# Primality testing in polynomial time

Michiel Smid*

May 28, 2003

**Abstract**

These notes contain a description and correctness proof of the deterministic polynomial-time primality testing algorithm of Agrawal, Kayal, and Saxena. Some background from number theory and algebra is given in Section 4.

## 1 A polynomial identity for prime numbers

**Theorem 1.1** *Let $n \geq 2$ and $a \geq 0$ be integers.*

1. *If $n$ is a prime number, then*

$$(x-a)^n = x^n - a$$

*in the ring $\mathbb{Z}_n[x]$.*

2. *If $gcd(a,n) = 1$ and $n$ is not a prime number, then*

$$(x-a)^n \neq x^n - a$$

*in the ring $\mathbb{Z}_n[x]$.*

*School of Computer Science, Carleton University, Ottawa, Ontario, Canada K1S 5B6. E-mail: michiel@scs.carleton.ca.

**Proof.** By Newton's binomial theorem, we have

$$(x-a)^n = \sum_{i=0}^{n} \binom{n}{i} x^i (-a)^{n-i}.$$

If $n$ is a prime number, then $\binom{n}{i} \equiv 0 \bmod n$ for $1 \leq i \leq n-1$, and $a^n \equiv a \bmod n$ (see Lemmas 4.1 and 4.2). Therefore, in $\mathbb{Z}_n[x]$,

$$(x-a)^n = (-a)^n + x^n = x^n - a^n = x^n - a,$$

proving the first assertion.

To prove the second assertion, assume that $gcd(a,n) = 1$ and $n$ is not a prime number. Let $q$ be a prime factor of $n$, and let $k \geq 1$ be the integer such that $q^k \mid n$ and $q^{k+1} \nmid n$.

Since $q \mid n$ and $gcd(a,n) = 1$, we have $gcd(a,q) = 1$. It follows that

$$gcd(a^{n-q}, q^k) = 1. \tag{1}$$

We claim that

$$q^k \nmid \binom{n}{q}. \tag{2}$$

The proof of this claim will be given later. The coefficient of $x^q$ in $(x-a)^n$ is equal to $\binom{n}{q}(-1)^{n-q}a^{n-q}$. Assume this coefficient is divisible by $n$. Then we can write

$$\binom{n}{q} a^{n-q} = \alpha n,$$

for some integer $\alpha$. Hence,

$$\frac{\binom{n}{q} a^{n-q}}{q^k} = \alpha \cdot n/q^k.$$

Since the right-hand side is an integer, the left-hand side is also an integer. But then (1) implies that $q^k$ divides $\binom{n}{q}$, contradicting (2). Hence, we have shown that the coefficient of $x^q$ in $(x-a)^n$ is non-zero modulo $n$. This proves that $(x-a)^n \neq x^n - a$ in the ring $\mathbb{Z}_n[x]$.

It remains to prove (2). The proof is by contradiction. So assume that $q^k \mid \binom{n}{q}$. Then $\binom{n}{q} = \alpha q^k$ for some positive integer $\alpha$, i.e.,

$$\frac{n(n-1)(n-2)\ldots(n-q+1)}{q!} = \alpha q^k.$$

We can rewrite this as

$$n = \frac{\alpha(q-1)!q^{k+1}}{(n-1)(n-2)\dots(n-q+1)}.$$

Observe that the right-hand side is an integer. Let $1 \leq j \leq q-1$, and assume that $q \mid (n-j)$. Then $n - j \equiv 0 \bmod q$. Since $n \equiv 0 \bmod q$, it follows that $j \equiv 0 \bmod q$, which is not true. Hence, $q \nmid (n-j)$ for all $1 \leq j \leq q-1$. Therefore, since $q$ is a prime number,

$$\frac{\alpha(q-1)!}{(n-1)(n-2)\dots(n-q+1)}$$

is an integer. But this implies that $q^{k+1} \mid n$, a contradiction. This completes the proof of (2). ∎

# 2 An improved polynomial identity

If we use Theorem 1.1 to test if $n$ is a prime number, then we have to compute the coefficients $(\bmod n)$ of the polynomial $(x-a)^n$, for one integer $a$ with $gcd(a,n) = 1$. Since this polynomial has $n+1$ terms, this will lead to an algorithm whose running time is exponential in the length of $n$. In this section, we present a more "efficient" polynomial identity that can be used to test if $n$ is the power of a prime number. The idea is to compare $(x-a)^n$ and $x^n - a$ modulo a polynomial of "low" degree, for only a "small" number of values for $a$. The following theorem formalizes this.

**Theorem 2.1** *Let $n \geq 2$ be an integer, and let $q$ and $r$ be prime numbers such that*

1. *$gcd(m,n) = 1$ for all $1 \leq m \leq r$,*
2. *$q$ divides $r-1$,*
3. *$q \geq 2\sqrt{r}\log n + 2$,*
4. *$n^{(r-1)/q} \not\equiv 1 \bmod r$, and*

5. *for all* $1 \leq a \leq \lfloor 2\sqrt{r}\log n \rfloor + 1$,

$$(x-a)^n = (x^n - a) \bmod (x^r - 1)$$

*in* $\mathbb{Z}_n[x]$.

*Then $n$ is the power of a prime number.*

In the rest of this section, we will prove this theorem. From now on, we assume that all assumptions in Theorem 2.1 hold.

**Lemma 2.2** *There is a prime factor $p$ of $n$ such that $q \mid d$, where $d$ is the multiplicative order of $p \bmod r$.*

**Proof.** Let $e$ be the multiplicative order of $n \bmod r$. Observe that $e$ exists, because $gcd(r, n) = 1$. We first prove that $q$ divides $e$. The proof is by contradiction. So assume that $q \nmid e$. By Fermat's theorem, $n^{r-1} \equiv 1 \bmod r$, which implies that $e \mid (r-1)$. Hence, we can write $r - 1 = \alpha e$, for some positive integer $\alpha$. Since $q$ divides $r-1$, it follows that $\alpha e/q$ is an integer. Then the facts that $q$ is a prime number and $q \nmid e$ imply that $q \mid \alpha$, which in turn implies that $e$ divides $(r-1)/q$. But then, $n^{(r-1)/q} \equiv 1 \bmod r$, which is a contradiction.

Consider the prime factorization $n = p_1^{k_1} \ldots p_m^{k_m}$ of $n$. For each $1 \leq i \leq m$, let $e_i$ be the multiplicative order of $p_i \bmod r$. We claim that $e$ divides the least common multiple of $e_1, \ldots, e_m$. To prove this, let $f$ be any common multiple of $e_1, \ldots, e_m$. Then, for each $1 \leq i \leq m$, we can write $f = \alpha_i e_i$, where $\alpha_i$ is a positive integer. Observe that

$$n^f \equiv p_1^{k_1 f} \ldots p_m^{k_m f} \equiv p_1^{k_1 \alpha_1 e_1} \ldots p_m^{k_m \alpha_m e_m} \equiv 1 \bmod r.$$

Therefore, $e$ divides $f$. Since $f$ was an arbitrary common multiple, it follows that the least common multiple of $e_1, \ldots, e_m$ is divisible by $e$.

Now let $f$ be as above and assume it is the least common multiple of $e_1, \ldots, e_m$. We have shown that $q \mid f$. We claim that there is an $1 \leq i \leq m$ such that $q \mid e_i$. Assume this is not the case. Then, since $q$ is a prime number, $q \mid \alpha_i$ for each $1 \leq i \leq m$. But then $f/q$ is a common multiple of $e_1, \ldots, e_m$. This contradicts our assumption that $f$ is the least common multiple of $e_1, \ldots, e_m$. Therefore, there is an $1 \leq i \leq m$ such that $q \mid e_i$. By choosing $p := p_i$ (and, hence, $d = e_i$), the proof is complete. ∎

From now on, we let $p$ be the prime number of Lemma 2.2. Furthermore, we define $\ell := \lfloor 2\sqrt{r}\log n \rfloor + 1$. Observe that Theorem 2.1 claims that $n$ is a power of $p$.

**Lemma 2.3** *For all $1 \leq a \leq \ell$, $i \geq 0$, and $j \geq 0$,*

$$(x-a)^{p^i n^j} = (x^{p^i n^j} - a) \bmod (x^r - 1)$$

*in $\mathbb{Z}_p[x]$.*

**Proof.** We fix $1 \leq a \leq \ell$. We know from the assumptions in Theorem 2.1 that $(x-a)^n = (x^n - a) \bmod (x^r - 1)$ in $\mathbb{Z}_n[x]$. Since $p$ divides $n$, this implies that

$$(x-a)^n = (x^n - a) \bmod (x^r - 1) \tag{3}$$

in $\mathbb{Z}_p[x]$. By Theorem 1.1, we have $(x-a)^p = x^p - a$ in $\mathbb{Z}_p[x]$. Therefore, we also have

$$(x-a)^p = (x^p - a) \bmod (x^r - 1) \tag{4}$$

in $\mathbb{Z}_p[x]$. Let $m$ and $m'$ be positive integers such that both equations

$$(x-a)^m = (x^m - a) \bmod (x^r - 1) \tag{5}$$

and

$$(x-a)^{m'} = (x^{m'} - a) \bmod (x^r - 1) \tag{6}$$

hold in $\mathbb{Z}_p[x]$. We claim that

$$(x-a)^{mm'} = (x^{mm'} - a) \bmod (x^r - 1) \tag{7}$$

in $\mathbb{Z}_p[x]$. This claim, together with (3) and (4), will prove the lemma. (Observe that the lemma obviously holds if $i = j = 0$.) So it remains to prove (7). If we replace $x$ by $x^m$ in (6), then we get

$$(x^m - a)^{m'} = (x^{mm'} - a) \bmod (x^{mr} - 1)$$

in $\mathbb{Z}_p[x]$. Since $x^r - 1$ divides $x^{mr} - 1$, it follows that

$$(x^m - a)^{m'} = (x^{mm'} - a) \bmod (x^r - 1)$$

in $\mathbb{Z}_p[x]$. By (5), we have

$$(x^m - a)^{m'} = (x-a)^{mm'} \bmod (x^r - 1)$$

in $\mathbb{Z}_p[x]$. ∎

**Lemma 2.4** *Let*

$$E := \{n^i p^j : 0 \leq i, j \leq \lfloor \sqrt{r} \rfloor\}.$$

*If $n$ is not a power of a prime number, then the set $E$ contains more than $r$ elements.*

**Proof.** We assume that $n$ is not a power of a prime number. Let $0 \leq i, j \leq \lfloor \sqrt{r} \rfloor$ and $0 \leq i', j' \leq \lfloor \sqrt{r} \rfloor$, and assume that $n^i p^j = n^{i'} p^{j'}$. We will prove that $(i, j) = (i', j')$. From this, it will follow that

$$|E| = (1 + \lfloor \sqrt{r} \rfloor)^2 > r.$$

Let $k \geq 1$ be such that $p^k \mid n$ and $p^{k+1} \nmid n$. Then $n = \alpha p^k$ for some integer $\alpha \geq 2$. Observe that $gcd(\alpha, p) = 1$. Since $n^i p^j = n^{i'} p^{j'}$, we have

$$\alpha^i p^{ik+j} = \alpha^{i'} p^{i'k+j'}.$$

Since $gcd(\alpha, p) = 1$, it follows that $\alpha^i = \alpha^{i'}$. Therefore, since $\alpha \geq 2$, we must have $i = i'$. This implies that $p^{ik+j} = p^{i'k+j'}$, from which we obtain $ik + j = i'k + j'$. Therefore $j = j'$. ∎

If $n$ is the power of a prime number, then Theorem 2.1 holds. So we assume from now on that $n$ is not a power of a prime number. Then, by Lemma 2.4, there are $0 \leq i, j \leq \lfloor \sqrt{r} \rfloor$ and $0 \leq i', j' \leq \lfloor \sqrt{r} \rfloor$, such that $(i, j) \neq (i', j')$ and

$$n^i p^j \equiv n^{i'} p^{j'} \bmod r.$$

Let $m := n^i p^j$ and $m' := n^{i'} p^{j'}$. We may assume without loss of generality that $m' \geq m$. Let $k$ be the nonnegative integer such that $m' = m + kr$. By Lemma 2.3, we have

$$(x - a)^{m'} = (x^{m'} - a) = (x^{m+kr} - a) \bmod (x^r - 1)$$

in $\mathbb{Z}_p[x]$, for all $1 \leq a \leq \ell$. Since $x^r = 1 \bmod (x^r - 1)$ in $\mathbb{Z}_p[x]$, it follows that

$$(x - a)^{m'} = (x^m - a) \bmod (x^r - 1)$$

in $\mathbb{Z}_p[x]$. We also know from Lemma 2.3 that

$$(x^m - a) = (x - a)^m \bmod (x^r - 1)$$

in $\mathbb{Z}_p[x]$. Therefore,

$$(x-a)^{m'} = (x-a)^m \bmod (x^r - 1)$$

in $\mathbb{Z}_p[x]$, for all $1 \leq a \leq \ell$.

We will prove below that $m = m'$. This will imply that $n^i p^j = n^{i'} p^{j'}$. As in the proof of Lemma 2.4, it follows that $i = i'$ and $j = j'$, which is a contradiction to our assumption that $n$ is not a power of a prime number. Therefore, Theorem 2.1 follows.

It remains to prove that $m = m'$. Let $h(x)$ be an irreducible polynomial in $\mathbb{Z}_p[x]$ that divides $(x^r - 1)/(x - 1)$. The degree of $h(x)$ is equal to $d$ (see Lemma 4.15; recall that $d$ is the multiplicative order of $p \bmod r$). Observe that

$$(x-a)^{m'} = (x-a)^m \bmod h(x) \tag{8}$$

in $\mathbb{Z}_p[x]$, for all $1 \leq a \leq \ell$. Define

$$S := \left\{ \prod_{a=1}^{\ell} (x-a)^{\alpha_a} : \alpha_a \in \{0,1\} \text{ for all } 1 \leq a \leq \ell \right\},$$

which we consider as a subset of $\mathbb{Z}_p[x]/(h(x))$. Consider the polynomial $G(z) = z^{m'} - z^m$ in the ring $(\mathbb{Z}_p[x]/(h(x)))[z]$. (Elements of this ring are polynomials in $z$ whose coefficients are elements of $\mathbb{Z}_p[x]/(h(x))$.) By (8), every element of $S$ is a root of $G(z)$. Since $m$ and $m'$ are both less than or equal to

$$n^{\lfloor \sqrt{r} \rfloor} p^{\lfloor \sqrt{r} \rfloor} \leq n^{2\sqrt{r}} < 2^\ell,$$

the degree of $G(z)$ is less than $2^\ell$. We will prove below that $S$ contains at least $2^\ell$ elements. Hence, if $m \neq m'$, the number of roots of $G(z)$ is larger than its degree. This is not possible, because $\mathbb{Z}_p[x]/(h(x))$ is a field (see Lemmas 4.11 and 4.12). Therefore, $m$ and $m'$ must be equal.

So it remains to prove that $|S| \geq 2^\ell$. We first claim that $p \geq \ell + 1$. Assume that $p \leq \ell$. Then

$$\ell = \lfloor 2\sqrt{r} \log n \rfloor + 1 < 2\sqrt{r} \log n + 2 \leq q < r$$

and, therefore, $p < r$. This contradicts the first assumption in Theorem 2.1. Hence, $p \geq \ell + 1$. This implies that the polynomials $x - a$, $1 \leq a \leq \ell$, are pairwise distinct elements of $\mathbb{Z}_p[x]$; each of them is obviously irreducible. Since

the factorization of any polynomial into irreducible polynomials is unique in $\mathbb{Z}_p[x]$ (see Lemma 4.12), the $2^\ell$ polynomials

$$\prod_{a=1}^{\ell}(x-a)^{\alpha_a}$$

where $\alpha_a \in \{0,1\}$ for all $1 \leq a \leq \ell$, are pairwise distinct in $\mathbb{Z}_p[x]$. Observe that the degree of each of these polynomials is less than or equal to $\ell$.

Recall that the degree of $h(x)$ is equal to $d$, that $q \mid d$, and $q \geq 2\sqrt{r}\log n + 2$. It follows that

$$d \geq q \geq 2\sqrt{r}\log n + 2 > \ell.$$

Therefore, the $2^\ell$ polynomials

$$\prod_{a=1}^{\ell}(x-a)^{\alpha_a}$$

where $\alpha_a \in \{0,1\}$ for all $1 \leq a \leq \ell$, are pairwise distinct in $\mathbb{Z}_p[x]/(h(x))$. This proves that the set $S$ contains at least $2^\ell$ elements. Hence, the proof of Theorem 2.1 is complete.

# 3 The primality testing algorithm

We now show that Theorem 2.1 leads to a polynomial-time primality testing algorithm. Of course, the main problem is to find the prime number $r$. (Observe that the prime number $q$ in Theorem 2.1 is the largest prime factor of $r-1$.) The algorithm, which we denote by *prime*$(n)$, is given in Figure 1. First, we prove the correctness of this algorithm. Then, we analyze its running time. We start by proving that the while-loop makes $O(\log^6 n)$ iterations.

**Lemma 3.1** *There exist constants $c_2 > c_1 > 0$ such that for each sufficiently large integer $n$, there exists a prime number $r$ such that*

1. $c_1 \log^6 n < r \leq c_2 \log^6 n$,
2. $r-1$ *has a prime factor $q$ with* $q \geq 2\sqrt{r}\log n + 2$, *and*
3. $n^{(r-1)/q} \not\equiv 1 \bmod r$.

**Algorithm** *prime*$(n)$
**Input:** Integer $n \geq 2$.
**Output:** *YES* if $n$ is a prime number, and *NO* otherwise.
$r := 2$;
*found* := *false*;
**while** $r < n$ and *found* = *false*
**do if** $gcd(r, n) \neq 1$ **then** return *NO* **endif**;
    **if** $r$ is a prime number and $r > 2$
    **then** $q :=$ largest prime factor of $r - 1$;
        **if** $q \geq 2\sqrt{r} \log n + 2$ and $n^{(r-1)/q} \not\equiv 1 \bmod r$
        **then** *found* := *true*
        **endif**
    **endif**;
    **if** *found* = *false* **then** $r := r + 1$ **endif**
**endwhile**;
**for** $a := 1$ **to** $\lfloor 2\sqrt{r} \log n \rfloor + 1$
**do if** $(x - a)^n \neq (x^n - a) \bmod (x^r - 1)$ in $\mathbb{Z}_n[x]$
    **then** return *NO*
    **endif**
**endfor**;
**if** $n = a^b$ for some integers $a, b \geq 2$
**then** return *NO*
**else** return *YES*
**endif**

Figure 1: *The primality testing algorithm.*

---

**Proof.** Let $\pi(x)$ denote the number of prime numbers that are less than or equal to $x$. Let $\pi'(x)$ denote the number of prime numbers $r$ such that $r \leq x$ and $r - 1$ has a prime factor that is greater than or equal to $r^{2/3}$. For all sufficiently large real numbers $x$, we have

$$\pi(x) \leq 5x/\log x$$

and

$$\pi'(x) \geq cx/\log x,$$

where $c$ is a constant; see Theorems 4.6 and 4.9.

Let $c_1 := 3^6$ and choose $c_2 > c_1$ such that $c_3 := cc_2/7 - 5c_1/6 > 0$. Let $n$ be a sufficiently large integer. Let $S$ denote the number of prime numbers $r$ such that $c_1 \log^6 n < r \leq c_2 \log^6 n$ and $r-1$ has a prime factor that is greater than or equal to $r^{2/3}$. Then

$$\begin{aligned} S &= \pi'(c_2 \log^6 n) - \pi'(c_1 \log^6 n) \\ &\geq \pi'(c_2 \log^6 n) - \pi(c_1 \log^6 n) \\ &\geq \frac{cc_2 \log^6 n}{\log c_2 + 6 \log\log n} - \frac{5c_1 \log^6 n}{\log c_1 + 6 \log\log n} \\ &\geq \frac{cc_2 \log^6 n}{7 \log\log n} - \frac{5c_1 \log^6 n}{6 \log\log n} \\ &= c_3 \frac{\log^6 n}{\log\log n}. \end{aligned}$$

Let $x := c_2 \log^6 n$ and

$$N := \prod_{i=1}^{\lfloor x^{1/3} \rfloor} (n^i - 1).$$

Since any positive integer $m$ has at most $\log m$ prime factors, the $i$-th term in the product defining $N$ has at most $i \log n \leq x^{1/3} \log n$ prime factors. Hence, the integer $N$ has at most $x^{2/3} \log n$ prime factors. Since, assuming that $n$ is sufficiently large,

$$x^{2/3} \log n = c_2^{2/3} \log^5 n < c_3 \frac{\log^6 n}{\log\log n},$$

there is a prime number $r$ such that $c_1 \log^6 n < r \leq c_2 \log^6 n$, $r-1$ has a prime factor $q$ that is greater than or equal to $r^{2/3}$, and $r \nmid N$.

Observe that $r^{2/3} = r^{1/6}\sqrt{r}$ and $r^{1/6} > c_1^{1/6} \log n = 3 \log n$. Therefore, we have $r^{2/3} > 3\sqrt{r} \log n$. It follows that

$$q \geq r^{2/3} > 3\sqrt{r} \log n \geq 2\sqrt{r} \log n + 2.$$

It remains to prove that $n^{(r-1)/q} \not\equiv 1 \bmod r$. Assume that $n^{(r-1)/q} \equiv 1 \bmod r$. Then $r$ divides $n^{(r-1)/q} - 1$. Since $q \geq r^{2/3}$, $r \leq c_2 \log^6 n$, and $x = c_2 \log^6 n$, we have

$$\frac{r-1}{q} < \frac{r}{r^{2/3}} = r^{1/3} \leq (c_2 \log^6 n)^{1/3} = x^{1/3}.$$

But this implies that $r \mid N$, which is a contradiction. ∎

**Lemma 3.2** *Algorithm prime($n$) is correct.*

**Proof.** We first assume that $n$ is a prime number. Since $gcd(r, n) = 1$ for all $2 \leq r < n$, the algorithm does not return *NO* during the while-loop. Consider the integer $r$ after the while-loop has been completed. Let $1 \leq a \leq \lfloor 2\sqrt{r} \log n \rfloor + 1$. By Theorem 1.1, $(x-a)^n = x^n - a$ in $\mathbb{Z}_n[x]$, which implies that

$$(x-a)^n = (x^n - a) \bmod (x^r - 1)$$

in $\mathbb{Z}_n[x]$. Therefore, the algorithm does not return *NO* during the for-loop. Finally, since $n$ is not of the form $a^b$ with $a, b \geq 2$, the algorithm does not return *NO* in the last if-then-else statement. Hence, the algorithm returns *YES*.

If $n$ is not a prime number, then it follows from Theorem 2.1 that the algorithm returns *NO*. ∎

**Lemma 3.3** *Algorithm prime($n$) can be implemented such that its running time is $O(\log^{19} n)$.*

**Proof.** Consider one iteration of the while-loop. By Lemma 4.16, the *gcd*-computation takes $O(\log^3 n)$ time. By Lemma 4.20, we can decide in $O(\sqrt{r} \log^2 r)$ time whether $r$ is a prime number. By Lemma 4.21, the largest prime factor $q$ of $r-1$ can be computed in $O(\sqrt{r} \log^2 r)$ time. By Lemma 4.18, we can compute $n^{(r-1)/q} \bmod r$ in $O(\log^2 n + \log^3 r)$ time. It follows that one iteration of the while-loop takes $O(\log^3 n + \sqrt{r} \log^2 r)$ time. Since there are $O(\log^6 n)$ iterations (see Lemma 3.1), the entire while-loop takes $O(\log^9 n + \sqrt{r} \log^2 r \log^6 n)$ time. Since $r = O(\log^6 n)$, this is $O(\log^9 n (\log \log n)^2)$.

Next, we consider one iteration of the for-loop. By Lemma 4.19, it takes $O(r^2 \log^3 n)$ time to compute the coefficients of the polynomial $(x-a)^n \bmod (x^r - 1)$ in $\mathbb{Z}_n[x]$. The polynomial $(x^n - a) \bmod (x^r - 1)$ can be computed within the same time bound. Hence, the entire for-loop takes $O(r^2 \sqrt{r} \log^4 n)$ time. Since $r = O(\log^6 n)$, this is $O(\log^{19} n)$. Finally, by Lemma 4.29, testing if $n$ is a perfect power takes $O(\log^4 n \log \log n)$ time. ∎

We remark that using more advanced algorithms for integer multiplication and the Fast Fourier Transform to multiply polynomials, the running time

of the algorithm can be improved to $O(\log^{12} n(\log\log n)^c)$, for some constant $c$. We summarize our result in the following theorem.

**Theorem 3.4** *There is an algorithm that decides in polynomial time whether an arbitrary given positive integer is a prime number.*

# 4 Background

## 4.1 Number theory

**Lemma 4.1** *Let $p$ be a prime number and let $1 \leq i \leq p-1$. Then*

$$\binom{p}{i} \equiv 0 \bmod p.$$

**Proof.** Observe that $\binom{p}{i}$ is an integer, and

$$\binom{p}{i} = \frac{p(p-1)(p-2)\ldots(p-i+1)}{i!}.$$

Since $gcd(p, i!) = 1$, it follows that

$$\frac{(p-1)(p-2)\ldots(p-i+1)}{i!}$$

is an integer. ∎

**Lemma 4.2** *Let $p$ be a prime number. Then $a^p \equiv a \bmod p$ for all integers $a$.*

**Proof.** Since $(-a)^p \equiv -a^p \bmod p$ for all integers $a$, it suffices to prove the lemma for all nonnegative integers $a$. The proof is by induction on $a$. For $a = 0$, the claim clearly holds. Let $a \geq 0$ and assume that $a^p \equiv a \bmod p$. We know from Newton's binomial theorem that

$$(a+1)^p = \sum_{i=0}^{p} \binom{p}{i} a^i.$$

Reducing this summation modulo $p$, and applying Lemma 4.1, it follows that

$$(a+1)^p \equiv a^p + 1 \equiv a + 1 \bmod p.$$

∎

**Theorem 4.3 (Fermat)** *Let $p$ be a prime number and let $a$ be an integer such that $p \nmid a$. Then*

$$a^{p-1} \equiv 1 \bmod p.$$

**Proof.** By Lemma 4.2, $a^p \equiv a \bmod p$. We can rewrite this as $a(a^{p-1} - 1) \equiv 0 \bmod p$. Since $a \not\equiv 0 \bmod p$, we must have $a^{p-1} \equiv 1 \bmod p$. ∎

**Lemma 4.4** *Let $n \geq 2$, $k \geq 1$, and $d \geq 1$ be integers such that $(n^k - 1) \mid (n^d - 1)$. Then $k \mid d$.*

**Proof.** Write $d = qk + r$, where $q$ and $r$ are integers with $q \geq 1$ and $0 \leq r < k$. Observe that

$$\begin{aligned} \frac{n^d - 1}{n^k - 1} &= \frac{n^r(n^{qk} - 1) + (n^r - 1)}{n^k - 1} \\ &= n^r \left(1 + n^k + n^{2k} + \ldots + n^{(q-1)k}\right) + \frac{n^r - 1}{n^k - 1}. \end{aligned}$$

Hence, $(n^r - 1)/(n^k - 1)$ is an integer. If $1 \leq r < k$, then $0 < (n^r - 1)/(n^k - 1) < 1$, which is clearly not possible. Therefore, $r = 0$. ∎

## 4.2 A weak version of the prime number theorem

For any real number $x$, let $\pi(x)$ denote the number of prime numbers that are less than or equal to $x$. It is known that

$$\lim_{x \to \infty} \frac{\pi(x)}{x/\ln x} = 1.$$

We will prove that

$$\pi(n) = O(n/\log n),$$

which is sufficient for our purposes. The proof given here is due to Erdős[1].

**Lemma 4.5** *For any positive integer $n$,*

$$\prod_{p \leq n} p \leq 4^n,$$

*where the product is taken over all prime numbers $p \leq n$.*

[1] I read the proof in an article by Pach in the Mathematical Intelligencer.

**Proof.** The proof is by induction on $n$. For $1 \leq n \leq 4$, the claim is easy to verify. Let $n \geq 5$ and assume the claim holds for all positive integers less than $n$. In the rest of this proof, all products are over prime numbers $p$. First, we assume that $n$ is odd. We have

$$\prod_{p \leq n} p = \left( \prod_{p \leq (n+1)/2} p \right) \cdot \left( \prod_{(n+3)/2 \leq p \leq n} p \right).$$

By the induction hypothesis, the first product on the right-hand side is less than or equal to $4^{(n+1)/2}$. We claim that

$$\left( \prod_{(n+3)/2 \leq p \leq n} p \right) \leq \binom{n}{(n+1)/2}.$$

This inequality holds, because

$$\binom{n}{(n+1)/2} = \frac{\frac{n+3}{2} \frac{n+5}{2} \ldots n}{\left(\frac{n-1}{2}\right)!},$$

which is an integer that is divisible by every prime number $p$ with $(n+3)/2 \leq p \leq n$. Since

$$\binom{n}{(n+1)/2} = \frac{1}{2} \left( \binom{n}{(n+1)/2} + \binom{n}{(n-1)/2} \right) \leq \frac{1}{2} 2^n = 2^{n-1},$$

it follows that

$$\prod_{p \leq n} p \leq 4^{(n+1)/2} \, 2^{n-1} = 4^n.$$

If $n$ is even, then we observe that $n$ is not a prime number. Therefore,

$$\prod_{p \leq n} p = \prod_{p \leq n-1} p \leq 4^{n-1} \leq 4^n.$$

■

**Theorem 4.6** *For all sufficiently large integers $n$,*

$$\pi(n) \leq 5n / \log n.$$

**Proof.** In this proof, all sums and products are over prime numbers $p$. We first observe that

$$\begin{aligned}\sum_{p\le n}\log p &\ge \sum_{\sqrt{n}<p\le n}\log p\\ &\ge \sum_{\sqrt{n}<p\le n}\log\sqrt{n}\\ &= \bigl(\pi(n)-\pi(\sqrt{n})\bigr)\log\sqrt{n}.\end{aligned}$$

By Lemma 4.5,

$$\sum_{p\le n}\log p=\log\left(\prod_{p\le n}p\right)\le 2n.$$

Therefore,

$$\pi(n)\le\pi(\sqrt{n})+2n/\log\sqrt{n}=\pi(\sqrt{n})+4n/\log n.$$

The proof is completed by observing that $\pi(\sqrt{n})\le\sqrt{n}\le n/\log n$ for all sufficiently large integers $n$. ∎

We next prove that

$$\pi(n)=\Omega(n/\log n).$$

The proof is due to Nair [9]. For any positive integer $n$, let $d_n$ denote the least common multiple of the integers $1,2,\ldots,n$.

**Lemma 4.7** *For any positive integer $n$, $d_n\ge 2^{n-2}$.*

**Proof.** Let $m$ be a positive integer and define $I:=\int_0^1 x^m(1-x)^m\,dx$. Since $0\le x(1-x)\le 1/4$ for all $x$ with $0\le x\le 1$, we have

$$0\le I\le(1/4)^m.$$

Next we observe that

$$\begin{aligned}I &= \int_0^1 x^m\sum_{k=0}^{m}(-1)^k\binom{m}{k}x^k\,dx\\ &= \sum_{k=0}^{m}(-1)^k\binom{m}{k}\int_0^1 x^{m+k}\,dx\\ &= \sum_{k=0}^{m}(-1)^k\binom{m}{k}\frac{1}{m+k+1}.\end{aligned}$$

The latter summation can be written as $A/d_{2m+1}$ for some integer $A \geq 1$. Hence, we have

$$d_{2m+1} = A/I \geq 4^m.$$

Observe that the inequality $d_{2m+1} \geq 4^m$ also holds if $m = 0$.

If $n$ is odd, then we can write $n = 2m + 1$ for some integer $m \geq 0$. We obtain

$$d_n = d_{2m+1} \geq 4^m = 2^{n-1}.$$

If $n$ is even, then we obtain

$$d_n \geq d_{n-1} \geq 2^{n-2}.$$

■

**Theorem 4.8** *For all positive integers $n$,*

$$\pi(n) \geq (n-2)/\log n.$$

**Proof.** Let $p_1, p_2, \ldots, p_k$ be all prime numbers that are less than or equal to $n$. We can write each integer $m$ with $1 \leq m \leq n$ as

$$m = \prod_{i=1}^{k} p_i^{\alpha_{mi}},$$

where each $\alpha_{mi}$, $1 \leq i \leq k$, is a nonnegative integer. Hence, the least common multiple $d_n$ of $1, 2, \ldots, n$ is given by

$$d_n = \prod_{i=1}^{k} p_i^{\max(\alpha_{1i}, \ldots, \alpha_{ni})}.$$

Observe that

$$p_i^{\max(\alpha_{1i}, \ldots, \alpha_{ni})} \leq n$$

for each $i$ with $1 \leq i \leq k$. Therefore, we have

$$d_n \leq \prod_{i=1}^{k} n = n^{\pi(n)}.$$

By Lemma 4.7, we have $d_n \geq 2^{n-2}$. It follows that

$$2^{n-2} \leq n^{\pi(n)},$$

from which the claim follows. ∎

The proof of the following theorem can be found in Fouvry [7].

**Theorem 4.9** *Let $\pi'(x)$ denote the number of prime numbers $p$ such that $p \leq x$ and $p-1$ has a prime factor that is greater than or equal to $p^{2/3}$. There exists a constant $c > 0$ such that*

$$\pi'(x) \geq cx/\log x$$

*for all sufficiently large real numbers $x$.*

## 4.3 Group theory

**Lemma 4.10** *Let $G$ be a finite multiplicative group with unit-element 1, let $n$ be the size of $G$, and let $a \in G$. Then $a^n = 1$.*

**Proof.** Let $d$ be the order of $a$ in $G$ and let $H := \{1, a, a^2, \ldots a^{d-1}\}$. We claim that $H$ is a subgroup of $G$. Assuming this claim is true, it follows from Lagrange's theorem that $|G| = n$ is a multiple of $|H| = d$. Hence, we can write $n = kd$ for some positive integer $k$. This implies that $a^n = (a^d)^k = 1$. So it remains to prove that $H$ is a subgroup of $G$.

First, we prove that $H$ is closed under multiplication. Let $0 \leq i < j \leq d-1$. Write $i + j = qd + r$, where $q$ is a nonnegative integer, and $r$ is an integer such that $0 \leq r < d$. Then $a^i a^j = a^{i+j} = (a^d)^q a^r = a^r$. Since $0 \leq r < d$, $a^r$ is an element of $H$. Hence, $a^i a^j$ is an element of $H$.

Next, we prove that the inverse of any element of $H$ is contained in $H$. Let $0 \leq i \leq d-1$, and let $g \in G$ be the inverse of $a^i$. Hence, $ga^i = 1$. If $i = 0$, then $g = 1$, which is contained in $H$. So we may assume that $1 \leq i \leq d-1$. Since $1 = a^d = a^{d-i+i} = a^{d-i}a^i$, it follows that $g = a^{d-i}$. Since $1 \leq d - i \leq d - 1$, it follows that $g \in H$. ∎

We can use Lemma 4.10 to give an alternative proof of Theorem 4.3. So let $p$ be a prime number and let $a$ be an integer such that $p \nmid a$. Let $1 \leq b \leq p-1$ be such that $a \equiv b \bmod p$. Let $G$ be the multiplicative group $(\mathbb{Z}_p)^*$. Then $G$ contains $p-1$ elements, $b$ being one of them. By Lemma 4.10, $b^{p-1} \equiv 1 \bmod p$. It follows that $a^{p-1} \equiv 1 \bmod p$.

## 4.4 Finite fields

Let $p$ be a prime number and let $d \geq 1$ be an integer. The ring $\mathbb{Z}_p$ is a field having $p$ elements. Let $h(x)$ be an irreducible polynomial of degree $d$ in the ring $\mathbb{Z}_p[x]$. That is, there are no polynomials $a(x)$ and $b(x)$ in $\mathbb{Z}_p[x]$ that both have a degree that is less than $d$ and for which $a(x)b(x) = h(x)$ in $\mathbb{Z}_p[x]$.

The residue class ring $\mathbb{Z}_p[x]/(h(x))$ consists of all polynomials in $\mathbb{Z}_p[x]$ of degree less than $d$, where addition and multiplication are done modulo $h(x)$.

**Lemma 4.11** *The residue class ring $F := \mathbb{Z}_p[x]/(h(x))$ is a field with $p^d$ elements.*

**Proof.** Since there are exactly $p^d$ polynomials in $\mathbb{Z}_p[x]$ of degree less than $d$, it is clear that $F$ has size $p^d$. We will only prove that each non-zero polynomial $f(x)$ in $F$ has a multiplicative inverse in $F$. Since the degree of $f(x)$ is less than $d$, and since $h(x)$ is irreducible, we have $gcd(f(x), h(x)) = 1$. Using the extended Euclidean algorithm, we obtain two polynomials $a(x)$ and $b(x)$ in $\mathbb{Z}_p[x]$ such that

$$a(x)f(x) + b(x)h(x) = 1$$

holds in $\mathbb{Z}_p[x]$. Hence, $a(x) \bmod h(x)$ is the multiplicative inverse of $f(x)$. ∎

**Lemma 4.12** *Let $F$ be a field, let $f(x)$ be a polynomial in $F[x]$, and let $d$ be the degree of $f(x)$. Assume that $d \geq 1$.*

- *There are at most $d$ elements $\alpha \in F$ such that $f(\alpha) = 0$.*
- *There are unique irreducible polynomials $g_1(x), \ldots, g_m(x)$ in $F[x]$, for some positive integer $m$, such that $f(x) = \prod_{i=1}^{m} g_i(x)$.*

**Proof.** See the book by Lidl and Niederreiter [8]. ∎

Lemma 4.12 does not hold if $F$ is a ring. For example, in $\mathbb{Z}_9[x]$, the polynomial $f(x) = x^2$ has three roots $\alpha = 0, 3$, and 6, and $f(x) = xx = (x-3)(x-6)$.

**Lemma 4.13** *The multiplicative group $F^*$ of any finite field $F$ is cyclic.*

**Proof.** Let $q$ denote the number of elements of $F$. We may assume that $q \geq 3$. Let $h := q - 1$ and let $h = p_1^{r_1} \ldots p_m^{r_m}$ be the prime factorization of $h$.

For each $1 \leq i \leq m$, the polynomial $x^{h/p_i} - 1$ has at most $h/p_i$ roots in $F$; see Lemma 4.12. Hence, since $h/p_i < h$, there exists an element $a_i$ in $F^*$ such that $a_i^{h/p_i} \neq 1$. Let $b_i := a_i^{h/p_i^{r_i}}$. Since, by Lemma 4.10, $a_i^h = 1$, we have $b_i^{p_i^{r_i}} = a_i^h = 1$. Let $e_i$ be the multiplicative order of $b_i$ in $F$. Then $e_i \mid p_i^{r_i}$. Since $p_i$ is a prime number, we have $e_i = p_i^{s_i}$ for some $0 \leq s_i \leq r_i$. Assume that $s_i \leq r_i - 1$. Then $b_i^{p_i^{r_i - 1}} = 1$. On the other hand, we have $b_i^{p_i^{r_i - 1}} = a_i^{h/p_i} \neq 1$. Therefore, $s_i = r_i$ and $e_i = p_i^{r_i}$.

Let $b := b_1 b_2 \ldots b_m$, and let $e$ be the multiplicative order of $b$ in $F$. We claim that $b$ is a generator of $F^*$, i.e., that $e = h$. Assume that this is not the case. Since $b^h = 1$, we have $e \mid h$. Since $e$ is a proper divisor of $h$, there is an index $1 \leq i \leq m$ such that $e$ divides $h/p_i$. We may assume w.l.o.g. that $i = 1$. Since $b^e = 1$, we have $b^{h/p_1} = 1$. Observe that

$$b^{h/p_1} = b_1^{h/p_1} b_2^{h/p_1} \ldots b_m^{h/p_1}.$$

Consider any $2 \leq j \leq m$. Then $p_j^{r_j}$ divides $h/p_1$. Since $b_j^{p_j^{r_j}} = 1$, it follows that $b_j^{h/p_1} = 1$. Hence,

$$1 = b^{h/p_1} = b_1^{h/p_1} b_2^{h/p_1} \ldots b_m^{h/p_1} = b_1^{h/p_1}.$$

It follows that $e_1$ divides $h/p_1$. Since $e_1 = p_1^{r_1}$ and $h/p_1 = p_1^{r_1 - 1} p_2^{r_2} \ldots p_m^{r_m}$, this is a contradiction. ∎

**Lemma 4.14** *Let $p$ be a prime number and let $f(x)$ be a polynomial in $\mathbb{Z}_p[x]$. Then the equation*

$$f(x^p) = (f(x))^p$$

*holds in $\mathbb{Z}_p[x]$.*

**Proof.** The proof is by induction on the degree $d$ of $f(x)$. If $d = 0$, then $f(x) = a$ for some $a \in \mathbb{Z}_p$. The claim follows from Lemma 4.2.

Let $d \geq 1$, and assume the claim is true for all polynomials in $\mathbb{Z}_p[x]$ having degree less than $d$. We can write $f(x) = ax^d + g(x)$, where $a \in \mathbb{Z}_p$ and $g(x)$

is a polynomial in $\mathbb{Z}_p[x]$ of degree less than $d$. We have

$$\begin{aligned}(f(x))^p &= (ax^d + g(x))^p \\ &= \sum_{i=0}^{p} \binom{p}{i} a^i x^{id} (g(x))^{p-i} \\ &= (g(x))^p + a^p x^{pd},\end{aligned}$$

where the last equality follows from Lemma 4.1. Since $a^p \equiv a \bmod p$ and, by the induction hypothesis, $(g(x))^p = g(x^p)$, it follows that

$$(f(x))^p = g(x^p) + a(x^p)^d = f(x^p).$$

■

**Lemma 4.15** *Let $p$ and $r$ be two distinct prime numbers, and let $d$ be the multiplicative order of $p \bmod r$. Every irreducible polynomial in $\mathbb{Z}_p[x]$ that divides $(x^r - 1)/(x - 1)$ has degree $d$.*

**Proof.** Let $h(x)$ be an irreducible polynomial in $\mathbb{Z}_p[x]$ that divides $(x^r - 1)/(x-1)$, and let $k$ be the degree of $h(x)$. We will show that $k \mid d$ and $d \mid k$. This will prove that $k = d$.

Since $h(x)$ is irreducible, $\mathbb{Z}_p[x]/(h(x))$ is a field of size $p^k$. Let $g(x)$ be a generator of the multiplicative group of $\mathbb{Z}_p[x]/(h(x))$. Hence, the multiplicative order of $g(x)$ in this group is equal to $p^k - 1$. By Lemma 4.14, we have $(g(x))^p = g(x^p)$ in $\mathbb{Z}_p[x]$. Applying this equation repeatedly yields

$$(g(x))^{p^d} = g(x^{p^d})$$

in $\mathbb{Z}_p[x]$. Since $p^d \equiv 1 \bmod r$, there is an integer $k \geq 1$ such that $p^d = 1 + kr$. Let $f(x)$ be the polynomial in $\mathbb{Z}_p[x]$ such that $f(x)h(x) = x^r - 1$. Then the following holds in $\mathbb{Z}_p[x]$:

$$x^{p^d} = xx^{kr} = x(1 + f(x)h(x))^k = x \bmod h(x).$$

Hence, we obtain the following equation in $\mathbb{Z}_p[x]$:

$$(g(x))^{p^d} = g(x) \bmod h(x).$$

Since $g(x)$ is a generator, $g(x) \neq 1 \bmod h(x)$. Hence, $g(x)$ has an inverse in the field $\mathbb{Z}_p[x]/(h(x))$, which implies that

$$(g(x))^{p^d-1} = 1 \bmod h(x).$$

Since the multiplicative order of $g(x)$ is equal to $p^k-1$, it follows that $(p^k-1) \mid (p^d-1)$. Then, by Lemma 4.4, $k \mid d$.

Since $h(x)$ divides $x^r - 1$ in $\mathbb{Z}_p[x]$, we have $x^r = 1 \bmod h(x)$. This, together with the fact that $r$ is a prime number, implies that, in the field $\mathbb{Z}_p[x]/(h(x))$, the order of the polynomial $x$ is equal to $r$. On the other hand, since the multiplicative group of $\mathbb{Z}_p[x]/(h(x))$ has size $p^k - 1$, we have (by Lemma 4.10) $x^{p^k-1} = 1 \bmod h(x)$. Then it follows that $r \mid (p^k - 1)$ or, equivalently, $p^k \equiv 1 \bmod r$. Since be the multiplicative order of $p \bmod r$ is equal to $d$, it follows that $d \mid k$. ∎

## 4.5 Algorithms in number theory

In this section, we present some number-theoretic algorithms. The running time of algorithms will be expressed as a function of the total number of bits in the input; time refers to the number of bit operations made by the algorithm.

### 4.5.1 Some elementary results

We start by considering some elementary problems, such as integer multiplication, division, exponentiation, and computing the largest prime factor of an integer. For each one, we show how they can be solved, even though we do not give the fastest known algorithm. For the purpose of these notes, the results in this section are sufficient.

**Lemma 4.16** *Let $a$, $b$, and $n$ be integers such that $1 \leq a \leq b \leq n$.*

1. *The product $ab$ can be computed in $O(\log^2 n)$ time.*
2. *The integer $\lfloor b/a \rfloor$ can be computed in $O(\log^2 n)$ time.*
3. *The integer $b \bmod a$ can be computed in $O(\log^2 n)$ time.*
4. *The greatest common divisor of $a$ and $b$ can be computed in $O(\log^3 n)$ time.*

**Proof.** The first three claims follow by using the "school method" to compute $ab$, $\lfloor b/a \rfloor$, and $b \bmod a$. A proof of the third claim can be found in the book by Cormen, Leiserson, Rivest, and Stein [5]. ∎

**Lemma 4.17** *Let $a \geq 2$ and $b \geq 1$ be integers. The power $n := a^b$ can be computed in $O(\log^2 n \log\log n)$ time.*

**Proof.** The following algorithm computes $a^b$, for any two positive integers $a$ and $b$.

**Input:** Positive integers $a$ and $b$.
**Output:** $a^b$.
$x := a$; $y := b$; $z := 1$;
**while** $y \neq 0$
**do if** $y$ is even
  **then** $y := y/2$; $x := x^2$
  **else** $y := y - 1$; $z := zx$
  **endif**
**endwhile**;
return $z$

The correctness of the algorithm follows from the fact that it maintains the invariant $zx^y = a^b$. Assume that $a \geq 2$. The number of iterations of the while-loop is $O(\log b)$, which is $O(\log\log n)$, because $n = a^b \geq 2^b$. Since, by Lemma 4.16, one iteration takes $O(\log^2 n)$ time, the entire algorithm takes $O(\log^2 n \log\log n)$ time. ∎

**Lemma 4.18** *Let $a \geq 2$, $b \geq 1$, and $r \geq 2$ be integers. The value of $a^b \bmod r$ can be computed in $O(\log^2 n + \log b \log^2 r)$ time, where $n = \max(a, r)$.*

**Proof.** We use the same algorithm as in the proof of Lemma 4.17, except that we initialize $x$ as $a \bmod r$, and in the while-loop, we update $x$ and $z$ as $x := x^2 \bmod r$ and $z := zx \bmod r$, respectively. Hence, at any moment during the algorithm, $x$ and $z$ are integers in the range $[0, r-1]$. By Lemma 4.16, the initialization of $x$ takes $O(\log^2 n)$ time, where $n = \max(a, r)$. The number of iterations of the while-loop is $O(\log b)$ and, by Lemma 4.16, each one takes $O(\log^2 r)$ time. ∎

**Lemma 4.19** *Let $n$, $r$, and $a$ be integers with $2 \leq r < n$ and $1 \leq a < n$. The $r$ coefficients of the polynomial $(x-a)^n \bmod (x^r - 1)$ in $\mathbb{Z}_n[x]$ can be computed in $O(r^2 \log^3 n)$ time.*

**Proof.** The algorithm is as follows.

**Input:** Integers $n$, $r$, and $a$ with $2 \leq r < n$ and $1 \leq a < n$.
**Output:** All coefficients of the polynomial $(x-a)^n \bmod (x^r - 1)$ in the ring $\mathbb{Z}_n[x]$.
$f(x) := 1$; $g(x) := x - a$; $y := n$;
**while** $y \neq 0$
**do if** $y$ is even
  **then** $y := y/2$; $h(x) := g(x)g(x)$;
    $g(x) := h(x) \bmod (x^r - 1)$
  **else** $y := y - 1$; $h(x) := f(x)g(x)$;
    $f(x) := h(x) \bmod (x^r - 1)$
  **endif**
**endwhile**;
return $f(x)$

It is assumed that in all operations involving $f(x)$, $g(x)$, and $h(x)$, the coefficients are reduced modulo $n$. The algorithm maintains the invariant

$$f(x)g(x)^y = (x-a)^n \bmod (x^r - 1) \text{ in } \mathbb{Z}_n[x].$$

First observe that at any moment during the algorithm, the degree of both $f(x)$ and $g(x)$ is less than or equal to $r-1$. Therefore, the degree of $h(x)$ is always less than or equal to $2r-2$.

The assignment $h(x) := g(x)g(x)$ takes $O(r^2)$ multiplications mod $n$. Each such multiplication is on two integers in the range $[0, n-1]$. Hence, by Lemma 4.16, $h(x)$ can be computed in $O(r^2 \log^2 n)$ time. The same time bound holds for the assignment $h(x) := f(x)g(x)$.

We can write $h(x)$ as $h(x) = \sum_{i=0}^{2r-2} h_i x^i$, where each coefficient $h_i$ is an integer in the range $[0, n-1]$. Then

$$h(x) \bmod (x^r - 1) = \sum_{i=0}^{r-2} ((h_i + h_{r+i}) \bmod n) x^i + h_{r-1} x^{r-1}.$$

Hence, reducing $h(x)$ modulo $x^r - 1$ takes $O(r \log n)$ time.

Since the algorithm makes $O(\log n)$ iterations, the proof is complete. ∎

**Lemma 4.20** *Let $n \geq 2$ be an integer. There is an algorithm that decides in $O(\sqrt{n}\log^2 n)$ time whether $n$ is a prime number.*

**Proof.** The algorithm is as follows.

**Input:** Integer $n \geq 2$.
**Output:** *YES* if $n$ is a prime number, and *NO* otherwise.
$r := 2$; $s := 4$; $\quad (* \ s = r^2 \ *)$
**while** $s \leq n$
**do if** $n \bmod r = 0$
  **then** return *NO*
  **else** $r := r + 1$; $s := s + 2r - 1$ $\quad (* \ s = r^2 \ *)$
  **endif**
**endwhile**;
return *YES*

The correctness proof follows from the fact that $n$ is not a prime number if and only if there is an integer $r \leq \sqrt{n}$ that divides $n$. The while-loop makes at most $\sqrt{n}$ iterations, each one taking $O(\log^2 n)$ time. ∎

**Lemma 4.21** *Let $n \geq 2$ be an integer. There is an algorithm that computes the largest prime factor of $n$ in $O(\sqrt{n}\log^2 n)$ time.*

**Proof.** Consider the following algorithm.

**Input:** Integer $n \geq 2$.
**Output:** The largest prime factor of $n$.
$f := 1$; $r := 2$; $x := n$;
**while** $x \neq 1$
**do while** $x \bmod r = 0$
  **do** $x := x/r$; $f := r$
  **endwhile**;
  $r := r + 1$
**endwhile**;
return $f$

To see the correctness of this algorithm, consider the prime factorization $n = p_1^{k_1} p_2^{k_2} \ldots p_m^{k_m}$, where $p_1 < p_2 < \ldots < p_m$ are the prime factors of $n$. In

the outer while-loop, the value of $r$ is first increased from 2 to $p_1$. Then, in the inner while-loop, the $k_1$ factors of $p_1$ are "removed" from $n$, and the largest prime factor found so far is stored in the variable $f$. Then, $r$ is increased from $p_1$ to $p_2$, the $k_2$ factors of $p_2$ are removed, and the largest prime factor found so far is stored in the variable $f$. The algorithm continues until all prime factors have been removed.

During the algorithm, the value of $x \bmod r$ is computed

$$(p_m - 1) + k_1 + k_2 + \ldots + k_m \leq n + \log n$$

times. Similarly, the integer quotient $x/r$ is computed $k_1+k_2+\ldots+k_m \leq \log n$ times. Therefore, the algorithm takes $O(n \log^2 n)$ time.

We now change the algorithm as follows.

**Input:** Integer $n \geq 2$.
**Output:** The largest prime factor of $n$.
$f := 1$; $r := 2$; $x := n$;
**while** $x \neq 1$ and $r^2 \leq n$
**do while** $x \bmod r = 0$
    **do** $x := x/r$; $f := r$
    **endwhile**;
    $r := r + 1$
**endwhile**;
**if** $x = 1$ **then** return $f$ **else** return $x$ **endif**

The correctness proof of this modified algorithm is left to the reader. Since the outer while-loop makes at most $\sqrt{n}$ iterations, the running time is $O(\sqrt{n} \log^2 n)$. ∎

### 4.5.2 Testing if $n$ is a perfect power

Let $n \geq 2$ be an integer. In this section, we consider the problem of deciding if $n$ is a perfect power, i.e., if there exist integers $a \geq 2$ and $b \geq 2$ such that $n = a^b$. Since $b$ cannot be larger than $\log n$, this problem reduces to deciding, for every $2 \leq b \leq \log n$, if there exists an integer $a \geq 2$ with $n = a^b$.

For any fixed integer $b \geq 2$, we give an algorithm that computes the integer $q := \lfloor n^{1/b} \rfloor$. Then, $n$ is a perfect $b$-th power if and only if $q^b = n$. The

**Algorithm** *power*$(n, b)$
**Input:** Integers $n \geq 2$ and $b \geq 2$.
**Output:** $\lfloor n^{1/b} \rfloor$.
$x_0 :=$ any integer such that $x_0^b \geq n$;
$i := 0$;
**while** $x_i^b > n$
**do** $x_{i+1} := \lfloor ((b-1)x_i + n/x_i^{b-1})/b \rfloor$;
    $i := i + 1$
**endwhile**;
return $x_i$

Figure 2: *The algorithm that computes* $\lfloor n^{1/b} \rfloor$.

---

algorithm for computing $q$ is an integer-arithmetic version of the Newton-Raphson algorithm for computing the root of the function $x^b - n$. The algorithm, which is denoted by *power*$(n, b)$, is given in Figure 2.

**Lemma 4.22** *For all $i \geq 0$ for which $x_i$ exists, we have $x_i \geq q$.*

**Proof.** Since $x_0$ is an integer and $x_0^b \geq n$, it is clear that $x_0 \geq q$. Consider the function

$$f(x) = (b-1)x^b - bn^{1/b}x^{b-1} + n.$$

Observe that

$$f'(x) = b(b-1)x^{b-2}(x - n^{1/b}) > 0$$

for all $x > n^{1/b}$. Therefore, for all $x$ with $x^b > n$, we have $f(x) > f(n^{1/b}) = 0$. The latter inequality is equivalent to

$$((b-1)x + n/x^{b-1})/b > n^{1/b}.$$

Hence, if $x_i^b > n$, then $x_{i+1}$ exists and

$$x_{i+1} = \lfloor ((b-1)x_i + n/x_i^{b-1})/b \rfloor \geq \lfloor n^{1/b} \rfloor = q.$$

■

**Lemma 4.23** *The integers $x_i$, $i \geq 0$, are decreasing.*

**Proof.** If $x_i^b > n$, then $x_{i+1}$ exists and

$$x_{i+1} \leq ((b-1)x_i + n/x_i^{b-1})/b.$$

Since $x_i^b > n$, the right-hand side is less than $x_i$. ∎

**Lemma 4.24** *Algorithm power$(n, b)$ computes $\lfloor n^{1/b} \rfloor$.*

**Proof.** It follows from Lemma 4.23 that the while-loop terminates. The output is the first $x_i$ for which $x_i^b \leq n$. Since $x_i \leq n^{1/b}$ and $x_i$ is an integer, we have $x_i \leq q$. By Lemma 4.22, we have $x_i \geq q$. Therefore, the output of the algorithm is equal to $q$. ∎

Observe that algorithm *power*$(n, b)$ computes $\lfloor n^{1/b} \rfloor$ for any choice of the start value $x_0$, as long as $x_0^b \geq n$. To obtain a good upper bound on the number of iterations made by algorithm *power*$(n, b)$, we have to choose $x_0$ in a careful way.

**Lemma 4.25** *Let $m$ be the integer such that $2^{(m-1)b} < n \leq 2^{mb}$, and let $x_0 := 2^m$. Then*

1. *$x_0^b \geq n$ (and, hence, algorithm power$(n, b)$ computes $q = \lfloor n^{1/b} \rfloor$, and*
2. *$q \leq x_0 < 2n^{1/b}$.*

**Proof.** We have $x_0^b = 2^{mb} \geq n$, and

$$q = \lfloor n^{1/b} \rfloor \leq n^{1/b} \leq 2^m = x_0 = 2 \cdot 2^{m-1} < 2n^{1/b}.$$

∎

**Lemma 4.26** *Let $i \geq 0$ be such that $n^{1/b} < x_i < 2n^{1/b}$. Then $x_{i+1}$ exists, and*

$$x_{i+1} - n^{1/b} \leq \frac{b-1}{b+1} \left( x_i - n^{1/b} \right).$$

**Proof.** We have

$$x_{i+1} - n^{1/b} \leq ((b-1)x_i + n/x_i^{b-1})/b - n^{1/b}.$$

The right-hand side is less than or equal to $((b-1)/(b+1))(x_i - n^{1/b})$ if and only if

$$(b-1)x_i^b - 2bn^{1/b}x_i^{b-1} + (b+1)n \leq 0.$$

Consider the function

$$g(x) = (b-1)x^b - 2bn^{1/b}x^{b-1} + (b+1)n.$$

Observe that

$$g'(x) = b(b-1)x^{b-2}(x - 2n^{1/b}) < 0$$

for all $x$ with $n^{1/b} < x < 2n^{1/b}$. Therefore, for all $x$ with $n^{1/b} < x < 2n^{1/b}$, we have $g(x) < g(n^{1/b}) = 0$. ∎

**Lemma 4.27** *Let $b$ be an integer with $2 \leq b \leq \log n$, and let $x_0$ be as in Lemma 4.25. The while-loop of algorithm power$(n, b)$ makes $O(\log n)$ iterations, where the constant in the Big-Oh bound does not depend on $b$.*

**Proof.** Denote the number of iterations by $\ell$. We may assume that $\ell \geq 2$. It follows from algorithm *power*$(n, b)$ and Lemmas 4.23 and 4.25 that

$$x_\ell \leq n^{1/b} < x_{\ell-1} < x_{\ell-2} < \ldots < x_0 < 2n^{1/b}.$$

By applying Lemma 4.26 repeatedly, it follows that

$$x_{\ell-2} - n^{1/b} \leq \left(\frac{b-1}{b+1}\right)^{\ell-2} \left(x_0 - n^{1/b}\right) \leq \left(\frac{b-1}{b+1}\right)^{\ell-2} n^{1/b}.$$

Since $n^{1/b} < x_{\ell-1} < x_{\ell-2}$, and since $x_{\ell-1}$ and $x_{\ell-2}$ are integers, we have $x_{\ell-2} - n^{1/b} > 1$. It follows that

$$1 < \left(\frac{b-1}{b+1}\right)^{\ell-2} n^{1/b}.$$

Taking logarithms yields

$$0 < (\ell-2)\log\left(\frac{b-1}{b+1}\right) + \log n^{1/b}$$

and, hence,

$$\ell - 2 < \frac{\log n^{1/b}}{\log((b+1)/(b-1))}.$$

It follows that the number $\ell$ of iterations is less than

$$2 + \frac{\log n^{1/b}}{\log((b+1)/(b-1))} = O\left(\frac{\log n}{b\log((b+1)/(b-1))}\right).$$

If $b$ is bounded from above by a constant, then this is clearly $O(\log n)$. Observe that

$$\left(\frac{b+1}{b-1}\right)^b = \left(1+\frac{2}{b-1}\right)^{b-1}\left(1+\frac{2}{b-1}\right)$$

converges to $e^2$ if $b \to \infty$, where $e$ is Euler's number. Therefore, if $b$ is larger than some constant $b_0$, we have $((b+1)/(b-1))^b \geq e$, i.e., $b\log((b+1)/(b-1)) \geq \log e$. This proves that for all $b$ with $b_0 \leq b \leq \log n$, the number of iterations is $O(\log n)$. ∎

**Lemma 4.28** *Let $n$ and $b$ be integers with $2 \leq b \leq \log n$. Algorithm power$(n,b)$, with $x_0$ as in Lemma 4.25, computes $\lfloor n^{1/b} \rfloor$ in $O(\log^3 n \log\log n)$ time, where the constant in the Big-Oh bound does not depend on $b$.*

**Proof.** Observe that, by Lemmas 4.23 and 4.25,

$$x_i^b \leq x_0^b < 2^b n \leq n^2,$$

for all $i$ for which $x_i$ exists. Given $x_i$, the powers $x_i^{b-1}$ and $x_i^b$ can be computed in $O(\log^2 n \log\log n)$ time by Lemma 4.17. Given these powers, $x_{i+1}$ can be computed as

$$x_{i+1} = \left\lfloor \frac{(b-1)x_i^b + n}{bx_i^{b-1}} \right\rfloor,$$

which takes $O(\log^2 n)$ time by Lemma 4.16. This, together with Lemma 4.27, implies the lemma. ∎

**Lemma 4.29** *Let $n \geq 2$ be an integer. There is an algorithm that decides in $O(\log^4 n \log\log n)$ time whether there exist integers $a \geq 2$ and $b \geq 2$ such that $n = a^b$.*

**Proof.** If $n = a^b$ for integers $a \geq 2$ and $b \geq 2$, then $b$ must be less than or equal to $\log n$. By Lemma 4.28, we can compute $q_b := \lfloor n^{1/b} \rfloor$ for all $2 \leq b \leq \log n$, in $O(\log^4 n \log\log n)$ total time. If $q_b^b = n$ for at least one such $b$, then $n$ is of the form $a^b$. ∎

# 5 Further reading

The primality testing algorithm is due to Agrawal, Kayal, and Saxena [1]. My notes are based on the original paper and on notes by Radhakrishnan [10]. For an alternative description of the primality testing algorithm, see Bernstein [3].

For Section 4, I used the books by Cormen, Leiserson, Rivest, and Stein [5], and Lidl and Niederreiter [8]. For Section 4.5.2, I used the book by Cohen [4]. Faster algorithms that test if an integer $n$ is a perfect power are given by Bernstein [2]. For a recent and thorough overview of algorithms in number theory, see the book by Crandall and Pomerance [6].

Recent news about polynomial-time primality testing algorithms can be found at

`http://fatphil.org/maths/AKS/`


# Acknowledgements

The author thanks Rose de Guzman for helpful discussions, and Isabel Logie for giving comments on an earlier version of these notes and for providing the proof of Lemma 3.1.


# References


[1] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. http://www.cse.iitk.ac.in/users/manindra/index.html, 2002.

[2] D. J. Bernstein. Detecting perfect powers in essentially linear time. *Mathematics of Computation*, 67:1253–1283, 1998.

[3] D. J. Bernstein. An exposition of the Agrawal-Kayal-Saxena primality-proving theorem. http://cr.yp.to/papers.html, 2002.

[4] H. Cohen. *A Course in Computational Algebraic Number Theory.* Springer-Verlag, 1993.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, Cambridge, MA, 2nd edition, 2001.

[6] R. Crandall and C. Pomerance. *Prime Numbers, a Computational Perspective*. Springer-Verlag, 2001.

[7] E. Fouvry. Théorème de Brun-Titchmarsh; application au théorème de Fermat. *Invent. Math.*, 79:383–407, 1985.

[8] R. Lidl and H. Niederreiter. *Introduction to Finite Fields and Their Applications*. Cambridge University Press, 1986.

[9] M. Nair. On Chebyshev-type inequalities for primes. *American Mathematical Monthly*, 89:126–129, 1982.

[10] J. Radhakrishnan. News from India, Primes is in P. *Bulletin of the EATCS*, 78:61–65, 2002.