb{Z}$, hence $p \nmid k_e := k_{e-1} + pk'_{e-1}$. ♯

(3.6) Theorem: [Miller, 1976; Rabin, 1980]. Let $9 \neq n \in \mathbb{N}$ be odd and composite. Keeping the notation of (3.4), we have $|B_n| \leq \frac{1}{4}\varphi(n)$.

Proof. Let $n = \prod_{i=1}^r p_i^{e_i}$, where the $p_i \in \mathbb{N}$ are pairwise distinct odd primes, $e_i \in \mathbb{N}$ and $\sum_{i=1}^r e_i \geq 2$. By the **Chinese remainder theorem**, see Exercise (6.12), the natural map $\mathbb{Z}_n \rightarrow \prod_{i=1}^r \mathbb{Z}_{p_i^{e_i}} \colon a \pmod{n} \mapsto [a \pmod{p_i^{e_i}}; i \in \{1, \ldots, r\}]$ is an isomorphism of rings. This induces an isomorphism of groups $\mathbb{Z}_n^* \rightarrow \prod_{i=1}^r \mathbb{Z}_{p_i^{e_i}}^*$. In particular, we have $\varphi(n) = \prod_{i=1}^r \varphi(p_i^{e_i}) = \prod_{i=1}^r p_i^{e_i-1}(p_i - 1) = \prod_{i=1}^r 2^{l_i} m_i$, where $\varphi(p_i^{e_i}) = 2^{l_i} m_i$, and in turn $l_i \in \mathbb{N}$ and $m_i \in \mathbb{N}$ odd. Hence we get $\gcd(n-1, \varphi(p_i^{e_i})) = 2^{\min\{l, l_i\}} m_i'$, where still $n - 1 = 2^l m$ and $m_i' := \gcd(m, m_i)$. Finally, let $l' := \min\{l_1, \ldots, l_r\}$.

We have $O_n \cong \prod_{i=1}^r \{a \in \mathbb{Z}_{p_i^{e_i}}^*; a^m = 1\}$ as commutative groups. Since $\mathbb{Z}_{p_i^{e_i}}^*$ is a cyclic group of order $\varphi(p^{e_i})$, and thus for any $d \mid \varphi(p^{e_i})$ has precisely d elements of order dividing d, we conclude that $|O_n| = \prod_{i=1}^r \gcd(m, \varphi(p_i^{e_i})) = \prod_{i=1}^r m_i'$.

We have $B_{n,k} \cong \prod_{i=1}^r \{a \in \mathbb{Z}_{p_i^{e_i}}^*; a^{2^k m} = -1\}$ as sets, thus for $k \geq l'$ we have $B_{n,k} = \emptyset$. For $k < l'$ we have $|\{a \in \mathbb{Z}_{p_i^{e_i}}^*; a^{2^k m} = -1\}| = |\{a \in \mathbb{Z}_{p_i^{e_i}}^*; a^{2^{k+1} m} = 1\}| - |\{a \in \mathbb{Z}_{p_i^{e_i}}^*; a^{2^k m} = 1\}| = \gcd(2^{k+1}m, \varphi(p_i^{e_i})) - \gcd(2^k m, \varphi(p_i^{e_i})) = (2^{k+1} - 2^k) \gcd(m, m_i) = 2^k m_i'$, for $i \in \{1, \ldots, r\}$, implying $|B_{n,k}| = 2^{kr} \cdot \prod_{i=1}^r m_i'$.

Hence we get $|B_n| = \prod_{i=1}^r m_i' + \sum_{k=0}^{l'-1}(2^{kr} \cdot \prod_{i=1}^r m_i') = (1 + \frac{2^{rl'}-1}{2^r-1}) \cdot \prod_{i=1}^r m_i' = \alpha \cdot \beta \cdot \varphi(n)$, where $\alpha := \frac{2^{rl'}+2^r-2}{(2^r-1) \cdot 2^{\sum_{i=1}^r l_i}}$ and $\beta := \prod_{i=1}^r \frac{m_i'}{m_i} \leq 1$. Since $\alpha \leq \frac{1}{2^r-1} \cdot (1 + \frac{2^r-2}{2^{rl'}}) \leq \frac{1}{2^r-1} \cdot (1 + \frac{2^r-2}{2^r}) = \frac{2(2^r-1)}{2^r(2^r-1)} = \frac{1}{2^{r-1}}$, we are done for $r \geq 3$. Similarly, if $r = 2$ and $[m_1, m_2] \neq [m_1', m_2']$, then we have $\alpha \leq \frac{1}{2}$ and $\beta \leq \frac{1}{2}$.

Hence let $r = 2$ and $m_i = m_i'$. Then we have $p_i^{e_i-1} \mid m_i = m_i' \mid m \mid n - 1$, and from $p_i \mid n$ we conclude $e_i = 1$, and thus $n = p_1 p_2$. Since $e_i = 1$ we have $p_i - 1 = 2^{l_i} m_i$, and hence $0 = \overline{n-1} = \overline{p_1 p_2 - 1} = \overline{p_j - 1} = \overline{2}^{l_j} \cdot \overline{m_j} \in \mathbb{Z}_{m_i}$, for $j \neq i$. This implies $m_i \mid m_j$, and hence we have $m_1 = m_2$ and thus $l_1 \neq l_2$. From this we obtain $\alpha \leq \frac{1}{2^2-1} \cdot \frac{2^{2l'}+2^2-2}{2^{l_1+l_2}} \leq \frac{1}{3}(\frac{2^{2l'}}{2^{2l'+1}} + \frac{2}{2^3}) = \frac{1}{3}(\frac{1}{2} + \frac{1}{4}) = \frac{1}{4}$.

Finally, for $r = 1$ and $e_1 > 1$, since $\mathbb{Z}_{p_1^{e_1}}^*$ is cyclic, we get $|B_n| \leq |U_n| = \gcd(\varphi(n), n-1) = \gcd(p_1^{e_1-1}(p_1 - 1), p_1^{e_1} - 1) = p_1 - 1$, thus since $n \neq 9$ we have $\frac{|B_n|}{\varphi(n)} \leq \frac{p_1-1}{p_1^{e_1-1}(p_1-1)} = \frac{1}{p_1^{e_1-1}} \leq \frac{1}{4}$. ♯

(3.7) Other primality and compositeness tests. The **Solovay-Strassen test [1977]**, based on **Euler's criterion** for being a modular square and using Jacobi symbols, also is a polynomial time Monte-Carlo algorithm to prove compositeness, the associated liars being called **Euler liars**. But since it is more expensive and has more liars, it is superseded by the Rabin-Miller test.

Adleman-Huang [1992] have given a polynomial time Monte-Carlo algorithm to decide primality; this algorithm uses **hyperelliptic curves** of genus 2 and is impractical. For practical purposes, the **elliptic curve primality proving (ECPP)** algorithm [Atkin-Morain, 1990] is used, which is based on the impractical **Goldwasser-Kilian test [1986]**, and needs expected polynomial time, but in the worst case might be much slower. The largest integers proven to be prime have size $\sim 10^{1000}$, but are of a special shape.

The simple deterministic primality test based on **Wilson's Theorem**, see Exercise (6.16), runs in exponential time $O(n\ln^2(n))$. Much better, the **Jacobi sum test** [Adleman-Pomerance-Rumely, 1983] to deterministically decide primality runs in time $O(\ln^{c\ln(\ln(\ln(n)))}(n))$, for some $c > 0$, which is close to polynomial time. Finally, **Agrawal-Kayal-Saxena [2002]** have given an astonishingly simple polynomial time algorithm to decide primality; this is as yet impractical, but it shows that primality can be deterministically decided in polynomial time.

4 Integer factorisation

(4.1) Trial division. a) The straightforward approach to compute the factorisation of $1 \neq n \in \mathbb{N}$ is to use **trial division** with respect to $\mathcal{P}_{\leq\sqrt{n}}$; since any proper prime divisor of n is contained in $\mathcal{P}_{\leq\sqrt{n}}$, this in particular proves primality or compositeness:

```
• $L \leftarrow []$
• for $p \in \mathcal{P}_{\leq\sqrt{n}}$ do
    • while $(n \bmod p) = 0$ do
        • append($L$,$p$)
        • $n \leftarrow \frac{n}{p}$
• if $n = 1$ then
    • return $L$      # $n$ decomposable, factorisation found
• else
    • return $[n]$      # $n$ is a prime
```

By the Prime Number Theorem, the number of trials is given as $\pi(\sqrt{n}) \sim \frac{\sqrt{n}}{\ln(\sqrt{n})}$, each needing $\ln(n)$ quotient and remainder steps, each in turn using $O(\ln(\sqrt{n})\ln(n))$ bit operations. Hence trial division is an exponential time algorithm which runs time $O(\sqrt{n}\cdot\ln^2(n)) \subseteq O(\exp((\frac{1}{2}+\epsilon)\ln(n)))$, for all $\epsilon > 0$. Although there are better integer factorisation algorithms, as we will see below, and much better primality testing or compositeness testing algorithms, as we have seen above, trial division is still used in practice to treat small n or to discard small prime divisors.

b) The set $\mathcal{P}_{\leq n}$, for $1 \neq n \in \mathbb{N}$, is computed by the **Sieve of Erathostenes**:

```
• $L \leftarrow [2, \ldots, n]$
• $k \leftarrow 1$
• while $k < n$ do
    • $k \leftarrow k + 1$
    • if $k$ in $L$ then
        • $j \leftarrow 2k$
    • while $j \leq n$ do
        • delete($L$,j)
        • $j \leftarrow j + k$
• return $L$
```

In practice, this is run only once for some fixed n, and the set $\mathcal{P}_{\leq n}$ is stored; a typical choice is $n := 10^6$, where $\pi(n) = 78498$. Note that to save space only the differences between the successive elements of $\mathcal{P}_{\leq n}$ are stored.

(4.2) The ρ-method [Pollard, 1975]. Let $1 \neq n \in \mathbb{N}$. Letting $x_0 \in \mathbb{Z}_n$, and given a function $f \colon \mathbb{Z}_n \to \mathbb{Z}_n$, let recursively $x_i := f(x_{i-1})$, for $i \in \mathbb{N}$. We assume that f is chosen such that it behaves as a uniformly distributed random variable. In practice, in order to minimise the number of operations needed and since linear functions do not fulfill the randomness assumption, functions $f_c \colon x \mapsto x^2 + c$, for some $c \in \mathbb{Z}_n \setminus \{0, \pm 2\}$, are used, where a typical choice is $c := 1$; note that $f_0(x) = x^2$, and $f_{\pm 2}(x \mp x^{-1}) = x^2 + x^{-2}$, for $x \in \mathbb{Z}_n^*$, do not fulfill the randomness assumption either. Actually it is an open problem whether there are random functions in the above sense at all.

Let $1 \neq p \in \mathbb{N}$ be a divisor of n. Then there are $k \in \mathbb{N}_0$ and $l \in \mathbb{N}$ minimal such that we have a **collision** $x_k = x_{k+l} \in \mathbb{Z}_p$, implying that $p \mid \gcd(x_k - x_{k+l}, n)$. Note that we have $x_k = x_{k+jl} \in \mathbb{Z}_p$, for all $j \in \mathbb{N}_0$, which is the name-giving property of the method. To find a collision, it can be avoided to store the sequence of x_i's by using **Floyd's cycle detection trick**: Let $y_0 := x_0 \in \mathbb{Z}_n$ and $y_i := f(f(y_{i-1})) = x_{2i} \in \mathbb{Z}_n$, for $i \in \mathbb{N}$. Then we have $x_i = y_i \in \mathbb{Z}_p$ if and only if $i \geq k$ and $l \mid 2i - i = i$. Thus the minimal $i \in \mathbb{N}$ fulfilling these conditions is an element of $\{k, \ldots, k + l\}$, hence we still need at most $k + l$ steps to arrive at a collision.

Now let $q := \frac{n}{p} \in \mathbb{N}$, and assume that $q \neq 1$ and $\gcd(p, q) = 1$; note that p and q exist if and only if n is not a prime power, see also Exercise (6.20). Hence by the **Chinese remainder theorem**, see Exercise (6.12), the natural map $\mathbb{Z}_n = \mathbb{Z}_{pq} \to \mathbb{Z}_p \times \mathbb{Z}_q \colon a \pmod{n} \mapsto [a \pmod{p}, a \pmod{q}]$ is a bijection. Thus we infer that the images of x_i in $\mathbb{Z}_p$ and $\mathbb{Z}_q$, respectively, can be considered as independent random variables. Hence with probability $\frac{q-1}{q}$ we have $q \not| \gcd(x_k - x_{k+l}, n)$, implying that $1 < \gcd(x_k - x_{k+l}, n) < n$, thus yielding a proper divisor of n. This yields the following **Las-Vegas algorithm**, where $t \in \mathbb{N}$ is the maximum number of tries made:

```
• choose $x \in \mathbb{Z}_n$ randomly
• $y \leftarrow x$
• $i \leftarrow 0$
• while $i < t$ do
    • $i \leftarrow i + 1$
    • $x \leftarrow f(x)$
    • $y \leftarrow f(f(y))$
    • $g \leftarrow \gcd(x - y, n)$
    • if $1 < g < n$ then
    • return $g$        # proper divisor found
• return fail       # no proper divisor found
```

Each execution of the loop runs in quadratic time. We are going to show that the number t of tries needed to arrive at a collision with a given probability $1-\epsilon$, where $0<\epsilon<1$, is in $O(\sqrt{p}) \subseteq O(\sqrt[4]{n})$, yielding exponential running time $O(\exp(\frac{1}{4}\ln(n))\cdot\ln^2(n))$, but actually depends on the smallest prime divisor p of n, and thus is in $O(\exp(\frac{1}{2}\ln(p))\cdot\ln^2(n))$:

For $t \in \mathbb{N}_0$ such that $t \leq p$, precisely $\prod_{i=0}^{t}(p-i)$ of the p^{t+1} sequences in $\mathbb{Z}_p^{t+1}$ have pairwise distinct entries. By **Taylor series expansion**, for $0 \leq \lambda \leq 1$ we have $0 \leq \exp(-\lambda)-(1-\lambda) \leq \frac{\lambda^2}{2}$, hence for the fraction of the sequences with pairwise distinct entries amongst all sequences we get $\prod_{i=0}^{t}(1-\frac{i}{p}) \leq \prod_{i=0}^{t}\exp(-\frac{i}{p}) = \exp(\frac{-t(t+1)}{2p}) \leq \exp(\frac{-t^2}{2p})$, where $\exp(\frac{-t^2}{2p}) < \epsilon$ if and only if $t > \sqrt{-2p\cdot\ln(\epsilon)} \in O(\sqrt{p})$. Note that for $\epsilon = \frac{1}{2}$ and $p = 365$ this yields $t \geq 23$, being called the **birthday paradox**. $\sharp$

For example, let $n := 7429 = 17\cdot 19\cdot 23$. Then using $x \mapsto x^2+1$ we get:

i	x	y	g
0	1	1	
1	2	5	1
2	5	677	1
3	26	2957	1
4	677	6890	19

(4.3) The $(p-1)$-method [Pollard, 1974]. We introduce a new concept, which will play a key role in the more efficient factorisation methods to be desribed below: An integer $n \in \mathbb{N}$ is called ***b*-smooth**, for some $b \in \mathbb{N}$, if it has the factorisation $n = \prod_{q\in\mathcal{P}_{\leq b}} q^{e_q(n)}$, where $e_q(n) \in \mathbb{N}_0$, that is the prime divisors of n do not exceed b.

Now let $1 \neq n \in \mathbb{N}$, let $b \leq n$, and let $p \in \mathbb{N}$ be a prime divisor of n such that $p-1$ is b-smooth. Then, for $x \in \mathbb{Z}_n^*$ and $e \in \mathbb{N}$ such that $p-1 \mid e$ we have $x^e = 1 \in \mathbb{Z}_p^*$, thus $\gcd(x^e-1,n) > 1$. To find a suitable exponent e, we observe that by smoothness $e_q(p-1) \neq 0$ only for $q \in \mathcal{P}_{\leq b}$, and in this case $e_q(p-1) \leq \log_q(n)$, hence we may choose $e := \prod_{q\in\mathcal{P}_{\leq b}} q^{\lfloor \log_q(n)\rfloor} \in \mathbb{N}$. Note that in practice b is chosen small enough so that $\mathcal{P}_{\leq b}$ can be determined explicitly, but this can be avoided by observing that $e = \operatorname{lcm}\{m^{\lfloor \log_m(n)\rfloor} \in \mathbb{N}; m \in \{2,\ldots,b\}\}$.

Letting $n = \prod_{q\in\mathcal{P}} q^{e_q(n)}$, by the **Chinese remainder theorem**, see Exercise (6.12), the natural map $\mathbb{Z}_n \to \prod_{q\in\mathcal{P},e_q(n)>0} \mathbb{Z}_{q^{e_q(n)}}$ is an isomorphism of rings, inducing an isomorphism of groups $\mathbb{Z}_n^* \to \prod_{q\in\mathcal{P},e_q(n)>0} \mathbb{Z}_{q^{e_q(n)}}^*$, where by (3.5) $\mathbb{Z}_{q^d}^*$ is a cyclic group of order $q^{d-1}(q-1)$, where $d \in \mathbb{N}$. This shows that if n additionally has a prime divisor $q \neq p$ such that $q-1$ is not b-smooth, then it is likely that $x^e \neq 1 \in \mathbb{Z}_q^*$, implying that $\gcd(x^e-1,n) < n$. Assuming again that n is not a prime power, see also Exercise (6.20), this yields the following Las-Vegas algorithm to find a proper divisor of n:

```
• choose $x \in \mathbb{Z}_n^*$ randomly
• for $q \in \mathcal{P}_{\leq b}$ do
    • $e \leftarrow q$
    • while $e \leq n$ do
        • $e \leftarrow e \cdot q$
    • $e \leftarrow \frac{e}{q}$
    • $x \leftarrow x^e \bmod n$
• $g \leftarrow \gcd(x-1, n)$
• if $1 < g < n$ then
    • return $g$
• else
    • return fail
```

In each execution of the loop, computing q^e needs $O(\ln^2(q)\log_q^2(n)) = O(\ln^2(n))$ bit operations, and modular exponentiation needs $O(\ln^3(n))$ bit operations. By the Prime Number Theorem, the loop is executed $\sim \frac{b}{\ln(b)}$ times, hence this runs in exponential time $O(\frac{b}{\ln(b)} \cdot \ln^3(n))$. Thus b must be chosen small indeed; in practice, a typical choice is $b \leq 10^6$. But note that this limits usability due to the strong assumptions on the prime divisors of n.

For example, let again $n := 7429 = 17 \cdot 19 \cdot 23$, where $17 - 1 = 2^4$ and $19 - 1 = 2 \cdot 3^2$ and $23 - 1 = 2 \cdot 11$. We choose $x := 2$. Then for $b := 2$ we get $e = 2^{12} = 4096$ and $x^e = 4064 \in \mathbb{Z}_n$, thus $g = \gcd(4063, 7429) = 17$; for $b := 3$ we get $e = 2^{12} \cdot 3^8 = 4096 \cdot 6561 = 26\,873\,856$ and $x^e = 1616 \in \mathbb{Z}_n$, thus $g = \gcd(1615, 7429) = 323 = 17 \cdot 19$.

(4.4) Example: Fermat numbers. We apply the ρ-method and the $(p-1)$-method to a few Fermat numbers; for the former, which in the considered cases always works, we use the iteration $x \mapsto x^2 + 1$ and the initial value $x := 1$, while the parameters for the latter in the successful cases are given below. Moreover, to exhibit Pratt certificates, we take the primes up to 10^6 for granted.

i) For the Fermat number $F_5 := 2^{2^5} + 1 = 4\,294\,967\,297 \sim 4 \cdot 10^9$ the ρ-method needs 22 tries to find the divisor $p_3 := 641$, hence we get $p_7 := \frac{F_5}{p_3} = 6\,700\,417$ [Euler, 1732]; the $(p-1)$-method finds p_3 with $b \leq 3$ and $x := 5$, and $b := 5$ and $x := 3$. From $p_3 - 1 = 2^7 \cdot 5$ we find the Lucas witness 3 for p_3; from $p_7 - 1 = 2^7 \cdot 3 \cdot 17449$ we find the Lucas witness 5 for p_7.

ii) For the Fermat number $F_6 := 2^{2^6} + 1 \sim 2 \cdot 10^{19}$, the ρ-method needs 808 tries to find the divisor $p_6 := 274177$, hence we get $p_{14} := \frac{F_6}{p_6} = 67\,280\,421\,310\,721$ [Landry, 1880]; the $(p-1)$-method finds p_6 with $3 \leq b \leq 5$ and $x := 479$, and $7 \leq b \leq 13$ and $x := 67$, and $b := 17$ and $x := 3$. From $p_6 - 1 = 2^8 \cdot 3^2 \cdot 7 \cdot 17$ we find the Lucas witness 5 for p_6; and from $p_{14} - 1 = 2^8 \cdot 5 \cdot 47 \cdot 373 \cdot 2\,998\,279$ and $2\,998\,279 - 1 = 2 \cdot 3^2 \cdot 166571$ we find the Lucas witnesses 3 for $2\,998\,279$, and 3 for p_{14}.

iii) For the Fermat number $F_7 := 2^{2^7} + 1 \sim 3 \cdot 10^{38}$ the ρ-method needs $455\,756\,940$ tries to find the divisor $p_{17} := 59\,649\,589\,127\,497\,217$, hence we get $p_{22} := \frac{F_7}{p_{17}} = 5\,704\,689\,200\,685\,129\,054\,721$ [Morrison-Brillhart, 1971]; the $(p-1)$-method is unsuccessful. The factorisation was originally obtained by the **continued fraction method (CFRAC)**.

From $p_{17} - 1 = 2^9 \cdot 116\,503\,103\,764\,643$ and $116\,503\,103\,764\,643 - 1 = 2 \cdot 7 \cdot 449 \cdot 18\,533\,742\,247$ and $18\,533\,742\,247 - 1 = 2 \cdot 3^3 \cdot 181 \cdot 1\,896\,229$ and $1\,896\,229 - 1 = 2^2 \cdot 3^2 \cdot 52673$ we find the Lucas witnesses 2 for $1\,896\,229$, and 11 for $18\,533\,742\,247$, and 2 for $116\,503\,103\,764\,643$, and 3 for p_{17}. From $p_{22} - 1 = 2^9 \cdot 3^5 \cdot 5 \cdot 12497 \cdot 733\,803\,839\,347$, and $733\,803\,839\,347 - 1 = 2 \cdot 3 \cdot 2203 \cdot 55\,515\,497$, and $55\,515\,497 - 1 = 2^3 \cdot 6\,939\,437$, and $6\,939\,437 - 1 = 2^2 \cdot 7 \cdot 139 \cdot 1783$ we find the Lucas witnesses 2 for $6\,939\,437$, and 3 for $55\,515\,497$, and 2 for $733\,803\,839\,347$, and 23 for p_{22}.

iv) For the Fermat number $F_8 := 2^{2^8} + 1 \sim 1 \cdot 10^{77}$, the ρ-method needs $14\,816\,648$ tries to find the divisor $p_{16} := 1\,238\,926\,361\,552\,897$, hence we get $p_{62} := \frac{F_8}{p_{16}}$ [Brent-Pollard, 1981]; the $(p-1)$-method is unsuccessful. The factorisation was indeed originally obtained by the ρ-method.

From $p_{16} - 1 = 2^{11} \cdot 157 \cdot 3\,853\,149\,761$ and $3\,853\,149\,761 - 1 = 2^6 \cdot 5 \cdot 719 \cdot 16747$ we find the Lucas witnesses 7 for $3\,853\,149\,761$, and 3 for p_{16}. We proceed to show that p_{62} is prime: We have $p_{62} - 1 = 2^{11} \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot n$, where $n \sim 3 \cdot 10^{55}$, and the ρ-method needs $4\,999\,212$ tries to find $n = 31\,618\,624\,099\,079 \cdot n'$, where $n' \sim 1 \cdot 10^{42}$; the $(p-1)$-method finds $31\,618\,624\,099\,079$ with $b := 876769$ and $x := 2$. From $31\,618\,624\,099\,079 - 1 = 2 \cdot 1789 \cdot 10079 \cdot 876769$ we find the Lucas witness 17 for $31\,618\,624\,099\,079$. We proceed to show that n' is prime; then we find the Lucas witness 43 for p_{62}:

The ρ-method needs 5890 and $4\,731\,257$ tries, respectively, to find successively $n' - 1 = 2^4 \cdot 3 \cdot 8861 \cdot 10\,608\,557 \cdot 25\,353\,082\,741\,699 \cdot 9\,243\,081\,088\,796\,207$; in the first step the $(p-1)$-method is successful with $b := 1699$ and $x := 2$, while in the second step the $(p-1)$-method is unsuccessful. From $10\,608\,557 - 1 = 2^2 \cdot 7 \cdot 223 \cdot 1699$ we find the Lucas witness 2 for $10\,608\,557$; from $25\,353\,082\,741\,699 - 1 = 2 \cdot 3^2 \cdot 16879 \cdot 83\,447\,159$ and $83\,447\,159 - 1 = 2 \cdot 41\,723\,579$ and $41\,723\,579 - 1 = 2 \cdot 13 \cdot 1\,604\,753$ and $1\,604\,753 - 1 = 2^4 \cdot 100297$ we find the Lucas witnesses 3 for $1\,604\,753$, and 2 for $41\,723\,579$, and 11 for $83\,447\,159$, and 2 for $25\,353\,082\,741\,699$; from $9\,243\,081\,088\,796\,207 - 1 = 2 \cdot 20939 \cdot 220\,714\,482\,277$ and $220\,714\,482\,277 - 1 = 2^2 \cdot 3^2 \cdot 6\,130\,957\,841$ and $6\,130\,957\,841 - 1 = 2^4 \cdot 5 \cdot 7 \cdot 10\,948\,139$ and $10\,948\,139 - 1 = 2 \cdot 23 \cdot 29^2 \cdot 283$ we find the Lucas witnesses 2 for $10\,948\,139$, and 3 for $6\,130\,957\,841$, and 5 for $220\,714\,482\,277$, and 5 for $9\,243\,081\,088\,796\,207$. This finally yields the Lucas witness 11 for n'.

v) For the Fermat number $F_9 := 2^{2^9} + 1 \sim 1 \cdot 10^{154}$, the ρ-method needs 1563 tries to find the divisor $p_7 := 2\,424\,833$; the $(p-1)$-method finds p_7 with $b \leq 31$ and $x := 37$, and $b := 37$ and $x := 3$. From $p_7 - 1 = 2^{16} \cdot 37$ we find the Lucas witness 3 for p_7. For $n := \frac{F_6}{p_6} \sim 6 \cdot 10^{146}$ we find the Fermat compositeness witness 3; indeed we have $n = p_{49} \cdot p_{99}$ [Lenstra, 1990], the prime factors having the indicated number of digits, where the factorisation was originally obtained by the **Number Field Sieve (NFS)**.

vi) For the Fermat number $F_{10} := 2^{2^{10}} + 1 \sim 2 \cdot 10^{308}$, the ρ-method needs 9005 and 167955 tries, respectively, to find successively the divisors $p_8 := 45\,592\,577$ and $p_{10} := 6\,487\,031\,809$ The $(p-1)$-method finds p_{10} with $b := 37$ and $x := 173$, and $b := 41$ and $x := 3$; then it finds p_8 with $b := 11131$ and $x := 3$. From $p_8 - 1 = 2^{12} \cdot 11131$ we find the Lucas witness 3 for p_8; and from $p_{10} - 1 = 2^{14} \cdot 3^2 \cdot 29 \cdot 37 \cdot 41$ we find the Lucas witness 7 for p_{10}. For $n := \frac{F_{10}}{p_8 \cdot p_{10}} \sim 6 \cdot 10^{290}$ we find the Fermat compositeness witness 3; indeed we have $n = p_{40} \cdot p_{252}$ [Brent, 1995], the prime factors having the indicated number of digits, where the factorisation was originally obtained by the **Elliptic Curve Method (ECM)**.

vii) For the Fermat number $F_{11} := 2^{2^{11}} + 1 \sim 3 \cdot 10^{616}$, the ρ-method needs 178 and 832 tries, respectively, to find successively the prime divisors $p_6 := 974849$ and $p'_6 := 319489$. The $(p-1)$-method finds p'_6 with $b := 2$ and $x := 103$, and $3 \leq b \leq 11$ and $x := 11$, and $b := 13$ and $x := 3$; subsequently it finds p_6 with $b := 17$ and $x := 3$. Indeed we have $p_6 - 1 = 2^{13} \cdot 7 \cdot 17$ and $p'_6 - 1 = 2^{13} \cdot 3 \cdot 13$. For $n := \frac{F_{11}}{p_6 \cdot p'_6} \sim 1 \cdot 10^{605}$ we find the Fermat compositeness witness 3; indeed we have $n = p_{21} \cdot p_{22} \cdot p_{564}$ [Brent, 1988], the prime factors having the indicated number of digits, where the factorisation was originally obtained by the ECM.

viii) For the Fermat number $F_{12} := 2^{2^{12}} + 1 \sim 1 \cdot 10^{1233}$, the ρ-method needs 343 and 730 and 5085 and 384615 and 49 572 772 tries, respectively, to find successively the prime divisor $p_6 := 114689$, and the divisors $p_8 := 26\,017\,793$ and $p'_8 := 63\,766\,529$ and $p_{12} := 190\,274\,191\,361$ and $p_{16} := 1\,256\,132\,134\,125\,569$. The $(p-1)$-method finds p_6 with $b \leq 5$ and $x := 7$, and $b := 7$ and $x := 3$; then it finds p'_8 with $b := 139$ and $x := 3$; next it finds p_8 with $b := 397$ and $x := 3$; and finally it finds p_{12} with $b := 211153$ and $x := 3$; but subsequently it is unsuccessful. Indeed we have $p_6 - 1 = 2^{14} \cdot 7$; from $p_8 - 1 = 2^{16} \cdot 397$ we find the Lucas witness 3 for p_8; from $p'_8 - 1 = 2^{16} \cdot 7 \cdot 139$ we find the Lucas witness 13 for p'_8; from $p_{12} - 1 = 2^{14} \cdot 5 \cdot 11 \cdot 211153$ we find the Lucas witness 3 for p_{12}; and from $p_{16} - 1 = 2^{14} \cdot 7^2 \cdot 53 \cdot 29\,521\,841$ and $29\,521\,841 - 1 = 2^4 \cdot 5 \cdot 369023$ we find the Lucas witnesses 3 for 29 521 841, and 3 for p_{16}. For $n := \frac{F_{12}}{p_6 \cdot p_8 \cdot p'_8 \cdot p_{12} \cdot p_{16}} \sim 2 \cdot 10^{1186}$ we find the Fermat compositeness witness 3; the factorisation of n is not known.

Note that in general already integers of size $\sim 10^{200}$ pose severe problems to factorisation methods. We proceed to present a few ideas behind modern factorisation methods, in particular the use of quadratic forms and sieving techniques.

5 Integer factorisation II

(5.1) The Fermat-Legendre method. Let $1 \neq n \in \mathbb{N}$ be odd. Then we have mutually inverse bijections $\{[x, y] \in \mathbb{N}^2; x \geq y, n = xy\} \leftrightarrow \{[s, t] \in \mathbb{N}_0^2; s > t, n = s^2 - t^2\}$ given by $[x, y] \mapsto [\frac{x+y}{2}, \frac{x-y}{2}]$ and $[s, t] \mapsto [s+t, s-t]$. Thus finding a divisor of n is equivalent to writing n as a difference of two squares.

Hence, if $x, y \in \mathbb{Z}_n$ such that $y \not\in \{\pm x\} \subseteq \mathbb{Z}_n$ and $x^2 = y^2 \in \mathbb{Z}_n$, then we have $(x+y)(x-y) = x^2-y^2 = kn \in \mathbb{Z}$, for some $k \in \mathbb{Z}$, and thus $1 < \gcd(x \pm y, n) < n$. This yields the **Fermat-Legendre method** to find a divisor of n: Choosing $k \in \mathbb{N}$ small, for increasing $x > \lfloor\sqrt{kn}\rfloor$ check whether $x^2 - kn \in \mathbb{N}$ is a square.

For example, let again $n := 7429 = 17 \cdot 19 \cdot 23$, hence $m := \lfloor\sqrt{n}\rfloor = 86$. Choosing $k := 1$, for $x := m + 87 = 173$ we get $x^2 - n = 29929 - 7429 = 22500 = 150^2$, thus letting $y := 150$ we get $n = x^2 - y^2 = (x-y)(x+y) = 23 \cdot 323$.

(5.2) The random squares method [Dixon, 1981]. We keep the setting of (5.1). Letting $n = \prod_{q \in \mathcal{P}} q^{e_q(n)}$ again, there is a natural isomorphism of groups $\mathbb{Z}_n^* \to \prod_{q \in \mathcal{P}, e_q(n)>0} \mathbb{Z}^*_{q^{e_q(n)}}$, where $\mathbb{Z}^*_{q^d}$, for $d \in \mathbb{N}$, is cyclic, and thus $\{\pm 1\} \subseteq \mathbb{Z}^*_{q^d}$ are the only elements of $\mathbb{Z}^*_{q^d}$ of order dividing 2. Hence any square in $\mathbb{Z}_n^*$ has precisely 2^r square roots, where $r := |\{q \in \mathcal{P}; e_q(n) \geq 1\}| \in \mathbb{N}$ is the number of factors in the above direct product, that is the number of prime divisors of n.

Thus, given $x \in \mathbb{Z}_n^*$, if $y \in \mathbb{Z}_n^*$ is randomly chosen such that $x^2 = y^2 \in \mathbb{Z}_n^*$, then with probability $\frac{2^r-2}{2^r} = 1 - \frac{1}{2^{r-1}}$ we have $y \not\in \{\pm x\}$. Hence, assuming again that n is not a prime power, that is $r \geq 2$, computing $\gcd(x \pm y, n)$ yields a proper divisor of n with probability $\geq \frac{1}{2}$. Still, given $x \in \mathbb{Z}_n^*$, the task to find some $y \in \mathbb{Z}_n^* \setminus \{\pm x\}$ such that $x^2 = y^2 \in \mathbb{Z}_n^*$ remains:

Let $b \in \mathbb{N}$, and assume that $\mathcal{P}_{\leq b} = \{p_1, \ldots, p_l\}$, where $l \in \mathbb{N}_0$, is known. Letting additionally $p_0 := -1$, the sequence $[p_0, p_1, \ldots, p_l]$ is called the **factor base** associated with b. For $x \in \mathbb{Z}_n^*$ let $x' \in \{-\frac{n-1}{2}, \ldots, \frac{n-1}{2}\}$ such that $x^2 = x' \in \mathbb{Z}_n^*$. If $|x'| \in \mathbb{N}$ is b-smooth, then we have $x' = \prod_{i=0}^{l} p_i^{e_i(x)} \in \mathbb{Z}$, where $e_i(x) \in \mathbb{N}_0$ and $e_0(x) \leq 1$; in this case x is called a ***b*-number**, the sequence $e(x) := [e_0(x), \ldots, e_l(x)] \in \mathbb{N}_0^{l+1}$ is called its **exponent vector**, and taking remainders modulo 2 componentwise, we get the **reduced** exponent vector $\overline{e}(x) := [\overline{e_0}(x), \ldots, \overline{e_l}(x)] \in \mathbb{Z}_2^{l+1}$; note that $\mathbb{Z}_2$ is a field.

Let $x_1, \ldots, x_k \in \mathbb{Z}_n^*$ be b-numbers fulfilling $\sum_{j=1}^{k} \overline{e}(x_j) = 0 \in \mathbb{Z}_2^{l+1}$, thus $f_i := \sum_{j=1}^{k} e_i(x_j) \in \mathbb{N}_0$ is even, for all $i \in \{0, \ldots, l\}$. Letting $x := \prod_{j=1}^{k} x_j \in \mathbb{Z}$ and $y := \prod_{i=0}^{l} p_i^{\frac{f_i}{2}} \in \mathbb{Z}$ we have $y^2 = \prod_{i=0}^{l} p_i^{f_i} = \prod_{j=1}^{k} \prod_{i=0}^{l} p_i^{e_i(x_j)} = \prod_{j=1}^{k} x_j' \in \mathbb{Z}$, implying that $y^2 = \prod_{j=1}^{k} x_j^2 = x^2 \in \mathbb{Z}_n^*$; note that x and y can be computed in $\mathbb{Z}_n$. Hence it suffices to find such a $\mathbb{Z}_2$-linear dependency between reduced exponent vectors in order to obtain a proper divisor of n with a probability $\geq \frac{1}{2}$.

Exponent vectors can be computed independently from each other, which hence can be done using distributed computing. Moreover, a $\mathbb{Z}_2$-linear dependency between reduced exponent vectors occurs for some $k \leq l + 2$, and is detected using Gaussian elimination, where specially tailored techniques for sparse matrices over $\mathbb{Z}_2$ can be used; note that the running time of Gaussian elimination, in general, is in $O(l^3) = O((\frac{b}{\ln(b)})^3)$, by the Prime Number Theorem.

If the $x_j \in \mathbb{Z}_n^*$ are chosen randomly, this yields a Las-Vegas algorithm to factor n, where using the **Canfield-Erdős-Pomerance Theorem [1983]** estimating the fraction of b-smooth integers in $\mathbb{Z}_n^*$, and finding an optimal tradeoff between the constraints that the fraction of b-numbers increases as b gets larger, but the cost of Gaussian elimination decreases as b gets smaller, we get **Dixon's Theorem**: Letting $L(n) := L_{\frac{1}{2},1}(n) = \exp(\sqrt{\ln(n) \cdot \ln(\ln(n))})$, see (1.2), a proper divisor of n is found, using a smoothness bound $b \in O(L_{\frac{1}{2},\frac{1}{2}}(n)) = O(\sqrt{L(n)})$, in **subexponential** time $O(L^{2+\epsilon}(n))$, for $\epsilon > 0$.

For example, let again $n := 7429 = 17 \cdot 19 \cdot 23$. We choose $b := \lfloor \sqrt{L(n)} \rfloor = 9$, hence the factor base is $[-1, 2, 3, 5, 7]$, and $l = 4$. Using the random choices $x_1 := 6708$ and $x_2 := 2468$, we get $x_1^2 = 7240 \in \mathbb{Z}_n$ and $x_1' = 7240 - n = -189 = (-1)^1 \cdot 3^3 \cdot 7$, as well as $x_2^2 = 6673 \in \mathbb{Z}_n$ and $x_2' = 6673 - n = -756 = (-1)^1 \cdot 2^2 \cdot 3^3 \cdot 7$, that is $e(x_1) = [1, 0, 3, 0, 1] \in \mathbb{N}_0^5$ and $e(x_2) = [1, 2, 3, 0, 1] \in \mathbb{N}_0^5$, in terms of exponent vectors, which yields the reduced exponent vectors $\overline{e}(x_1) = \overline{e}(x_2) = [1, 0, 1, 0, 1] \in \mathbb{Z}_2^5$. Hence we infer $\overline{e}(x_1) + \overline{e}(x_2) = 0 \in \mathbb{Z}_2^5$, and thus $\frac{1}{2}(e(x_1) + e(x_2)) = [1, 1, 3, 0, 1] \in \mathbb{N}_0^5$. From that we obtain $x = x_1 x_2 = 16\,555\,344 = 3532 \in \mathbb{Z}_n$, and $y = (-1)^1 \cdot 2^1 \cdot 3^3 \cdot 5^0 \cdot 7^1 = -378 = 7051 \in \mathbb{Z}_n$, thus $\gcd(x + y, n) = 19$ and $\gcd(x - y, n) = 391$, hence $n = 19 \cdot 391$.

(5.3) The Quadratic Sieve (QS) [Pomerance, 1984]. We keep the setting of (5.2). Let $m := \lfloor \sqrt{n} \rfloor \in \mathbb{N}$ and $f := (X + m)^2 - n \in \mathbb{Z}[X]$. Rather than looking for b-numbers amongst random choices from $\mathbb{Z}_n^*$, it should be much more likely to find b-smooth integers amongst the $f(x) \in \mathbb{Z}$ whenever $x \in \mathbb{Z}$ is small.

To this end, we choose $c \in \mathbb{N}$, small compared to m, and consider the **sieve interval** $\mathcal{I} := \{-c, \ldots, c\} \subseteq \mathbb{Z}$. Thus for $x \in \mathcal{I}$ we have $f(x) = x^2 + 2xm + (m^2 - n) \sim 2xm$, hence $|f(x)|$ is bounded by $\sim 2cm$. Then the **Pomerance Conjecture** says that the fraction of b-smooth integers in $\{f(x) \in \mathbb{Z}; x \in \mathcal{I}\}$ is asymptotically the same as the fraction of b-smooth integers in $\{1, \ldots, m\}$. This shows that running through $x \in \mathcal{I}$ provides a sufficiently good supply of b-numbers. Moreover, instead of finding the latter one by one, applying a **sieve strategy** to $\mathcal{I}$ yields whole bunches of b-numbers at the same time:

If $p \in \mathbb{N}$ is a prime, then from $f(x+p) = (x+m+p)^2 - n = f(x) + p(2x+2m+p) \in \mathbb{Z}$ we conclude that $f(x) = 0 \in \mathbb{Z}_p$ if and only if $f(x + kp) = 0 \in \mathbb{Z}_p$ for all $k \in \mathbb{Z}$. Thus, we first compute $f(x) \in \mathbb{Z}$ for all $x \in \mathcal{I}$. Subsequently, for the primes $p \in \mathcal{P}_{\leq b}$ in turn, we determine all $x \in \{-\frac{p-1}{2}, \ldots, \frac{p-1}{2}\}$ if p is odd, and $x \in \{0, 1\}$ for $p = 2$, such that $f(x) = 0 \in \mathbb{Z}_p$, then for those x we run through $\{x + kp \in \mathbb{Z}; k \in \mathbb{Z}\} \cap \mathcal{I}$, divide out the maximum p-power dividing $f(x + kp) \in \mathbb{Z}$, and replace the latter by the quotient.

Assuming the validity of the Pomerance Conjecture, and using sieving as described above, this yields a Las-Vegas algorithm to factor n, where we have **Pomerance's Theorem**: A proper divisor of n is found, using a smoothness bound of size $b \in O(\sqrt{L(n)})$ and a sieving interval of size $c \sim b^2 \in O(L(n))$, in subexponential time $O(L^{1+\epsilon}(n))$, for $\epsilon > 0$.

In practice, smaller values are used: For $n \sim 10^{50}$ smoothness bounds of size $b \sim 5 \cdot 10^4$, by the Prime Number Theorem corresponding to factor bases of size

$l \sim \frac{b}{\ln(b)} \sim 5 \cdot 10^3$, and sieve intervals of size $c \sim 10^5$ are used, while $L \sim 10^{10}$; for $n \sim 10^{100}$ smoothness bounds of size $b \sim 10^6$, corresponding to factor bases of size $l \sim 5 \cdot 10^4$, and sieve intervals of size $c \sim 10^7$ are used, while $L \sim 10^{15}$.

We describe a further improvement: For any odd prime $p \in \mathbb{N}$ such that $p \nmid n$, from $f(x) = 0 \in \mathbb{Z}_p$ we infer that $n = (x+m)^2 \in \mathbb{Z}_p^*$ is a square. Since by Artin's Theorem $\mathbb{Z}_p^*$ is cyclic, the latter by **Euler's Criterion** is equivalent to $n^{\frac{p-1}{2}} = 1 \in \mathbb{Z}_p^*$. Thus the odd primes p such that $n^{\frac{p-1}{2}} \neq 1 \in \mathbb{Z}_p^*$ can be discarded from the factor base in advance. Moreover, since $\{\pm 1\} \subseteq \mathbb{Z}_p^*$ are the only elements of order dividing 2, for the admissible odd primes there are precisely two square roots $\pm\sqrt{n} \in \mathbb{Z}_p^*$ of $n \in \mathbb{Z}_p^*$, hence we have $f(x) = 0 \in \mathbb{Z}_p$ if and only if $x \in \{-m \pm \sqrt{n}\} \subseteq \mathbb{Z}_p$. Similarly, $p = 2$ is always admissible, and from $n = 1 \in \mathbb{Z}_2$ we get $f(x) = 0 \in \mathbb{Z}_2$ if and only if $x = m + 1 \in \mathbb{Z}_2$.

For example, let again $n := 7429 = 17 \cdot 19 \cdot 23$, hence we have $m := \lfloor \sqrt{n} \rfloor = 86$. We again choose $b := 9$, hence the factor base is $[-1, 2, 3, 5, 7]$, and $l = 4$; note that by Euler's Criterion all these primes are admissible. We choose the sieve interval $\mathcal{I} := \{-3, \ldots, 3\}$, that is $c := 3$:

x	-3	-2	-1	0	1	2	3
$(x+m)^2 - n$	-540	-373	-204	-33	140	315	492
sieve with 2	-135		-51		35		123
sieve with 3	-5		-17	-11		35	41
sieve with 5	-1				7	7	
sieve with 7					1	1	

Thus $\{-3, 1, 2\} \subseteq \mathcal{I}$ yields b-numbers, whose associated matrix of exponent vectors is $M := \begin{bmatrix} 1 & 2 & 3 & 1 & . \\ . & 2 & . & 1 & 1 \\ . & . & 2 & 1 & 1 \end{bmatrix} \in \mathbb{N}_0^{3 \times 5}$. Reduction yields $\begin{bmatrix} 1 & . & 1 & 1 & . \\ . & . & . & 1 & 1 \\ . & . & . & 1 & 1 \end{bmatrix} \in \mathbb{Z}_2^{3 \times 5}$, whose kernel is $\langle [0,1,1] \rangle_{\mathbb{Z}_2}$. This yields $x = (1+m) \cdot (2+m) = 87 \cdot 88 = 7656 = 227 \in \mathbb{Z}_n$, and from $\frac{1}{2} \cdot [0,1,1] \cdot M = [0,1,1,1,1] \in \mathbb{N}_0^5$ we get $y = (-1)^0 \cdot 2^1 \cdot 3^1 \cdot 5^1 \cdot 7^1 = 210 \in \mathbb{Z}_n$. Thus we obtain $\gcd(x - y, n) = 17$ and $\gcd(x + y, n) = 437$, hence $n = 17 \cdot 437$.

(5.4) Other factorisation methods. Generalising the $(p-1)$-method, which is based on the cyclic group $\mathbb{Z}_p^*$ of order $p - 1$, using the cyclic group $\mathbb{F}_{p^2}^* / \mathbb{F}_p^*$ of order $p + 1$ instead, where $\mathbb{F}_{p^2}$ is the field with p^2 elements, yields the **$(p+1)$-method** [Guy, 1975; Williams, 1982], and more generally the **$\Phi_k(p)$-method** [Bach-Shallit, 1988], where $\Phi_k \in \mathbb{Z}[X]$ is the k-th **cyclotomic polynomial**.

In the **Elliptic Curve Method (ECM)** [Lenstra, 1987], which also generalises the $(p-1)$-method, the group G of points of an elliptic curve over $\mathbb{F}_p$ is used instead. The **Hasse bound** yields $||G| - (p+1)| \leq 2\sqrt{p}$, hence we again have $|G| \sim p$, and varying the elliptic curve it is likely to find a group G such that $|G|$ only has small prime divisors. Conjecturally, the ECM has Las-Vegas subexponential running time $O(L^{1+\epsilon}(n))$, for $\epsilon > 0$. Actually the running time of the ECM depends on the smallest prime divisor p of n, inasmuch it conjecturally has Las-Vegas subexponential running time $O(L^{\sqrt{\frac{1}{2}}+\epsilon}(p))$, for $\epsilon > 0$; thus the ECM is superior to the QS and its improvements mentioned below whenever n has a small prime divisor.

The earliest method using quadratic forms is the **continued fraction method (CFRAC)** [Morrison-Brillhart, 1971], which conjecturally runs in Las-Vegas subexponential time $O(L^{\sqrt{\frac{3}{2}}+\epsilon}(n))$, for $\epsilon > 0$. **Shanks's class group method [1969]** and the **Square Form Factorisation method (SQUFOF)** [Shanks, 1972], using the **ideal class groups** of imaginary and real **quadratic number fields**, respectively, have Las-Vegas exponential running time $O(\exp((\frac{1}{4}+\epsilon)\ln(n)))$, for $\epsilon > 0$, but combining these with the ideas of the $(p-1)$-method and sieving techniques yields the **Schnorr-Lenstra class group method [1984]**, which has Las-Vegas subexponential running time $O(L^{1+\epsilon}(n))$, for $\epsilon > 0$.

Generalising the quadratic sieve, the **Multi-Polynomial Quadratic Sieve (MPQS)** [Pomerance, 1987], conjecturally has Las-Vegas subexponential running time $O(L^{1+\epsilon}(n))$, for $\epsilon > 0$. The **Number Field Sieve (NFS)** [Lenstra-Lenstra-Manasse-Pollard, 1990] generalises the QS from quadratic to general **number fields**, and conjecturally has Las-Vegas subexponential running time $O(L_{\frac{1}{3}}^{c+\epsilon}(n))$, where $L_{\frac{1}{3}}(n) := L_{\frac{1}{3},1}(n) = \exp(\sqrt[3]{\ln(n) \cdot \ln(\ln(n))^2})$, see (1.2), and $c := \sqrt[3]{\frac{64}{9}}$, for $\epsilon > 0$. The running times of the latter methods do not depend on the size of the prime divisors of n, contrary to the ECM method. Although the running times of the ECM and the MPQS have the same asymptotics, for arbitrary n the latter is superior, and the asymptotically better NFS becomes faster than the MPQS for $n \sim 10^{130}$. Finally, although the asymptotics of the running time of the NFS is quite close to polynomial, it still remains an open problem whether integer factorisation can be done in Las-Vegas polynomial time.

6 Exercises (in German)

(6.1) Aufgabe: Asymptotisches Verhalten.
a) Man zeige ohne Benutzung der Stirling-Formel: Für $n \in \mathbb{N}$ gelten $\ln(n!) \in O(n\ln(n))$ und $n\ln(n) \in O(\ln(n!))$.
b) Für festes $k \in \mathbb{N}_0$ und variables $n \in \mathbb{N}$ zeige man: Es gilt $\sum_{i=1}^{n} i^k \sim \frac{n^{k+1}}{k+1}$. Bleibt diese Aussage auch richtig, wenn man n fest läßt und k variiert?
c) Man betrachte die **Fibonacci-Zahlen** $F_n := F_{n-1} + F_{n-2} \in \mathbb{N}$, für $n \geq 3$, wobei $F_2 = F_1 := 1$. Man gebe einfache Funktionen an, die zu F_n bzw. der Bitlänge von F_n asymptotisch äquivalent sind.

(6.2) Aufgabe: Größenordnungen.
a) Man ordne die folgenden Zahlen ihrer asymptotischen Größe nach:
i) die Bitlänge von 2^n; **ii)** die Bitlänge von $\lfloor\sqrt{n}\rfloor!$; **iii)** die Anzahl konsekutiver 0en am Ende der Binärdarstellung von $n!$; **iv)** die Bitlänge des Werts eines Polynoms fünften Grades mit Koeffizienten der Bitlänge 20 an der Stelle n; **v)** die Anzahl der nötigen Probedivisionen zum Beweis, daß n prim ist.

b) Man betrachte die folgenden Funktionen $\mathbb{N}\setminus\{1\} \to \mathbb{R}_{>0}$, wobei $0 < \epsilon < 1 < c$:

$$1 < \ln(\ln(n)) < \ln(n) < e^{(\ln(n))^{\frac{1}{2}} \cdot (\ln(\ln(n)))^{\frac{1}{2}}} < n^{\epsilon} < n^{c} < n^{\ln(n)} < c^{n} < n^{n} < c^{c^{n}}$$

Man zeige, daß für je zwei dieser Funktionen mit $f < g$ auch $f \in o(g)$ gilt.

(6.3) Aufgabe: Bitoperationen.
a) Man zeige: Für $n \in \mathbb{N}$ kann man $n!$ mit $O(n^2\ln^2(n))$ Bitoperationen berechnen. Wieviele Bitoperationen braucht man für n^n bzw. n^k, für $k \in \mathbb{N}$?
b) Man zeige: Für $n \in \mathbb{N}$ gilt $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$. Wieviele Bitoperationen braucht man zur Berechnung der linken bzw. der rechten Seite dieser Gleichung?
c) Für $i \in \mathbb{N}$ sei $F_i \in \mathbb{N}$ die zugehörige Fibonacci-Zahl. Wieviele Bitoperationen braucht man zur Berechnung von $\sum_{i=1}^{n} F_i$ bzw. $\prod_{i=1}^{n} F_i$, für $n \in \mathbb{N}$?
d) Für $1 \neq z \in \mathbb{N}$ und $n \in \mathbb{N}$ seien $\mathcal{P}_{z,n} := \{p \in \mathbb{N} \text{ prim}; b_z(p) \leq n\}$. Wieviele Bitoperationen braucht man zur Berechnung von $\sum \mathcal{P}_{z,n}$ bzw. $\prod \mathcal{P}_{z,n}$?

(6.4) Aufgabe: Subtraktion.
Man gebe einen Algorithmus zur Subtraktion zweier Zahlen $m, n \in \mathbb{N}$ an. Wie entscheidet man, ob $m \geq n$ gilt? Wieviele Bitoperationen werden benötigt?

(6.5) Aufgabe: Euklidischer Algorithmus.
Für $q, m, n \in \mathbb{N}$ mit $q \neq 1$ zeige man: Es gilt $\mathrm{ggT}(q^m - 1, q^n - 1) = q^{\mathrm{ggT}(m,n)} - 1$.

(6.6) Aufgabe: Satz von Lamé.
Es seien $m \geq n \in \mathbb{N}$. Man zeige, daß der Euklidische Algorithmus zur Berechnung von $\mathrm{ggT}(m,n)$ höchstens $\lfloor\frac{\ln(\sqrt{5}\cdot n)}{\ln(\frac{1+\sqrt{5}}{2})}\rceil - 1$ Divisionsschritte benötigt.

(6.7) Aufgabe: Binärer Euklidischer Algorithmus [Stein, 1967].
a) Es seien $m, n \in \mathbb{N}$. Man zeige: Es gilt

$$\mathrm{ggT}(m,n) = \begin{cases} 2 \cdot \mathrm{ggT}(\frac{m}{2}, \frac{n}{2}), & \text{falls } m \text{ und } n \text{ gerade sind,} \\ \mathrm{ggT}(\frac{m}{2}, n), & \text{falls } m \text{ gerade und } n \text{ ungerade ist,} \\ \mathrm{ggT}(\frac{m-n}{2}, n), & \text{falls } m \text{ und } n \text{ ungerade sind.} \end{cases}$$

b) Man zeige, daß der folgende Algorithmus $\mathrm{ggT}(m,n) \in \mathbb{N}$ berechnet, und gebe unter Verwendung von $\max\{b_2(m), b_2(n)\}$ eine Abschätzung für die benötigte Anzahl von Bitoperationen an. Welche Vorteile und Nachteile besitzt dieser Algorithmus gegenüber dem Euklidischen Algorithmus?

```
• $k \leftarrow 0$.
• while $(m \bmod 2) = 0$ and $(n \bmod 2) = 0$ do
    • $m \leftarrow \frac{m}{2}$
    • $n \leftarrow \frac{n}{2}$
    • $k \leftarrow k + 1$
• while $(m \bmod 2) = 0$ do
    • $m \leftarrow \frac{m}{2}$
• while $(n \bmod 2) = 0$ do
    • $n \leftarrow \frac{n}{2}$
• while $m \neq n$ do
    • $t \leftarrow \frac{m-n}{2}$
    • while $(t \bmod 2) = 0$ do
        • $t \leftarrow \frac{t}{2}$
    • if $t > 0$ then
        • $m \leftarrow t$
    • else
        • $n \leftarrow -t$
• return $2^k \cdot m$
```

(6.8) Aufgabe: Lineare diophantische Gleichungen.
Es seien $a, b, c \in \mathbb{Z}$ mit $[a, b] \neq [0, 0]$. Man zeige: Die **lineare diophantische Gleichung** $aX + bY = c$ hat genau dann eine Lösung $[x, y] \in \mathbb{Z}^2$, wenn $\mathrm{ggT}(a, b) \mid c$ gilt. Wie sieht in diesem Fall die Lösungsgesamtheit aus?

(6.9) Aufgabe: Matrixmultiplikation.
Für $k, m, n \in \mathbb{N}$ seien $A \in \mathbb{Z}^{k \times m}$ und $B \in \mathbb{Z}^{m \times n}$. Wieviele Ringoperationen braucht man zur klassischen Berechnung des Matrixprodukts $AB \in \mathbb{Z}^{k \times n}$?

(6.10) Aufgabe: Polynomarithmetik.
Es seien R ein kommutativer Ring, F ein Körper, und $R[X]$ sowie $F[X]$ die zugehörigen Polynomringe.
a) Man formuliere den Karatsuba-Algorithmus zur Multiplikation der Polynome $0 \neq f, g \in R[X]$, und zeige für $\deg(f) \geq \deg(g)$, daß hierzu $O(\deg(f)^{\log_2 3})$ Ringoperationen im Ring R benötigt werden.

b) Man formuliere den Erweiterten Euklidischen Algorithmus für Polynome $0 \neq f, g \in F[X]$, und zeige, daß hierzu $O(\deg(f) \cdot \deg(g))$ Ringoperationen im Ring F benötigt werden.

(6.11) Aufgabe: Restklassen.
a) Es seien $n \in \mathbb{N}$ und $\overline{a} \in \mathbb{Z}_n := \{0, \ldots, n-1\}$ der Rest der Division von $a \in \mathbb{Z}$ durch n. Zahlen $a, b \in \mathbb{Z}$ mit $\overline{a} = \overline{b}$ heißen **kongruent** modulo n; man schreibt $a \equiv b \pmod{n}$. Man zeige: Es gilt genau dann $a \equiv b \pmod{n}$, wenn n ein Teiler von $a - b$ ist, und folgere, daß Kongruenz modulo n eine Äquivalenzrelation auf $\mathbb{Z}$ ist. Die zugehörigen Äquivalenzklassen heißen **Restklassen** modulo n, und die Restklasse von $a \in \mathbb{Z}$ wird mit $[a]$ bezeichnet.
b) Man zeige: Sind $a, a', b, b' \in \mathbb{Z}$ mit $[a] = [a']$ und $[b] = [b']$, so gilt auch $[a+b] = [a'+b']$ und $[ab] = [a'b']$. Man folgere, daß dadurch (repräsentantenweise) Addition und Multiplikation auf der Menge $\mathbb{Z}/n\mathbb{Z} := \{[a] \subseteq \mathbb{Z}; a \in \mathbb{Z}\}$ der Restklassen modulo n definiert werden, die die Kommutativ-, Assoziativ- und Distributivgesetze erfüllen, und daß $[0]$ bzw. $[1]$ die jeweiligen eindeutigen neutralen Elemente sind. Was sind die additiven bzw. multiplikativen Inversen zu $[a] \in \mathbb{Z}/n\mathbb{Z}$?
c) Man zeige: Die Abbildung $\rho_n \colon \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}_n \colon [a] \mapsto \overline{a}$ ist eine Bijektion, die mit Addition und Multiplikation auf $\mathbb{Z}/n\mathbb{Z}$ bzw. $\mathbb{Z}_n$ verträglich ist; dabei trage $\mathbb{Z}_n$ die bekannten Operationen.

(6.12) Aufgabe: Chinesischer Restsatz.
a) Für $k \in \mathbb{N}$ seien $n_1, \ldots, n_k \in \mathbb{N}$ **paarweise teilerfremd**, d. h. es ist $\mathrm{ggT}(n_i, n_j) = 1$, für alle $i \neq j \in \{1, \ldots, k\}$, sowie $n := \prod_{i=1}^{k} n_i \in \mathbb{N}$. Für $a \in \mathbb{Z}$ seien $\overline{a} \in \mathbb{Z}_n$ und $\overline{a}_{(i)} \in \mathbb{Z}_{n_i}$ die Reste von a bei Division durch n bzw. n_i, für alle $i \in \{1, \ldots, k\}$. Man zeige: Die **natürliche Abbildung** $\mathbb{Z}_n \to \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k} \colon \overline{a} \mapsto [\overline{a}_{(1)}, \ldots, \overline{a}_{(k)}]$ ist wohldefiniert, bijektiv, und verträglich mit Addition und Multiplikation.
b) Man gebe einen Algorithmus an, der für ein gegebenes Tupel $[b_1, \ldots, b_k] \in \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k}$ von **simultanen Kongruenzen** das eindeutig bestimmte Element $a \in \mathbb{Z}_n$ mit $[\overline{a}_{(1)}, \ldots, \overline{a}_{(k)}] = [b_1, \ldots, b_k]$ berechnet.

(6.13) Aufgabe: Modulare Inverse.
Man bestimme alle $a \in \{0, \ldots, 999\}$, so daß $67a$ in Dezimaldarstellung die drei letzten Ziffern 123 hat. Was passiert, wenn man $68a$ und/oder 124 betrachtet?

(6.14) Aufgabe: Satz von Fermat.
a) Es sei $p \in \mathbb{N}$ prim. Man zeige den **Kleinen Satz von Fermat [1632]**: Für alle $a \in \mathbb{Z}$ gilt $\overline{a}^p = \overline{a} \in \mathbb{Z}_p$. Man gebe zwei Beweise an, zum einen als Folgerung aus dem Satz von Fermat, zum anderen unter Benutzung von **Freshman's Dream** $(a+b)^p = a^p + b^p \in \mathbb{Z}_p$, für $a, b \in \mathbb{Z}_p$.
b) Wie kann man mittels des Satzes von Fermat das multiplikative Inverse von $a \in \mathbb{Z}_p^*$ bestimmen? Wieviele Bitoperationen sind dazu notwendig?

(6.15) Aufgabe: Teilbarkeit.
a) Es sei $n \in \mathbb{N}$. Man zeige: Gilt $2^n = 1 \in \mathbb{Z}_n$, so ist $n = 1$. Daraus folgere man: Gilt $2^{n+1} = 2 \in \mathbb{Z}_n$, so ist $n = 1$ oder $n = 2$.
b) Es sei $n \in \mathbb{N}$. Man zeige: Ist $n^4 + 4^n \in \mathbb{N}$ prim, so ist $n = 1$.

(6.16) Aufgabe: Wilson-Primzahltest.
a) Man zeige den **Satz von Wilson**: Es sei $1 \neq n \in \mathbb{N}$. Dann ist n genau dann prim, wenn $\overline{(n-1)!} = -1 \in \mathbb{Z}_n$ gilt.
b) Wieviele Bitoperationen benötigt ein Primzahltest, der diesen Satz benutzt?

(6.17) Aufgabe: Pocklington-Lehmer-Primzahltest.
a) Es seien $n \in \mathbb{N}$, sowie $p \in \mathbb{N}$ prim und $e \in \mathbb{N}$ mit $p^e \mid n-1$ und $p^{e+1} \nmid n-1$. Weiter sei $a \in \mathbb{Z}_n$ mit $a^{n-1} = 1 \in \mathbb{Z}_n$ und $a^{\frac{n-1}{p}} - 1 \in \mathbb{Z}_n^*$. Man zeige: Für jeden Teiler $d \in \mathbb{N}$ von n gilt $p^e \mid d-1$.
b) Es sei $n - 1 = ml$, wobei $m, l \in \mathbb{N}$ mit $\mathrm{ggT}(m, l) = 1$ und $m \geq \sqrt{n}$. Man zeige: Es ist n genau dann prim, wenn es für jeden Primteiler $p \mid m$ ein $a \in \mathbb{Z}_n$ gibt, das die Bedingung in a) erfüllt.

(6.18) Aufgabe: Carmichael-Zahlen.
Es sei $n \in \mathbb{N}$ zerlegbar. Man zeige:
a) Es ist n genau dann eine Carmichael-Zahl, wenn n quadratfrei ist und für jeden Primteiler $p \in \mathbb{N}$ von n gilt $p - 1 \mid n - 1$.
b) Ist n eine Carmichael-Zahl, so ist n ungerade und hat mindestens drei paarweise verschiedene Primteiler.
c) Ist $p \in \mathbb{N}$ eine ungerade Primzahl, so gibt es nur endlich viele durch p teilbare Carmichael-Zahlen mit genau drei paarweise verschiedenen Primteilern. Man bestimme diese für $p = 3$ und $p = 5$.

(6.19) Aufgabe: Carmichael-Funktion.
Für $n = \prod_{i=1}^{r} p_i^{e_i} \in \mathbb{N}$ ungerade, mit $r \in \mathbb{N}$, paarweise verschiedenen Primzahlen $p_i \in \mathbb{N}$ sowie $e_i \in \mathbb{N}$, sei $\lambda(n) := \mathrm{kgV}(\varphi(p_1^{e_1}), \ldots, \varphi(p_r^{e_r})) \in \mathbb{N}$.
a) Man zeige: Für alle $a \in \mathbb{Z}_n^*$ gilt $a^{\lambda(n)} = 1 \in \mathbb{Z}_n^*$. Daraus folgere man, daß n genau dann eine Carmichael-Zahl ist, wenn $\lambda(n) \mid n - 1$ gilt.
b) Man zeige: Es ist $V_n := \{a \in \mathbb{Z}_n^*; a^{\frac{\lambda(n)}{2}} = \pm 1 \in \mathbb{Z}_n^*\}$ eine Untergruppe von $\mathbb{Z}_n^*$, und es gilt $V_n = \mathbb{Z}_n^*$ genau dann, wenn n eine Primzahlpotenz ist.
c) Kann man daraus einen randomisierten Zerlegbarkeitstest entwickeln, wenn man voraussetzt, daß n keine Primzahlpotenz ist?

(6.20) Aufgabe: Primzahlpotenzen.
a) Für $n \in \mathbb{N}$ zeige man: Gibt es (einen Fermat-Zeugen) $a \in \mathbb{Z}_n^*$ mit $a^{n-1} - 1 \in \mathbb{Z}_n^*$, so ist n keine Primzahlpotenz. Man beweise ein Analogon für starke Zeugen.
b) Man zeige: Ist $n \in \mathbb{N}$ eine zerlegbare Primzahlpotenz, so hat n einen Fermat-Zeugen. Man gebe einen randomisierten Algorithmus zur Bestimmung des Primteilers von n an.

(6.21) Aufgabe: Fermat-Lügner.
a) Es seien $p \in \mathbb{N}$ prim, so daß auch $2p-1 \in \mathbb{N}$ prim ist, und $n := p(2p-1) \in \mathbb{N}$. Man zeige: Es gibt genau $\frac{1}{2}\varphi(n)$ Fermat-Lügner für n.
b) Es seien $p \neq q \in \mathbb{N}$ prim mit $p, q \equiv 3 \pmod 4$ und $\mathrm{ggT}(p-1, q-1) = 2$, sowie $n := pq \in \mathbb{N}$. Man zeige, daß $|\{a^{n-1} \in \mathbb{Z}_p^*; a \in \mathbb{Z}_p^*\}| = |\{a^2 \in \mathbb{Z}_p^*; a \in \mathbb{Z}_p^*\}|$ gilt, und bestimme daraus den Anteil der Fermat-Lügner in $\mathbb{Z}_n^*$.

(6.22) Aufgabe: Starke Lügner.
a) Es seien $p \in \mathbb{N}$ prim mit $p \equiv 3 \pmod 4$, so daß auch $2p-1 \in \mathbb{N}$ prim ist, und $n := p(2p-1) \in \mathbb{N}$. Man zeige: Es gibt genau $\frac{1}{4}\varphi(n)$ starke Lügner für n.
b) Für $t \in \mathbb{N}$ sei $n_t := \prod\{p \in \{3, \ldots, t\}; p \text{ prim}\} \in \mathbb{N}$. Man zeige, daß $\{\pm 1\} \subseteq \mathbb{Z}_{n_t}^*$ die einzigen starken Lügner für n sind.

(6.23) Aufgabe: Satz von Miller-Rabin.
Es seien $1 \neq n \in \mathbb{N}$ ungerade, und $n-1 = 2^l \cdot m$, wobei $l, m \in \mathbb{N}$ mit m ungerade. Weiter seien $O_n := \{a \in \mathbb{Z}_n^*; a^m = 1\}$ und $B_{n,k} := \{a \in \mathbb{Z}_n^*; a^{2^k m} = -1\}$ für $k \in \mathbb{N}_0$, sowie $B_n := O_n \,\dot\cup\, \coprod_{k \in \mathbb{N}_0} B_{n,k} \subseteq \mathbb{Z}_n^*$. Man zeige:
a) Sind $a \in \mathbb{Z}_n^*$ und $e \in \mathbb{N}$ mit $a^e = -1 \in \mathbb{Z}_n^*$, so gilt $2^l \nmid e$. Daraus folgere man: Es gilt $B_{n,k} = \emptyset$ für $k \geq l$.
b) Die Menge B_n ist genau dann eine Untergruppe von $\mathbb{Z}_n^*$, wenn n eine Primzahlpotenz ist oder einen Primteiler p mit $4 \mid p+1$ besitzt.

(6.24) Aufgabe: Pollard-ρ-Algorithmus.
Man betrachte die durch $x_0 := 1 \in \mathbb{N}$ und $x_i := x_{i-1}^2 + 1 \in \mathbb{N}$, für $i \in \mathbb{N}$, rekursiv definierte Folge. Für $1 \neq n \in \mathbb{N}$ sei $\kappa(n) := \min\{i \in \mathbb{N}; x_i = x_{2i} \in \mathbb{Z}_n\}$
a) Es seien $1 \neq n \in \mathbb{N}$, und $k \in \mathbb{N}$ mit $\mathrm{ggT}(x_i - x_{2i}, n) = 1$ für alle $i \in \{1, \ldots, k\}$. Man zeige: Für alle Primteiler $p \in \mathbb{N}$ von n gilt $\kappa(p) > k$.
b) Mittels GAP bestimme man $K \in \mathbb{N}$ minimal mit $\kappa(p) \leq K$ für alle Primzahlen $p < 10^6$. Wie kann man damit zeigen, daß n keinen Primteiler $< 10^6$ hat?

(6.25) Aufgabe: Quadratwurzeln.
Es sei $1 \neq n \in \mathbb{N}$ ungerade und keine Primzahlpotenz. Man zeige: Die Probleme 'Berechnung eines nichttrivialen Teilers von n' und 'Berechnung einer Wurzel eines Quadrats in $\mathbb{Z}_n^*$' sind randomisiert polynomzeit-äquivalent.

(6.26) Aufgabe: $(n+1)$-Methoden.
a) Es sei $1 \neq n \in \mathbb{N}$ ungerade. Man gebe einen Primzahltest an, der analog zum Pocklington-Lehmer-Primzahltest, siehe Aufgabe (6.17), statt der Faktorisierung von $n-1$ diejenige von $n+1$ benutzt.
b) Analog zur $(p-1)$-Methode gebe man eine $(p+1)$-Methode zur Berechnung eines nichttrivialen Teilers von n an.

Hinweis. Man benutze die zyklische Gruppe $\mathbb{F}_{p^2}^*/\mathbb{F}_p^*$ der Ordnung $p+1$.

7 References

[1] H. Cohen: A course in computational algebraic number theory, Graduate Texts in Mathematics 138, Springer, 1993.

[2] J. von zur Gathen, J. Gerhard: Modern computer algebra, second edition, Cambridge University Press, 2003.

[3] D. Knuth: The art of computer programming, vol. 1: fundamental algorithms, 2. printing of the 2. edition, Addison-Wesley Series in Computer Science and Information Processing, Addison-Wesley, 1975.

[4] D. Knuth: The art of computer programming, vol. 2: seminumerical algorithms, 2. edition, Addison-Wesley Series in Computer Science and Information Processing, Addison-Wesley, 1981.

[5] D. Knuth: The art of computer programming, vol. 3: sorting and searching, Addison-Wesley Series in Computer Science and Information Processing, Addison-Wesley, 1973.