AKS PRIMALITY TESTING
THE ALGORITHM, THEN THE PROOF

Abstract. AKS decides whether an integer is prime using a deterministic number of bit operations polynomial in its input length. Its main test is a polynomial version of Fermat's little theorem, compressed so that the polynomials stay small. We present the algorithm first and then prove it from the top down. The central argument counts one finite group in two ways: passing the tests creates many distinct products, while a composite input forces too many of those products to satisfy one low-degree equation. Slightly rounded parameters let us express the lower bound using subsets. All the algebraic and counting lemmas needed for this argument are proved below, including an elementary bound on the search for the parameter r.

1. Introduction: what are we trying to accomplish?

Give an algorithm the integer $N = 31$. It should answer "prime." Give it $N = 35$, and it should answer "composite." Trial division already does this. The interesting question is how the running time grows when N has hundreds or thousands of bits.

Throughout, N is the integer being tested and $n = \lfloor \log_2 N \rfloor + 1$ is its bit length. After the preliminary checks handle powers of two, we have

$$2^{n-1} < N < 2^n.$$

All logarithms below are base 2. A bound polynomial in n is the right notion of efficiency. A bound such as $O(\sqrt{N})$ is exponential in the input length.

Agrawal, Kayal, and Saxena proved that primality has a deterministic polynomial-time algorithm [AKS04]. Here is the version we will analyze. Its parameters are rounded a little more generously than those in the published algorithm; this makes the proof easier to read.

Main result

Theorem 1 (AKS, with rounded parameters)**.** *Algorithm 1 returns PRIME exactly when its input $N \geq 2$ is prime. It is deterministic and uses $O(n^{17})$ bit operations with ordinary schoolbook arithmetic.*

The exponent here is a convenient, deliberately loose bound for this implementation. The mathematical conclusion is PRIMES $\in$ P.

The algorithm asks whether a few polynomial identities hold. The difficult direction is to show that a composite number cannot pass all of them. Here is the whole proof idea before we introduce its notation.

The counting argument we will prove

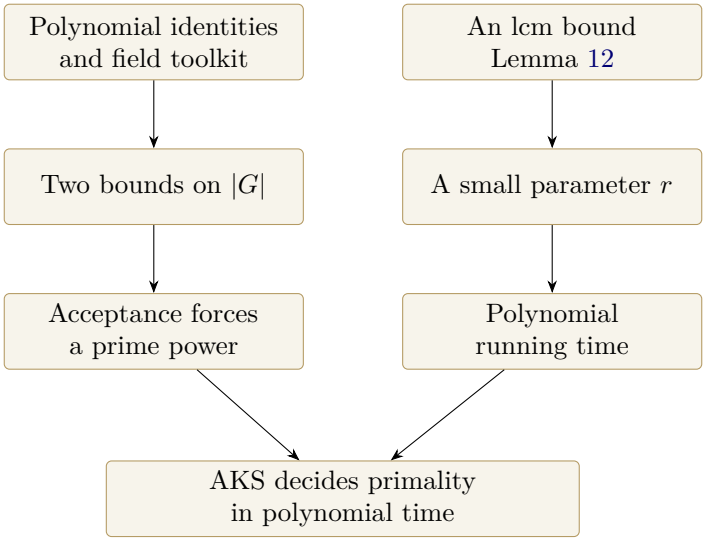
Suppose a composite input survives the preliminary checks and passes all the polynomial tests. From any suitable prime divisor of the input, we construct a finite multiplicative group G. For an integer $k \geq 1$, the two sides of the argument give

$$|G| \geq N^k \qquad \text{and} \qquad |G| < N^k.$$

The first inequality comes from distinct subsets of linear factors. The second comes from the pigeonhole principle and the fact that a nonzero polynomial cannot have more roots than its degree.

How to read the note. Section 2 gives the complete algorithm and its data representation. Section 3 proves the main theorem using two precise intermediate results. We then open those results one at a time: the counting argument, the polynomial identities behind it, the small parameter r, and the running time. The field used in the proof is built explicitly in Section 7. It is never computed by the algorithm.

We use basic modular arithmetic, the Euclidean algorithm, the binomial theorem, and prime factorization as mathematical facts. We do not assume that the algorithm can factor its input. No prior course in finite fields is needed: the field facts used in the proof are established here.



An arrow $A \longrightarrow B$ means that A is used to prove B.

2. The algorithm and the objects it computes

2.1. **A polynomial version of Fermat's theorem.** For a prime p,

$$(X+a)^p = X^p + a \pmod p \tag{2.1}$$

for every integer a. Equality here means equality of every coefficient modulo p. For example,

$$(X+1)^5 = X^5 + 5X^4 + 10X^3 + 10X^2 + 5X + 1 = X^5 + 1 \pmod 5.$$

For $N = 6$, the middle coefficients do not disappear:

$$(X+1)^6 = X^6 + 3X^4 + 2X^3 + 3X^2 + 1 \pmod 6.$$

In fact, $(X+1)^N = X^N + 1 \pmod N$ characterizes primes; we prove this in Lemma 7. But its degree is N, so explicitly storing it would require exponentially many coefficients. AKS keeps the identity and compresses its representation [AKS04, Lemma 2.1 and equation (2), p. 783].

2.2. **Compression means that exponents wrap around.** Fix a positive integer r. Working modulo $X^r - 1$ means using the rule $X^r = 1$. Every polynomial has a unique representative of degree less than r. For instance, if $r = 3$,

$$X^8 + 4X^4 + 2 = X^2 + 4X + 2 \pmod{X^3 - 1}.$$

We write

$$R_{N,r} = (\mathbb{Z}/N\mathbb{Z})[X]/(X^r - 1)$$

for this arithmetic. An element is just an array of r coefficients modulo N. To multiply two arrays, multiply polynomials and send the coefficient of X^j to position $j \bmod r$. Nothing needs to be divided by a coefficient. In particular, this arithmetic works whether or not N is prime.

The test we can afford is

$$(X+a)^N = X^N + a \quad \text{in } R_{N,r}. \tag{2.2}$$

Repeated squaring computes the left side using $O(n)$ multiplications of arrays of length r. The right side has a 1 in position $N \bmod r$ and an a in position 0, with addition if these positions coincide.

2.3. **The parameter that prevents too much compression.** If $\gcd(N,r)=1$, the *multiplicative order* $\mathrm{ord}_r(N)$ is the smallest positive j with $N^j = 1 \pmod r$. Such a j exists: multiplication by a unit permutes a finite set. AKS asks for

$$\mathrm{ord}_r(N) > n^2. \tag{2.3}$$

This says that $1, N, N^2, \ldots, N^{n^2}$ are distinct modulo r. Those distinct exponents will become distinct roots in the correctness proof.

To check (2.3), there is no need to factor r or compute its exact order. Compute $N, N^2, \ldots, N^{n^2} \pmod r$ and check that none is 1.

Algorithm 1: AKS (rounded parameters)

Input: an integer $N \geq 2$. Set $n = \lfloor \log N \rfloor + 1$.

1. If $N = 2$, return PRIME.
2. If $N = u^v$ for integers $u \geq 2$, $v \geq 2$, return COMPOSITE.
3. Find the least $r \geq 2$ such that $\gcd(N,r) = 1$ and $\mathrm{ord}_r(N) > n^2$.
4. For every integer a with $2 \leq a \leq r$, compute $d = \gcd(a, N)$. If $1 < d < N$, return COMPOSITE.
5. If $N \leq r$, return PRIME.
6. Set $\ell = \lceil n\sqrt{r} \rceil$. For every $1 \leq a \leq \ell$, check
$$(X+a)^N = X^N + a \quad \text{in } R_{N,r}.$$
If a check fails, return COMPOSITE.
7. Return PRIME.

The search in Step 3 terminates with $r \leq 2n^5$ for $N \geq 3$ (Theorem 3). The ceiling in Step 6 can be computed using integer arithmetic: ℓ is the least integer with $\ell^2 \geq n^2 r$.

For comparison, the published AKS algorithm uses $\log^2 N$ for the order threshold and $\lfloor \sqrt{\varphi(r)} \log N \rfloor$ tests [AKS04, p. 784]. Our larger bounds remove $\varphi(r)$ from the algorithm and make a subset-counting proof possible. This is the same AKS construction with different numerical parameters.

3. Prove the main theorem first

Here are the two intermediate results that drive the proof. Their proofs come later, so we can first see how the whole argument fits together.

The correctness criterion

Theorem 2. *Let $N \geq 3$, $n = \lfloor \log N \rfloor + 1$, and let $r \geq 2$ satisfy $\gcd(N,r) = 1$ and $\mathrm{ord}_r(N) > n^2$. Suppose every prime divisor of N is greater than r. If all the identities (2.2) hold for $1 \leq a \leq \ell = \lceil n\sqrt{r} \rceil$, then N is a power of a prime.*

The search for r stays small

Theorem 3. *For every integer $N \geq 3$, some $r \leq 2n^5$ satisfies $\gcd(N, r) = 1$ and $\mathrm{ord}_r(N) > n^2$.*

Proof of Theorem 1, using these two results. A prime is always accepted. It is not a proper perfect power. Every gcd in Step 4 is either 1 or N, and the polynomial identities hold by (2.1). Thus no step rejects a prime.

An accepted input is prime. The case $N = 2$ is settled at Step 1. Suppose first that Step 5 accepts. If N were composite, it would have a prime divisor $p < N \leq r$. Step 4 would have found $1 < \gcd(p, N) = p < N$, a contradiction.

Otherwise, the algorithm reaches Step 6 with $N > r$. Since Step 4 found no proper divisor, every prime divisor of N exceeds r. Passing all the polynomial tests therefore makes N a prime power by Theorem 2. Step 2 has already ruled out every proper perfect power. The remaining possibility is that N itself is prime.

Finally, Theorem 3 bounds the search parameter by $2n^5$. The implementation analysis in Section 9 then gives the claimed $O(n^{17})$ bit bound. Every step uses a specified finite search or exact arithmetic, so the algorithm is deterministic. □

What remains

The difficult part is Theorem 2. We will prove it by assuming that N has at least two distinct prime divisors and deriving the two incompatible bounds on $|G|$. After that, the search bound for r is an elementary argument about least common multiples.

4. The central argument: one group, two counts

This section proves Theorem 2 using three lemmas whose statements appear before they are used. The two counting lemmas are proved in Section 5; the algebraic setup is proved in Section 6.

4.1. **Choose a prime divisor for the proof.** Assume the hypotheses of Theorem 2, and suppose for a contradiction that N is not a prime power. We may choose a prime divisor $p \mid N$ such that

$$p \not\equiv 1 \pmod{r}. \tag{4.1}$$

Indeed, if every prime divisor were 1 (mod r), their product with multiplicity would give $N = 1$ (mod r). That would make $\mathrm{ord}_r(N) = 1$. Also, $p > r$ by hypothesis. Put $b = N/p$.

This is a choice inside the proof. The algorithm neither finds p nor uses it in a computation.

4.2. **Two finite groups.** Here a multiplicative group is a set containing 1, closed under multiplication and inverses, with the usual associative multiplication. The order of an element is the least positive exponent giving 1.

Write $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$, a field because every nonzero residue has an inverse. We will work in a finite field K containing $\mathbb{F}_p$ and an element ζ whose multiplicative order is exactly r. Thus

$$\zeta^r = 1, \qquad \zeta^i = \zeta^j \iff i = j \pmod{r}.$$

The element ζ is called a *primitive rth root of unity*. Its existence, and all the identities we need, are included in the next lemma.

Let $\mathcal{P}$ be the set of formal polynomials

$$f(X) = \prod_{a=1}^{\ell} (X + a)^{e_a}, \qquad e_a \geq 0,$$

with coefficients in $\mathbb{F}_p$; the empty product is 1. Set

$$\mathcal{I} = \{b^i p^j : i, j \geq 0\}, \qquad H = \{m \bmod r : m \in \mathcal{I}\}, \qquad G = \{f(\zeta) : f \in \mathcal{P}\}.$$

Because b, p divide N and $\gcd(N, r) = 1$, both are invertible modulo r. Thus H is a group of invertible residues modulo r, and G will be a multiplicative group in K. In a finite group, all inverses of a generator are already nonnegative powers of it. This is why the nonnegative exponents in these definitions give the whole generated groups.

Lemma 4 (Algebraic setup)**.** *There is a finite field K and a primitive rth root $\zeta \in K$ such that $\zeta \notin \mathbb{F}_p$. In particular, $\zeta + a \neq 0$ for every $a \in \mathbb{F}_p$, so G is a finite multiplicative group. For every $f \in \mathcal{P}$ and $m \in \mathcal{I}$,*

$$f(\zeta)^m = f(\zeta^m). \tag{4.2}$$

The identity says that two operations agree: first evaluate and then take the mth power, or first take the mth power of the input and then evaluate. This is how one equality will spread to many points.

Write $t = |H|$. Since $N = bp$, the group H contains all powers of N. Therefore

$$n^2 < \operatorname{ord}_r(N) \leq t \leq r - 1. \tag{4.3}$$

Set

$$k = \left\lfloor \sqrt{t} \right\rfloor, \qquad D = nk. \tag{4.4}$$

These parameters satisfy

$$D < t, \qquad D \leq \ell, \qquad D < p. \tag{4.5}$$

For the first inequality, $n < \sqrt{t}$ and $k \leq \sqrt{t}$, so $nk < t$. For the second, $nk \leq n\sqrt{r} \leq \ell$. The last follows from $D < t \leq r - 1 < p$.

Lemma 5 (Many products)**.** *In this setup, $|G| \geq 2^D \geq N^k$.*

Lemma 6 (A small equation for every product)**.** *If N is not a power of p, then $|G| < N^k$.*

Proof of Theorem 2. If N were not a prime power, choose p, b, K, ζ, H, G as above. In particular, N would not be a power of this chosen p. Lemma 5 gives $|G| \geq N^k$, while Lemma 6 gives $|G| < N^k$. The contradiction proves the theorem. □

This is the architecture of the AKS proof. The published version counts more general products and uses a binomial coefficient. Our rounded parameters let us count subsets instead; compare [AKS04, Lemmas 4.7–4.8, pp. 787–788] and [Sho08, Lemmas 21.9–21.10, pp. 556–558].

5. Open the two counting lemmas

We will use the following elementary fact several times: a nonzero polynomial of degree d over a field has at most d roots. A proof appears in Lemma 8. Notice that this statement applies over the field K; it need not hold over $\mathbb{Z}/N\mathbb{Z}$ for composite N.

5.1. The lower bound: every subset gives something new.

Proof roadmap

Use the first D linear factors $X + 1, \ldots, X + D$. Each subset gives one product of degree at most $D < t$. If two products have the same value at ζ, identity (4.2) makes them agree at t distinct points. Their difference would have more roots than its degree.

Proof of Lemma 5. For each subset $S \subseteq \{1, \ldots, D\}$, define

$$f_S(X) = \prod_{a \in S} (X + a).$$

There are 2^D such products, and their degrees are at most $D < t$. They are distinct as formal polynomials over $\mathbb{F}_p$: if $S \neq T$, choose (after interchanging S, T if necessary) $a \in S \setminus T$. Then $f_S(-a) = 0$, whereas

$$f_T(-a) = \prod_{c \in T} (c - a) \neq 0.$$

Every factor is nonzero modulo p because $1 \leq a, c \leq D < p$ and $c \neq a$.

We claim that these products also have distinct values at ζ. Suppose $f_S(\zeta) = f_T(\zeta)$. For any $m \in \mathcal{I}$, raise this equality to the mth power and use (4.2) to obtain

$$f_S(\zeta^m) = f_T(\zeta^m).$$

As m ranges over $\mathcal{I}$, its residues range over the t elements of H. Since ζ has order r, these give t distinct points ζ^m. Thus $f_S - f_T$ is a nonzero polynomial of degree at most $D < t$ with at least t roots. This is impossible.

Every $f_S(\zeta)$ belongs to G, so

$$|G| \geq 2^D = 2^{nk} \geq N^k,$$

where the last inequality uses $2^n \geq N$. □

5.2. **The upper bound: a collision of exponents.**

Proof roadmap

Take a square grid of exponents $b^i p^j$, with $0 \leq i, j \leq k$. The grid contains more than t distinct integers, so two have the same residue in H. Their difference becomes an exponent that sends every element of G to 1. The root bound then makes G small.

Proof of Lemma 6. Because N is not a power of p, another prime $q \neq p$ divides $b = N/p$. The integers

$$b^i p^j, \qquad 0 \leq i, j \leq k,$$

are all distinct. To check this, suppose $b^i p^j = b^{i'} p^{j'}$. Comparing the exponents of q gives $i = i'$. Cancelling b^i then gives $j = j'$.

There are $(k+1)^2 > t$ integers in this grid and only t possible residues in H. Hence there are $m_1 > m_2$ in the grid with $m_1 = m_2 \pmod{r}$. Put $\delta = m_1 - m_2 > 0$.

Take any $z \in G$ and write $z = f(\zeta)$ for some $f \in \mathcal{P}$. Using (4.2) twice gives

$$z^{m_1} = f(\zeta^{m_1}) = f(\zeta^{m_2}) = z^{m_2}.$$

Since $z \neq 0$, division gives $z^\delta = 1$. Thus every element of G is a root of the nonzero polynomial $Y^\delta - 1$ over K, so $|G| \leq \delta$. Finally, every grid entry is at most

$$b^k p^k = (bp)^k = N^k.$$

Consequently

$$|G| \leq m_1 - m_2 < m_1 \leq N^k.$$

This is the required strict upper bound. □

Pause and check the mechanism. There are two different uses of the root bound. For the lower bound, it prevents two low-degree products from colliding at ζ. For the upper bound, it limits how many nonzero elements can satisfy $z^\delta = 1$. The same evaluation identity connects both uses.

The choice of $b = N/p$ is also doing real work. The largest grid entry is exactly $b^k p^k = N^k$, which matches the lower bound. Replacing this grid by $N^i p^j$ would give a weaker upper bound.

6. Where the evaluation identity comes from

We now prove Lemma 4. The field element ζ is constructed in Section 7; first we explain the identities that make it useful. This is the "introspective" part of the original proof [AKS04, Definition 4.4 and Lemmas 4.5–4.6, pp. 786–787].

Proof roadmap

Reduce the successful tests modulo p. Then cancel a pth power to get the identity for $b = N/p$. Close under products of factors and exponents, and finally evaluate at ζ. Each step preserves exactly the identity needed by the counting argument.

6.1. **Frobenius is easy in characteristic p.** For $0 < j < p$, the binomial coefficient $\binom{p}{j}$ is divisible by p. Hence, in any commutative ring of characteristic p,

$$(u+v)^p = u^p + v^p.$$

Here characteristic p means that adding 1 to itself p times gives 0. Also $a^p = a$ for $a \in \mathbb{F}_p$: apply this identity repeatedly to a sum of copies of 1. For a polynomial f with coefficients in $\mathbb{F}_p$, it follows that

$$f(X)^p = f(X^p). \tag{6.1}$$

Work now in

$$R = \mathbb{F}_p[X]/(X^r - 1).$$

Because $p > r$, we have $\gcd(p, r) = 1$. If $u = \sum_{j=0}^{r-1} c_j X^j$, then

$$u^p = \sum_{j=0}^{r-1} c_j X^{pj} \quad \text{in } R.$$

The map $j \mapsto pj \bmod r$ permutes the r coefficient positions. Thus $u \mapsto u^p$ is injective on R: if $u^p = v^p$, then $u = v$. This is why we are allowed to cancel a pth power below. We are not assuming that R is a field.

6.2. **First obtain the identity for p and $b = N/p$.** Reduce the algorithm's successful tests modulo the prime divisor p. For each tested factor $f(X) = X + a$, they give

$$f(X)^N = X^N + a \quad \text{in } R.$$

Since $N = pb$, we have

$$(f(X)^b)^p = X^{pb} + a = (X^b + a)^p.$$

Injectivity of the pth-power map gives

$$f(X)^b = f(X^b) \quad \text{in } R. \tag{6.2}$$

Equation (6.1) gives the same statement for the exponent p.

6.3. **Then close under multiplication.** Say that an exponent m *respects* a polynomial f when

$$f(X)^m = f(X^m) \quad \text{in } R.$$

Two short closure rules are enough.

(1) If m respects both f and g, it respects fg, by multiplying their two identities.
(2) If m and m' both respect f, then mm' respects f. Indeed,

$$f(X)^{mm'} = f(X^m)^{m'} = f(X^{mm'}) \quad \text{in } R.$$

The last step substitutes X^m for X in the identity for m'.

This substitution is legitimate: a multiple of $X^r - 1$ becomes a multiple of $X^{mr} - 1$, which is still divisible by $X^r - 1$.

Starting with the tested factors and the exponents b, p, the closure rules show that every $m \in \mathcal{I}$ respects every $f \in \mathcal{P}$. Evaluating at any ζ with $\zeta^r = 1$ gives (4.2).

6.4. **Finish the setup lemma.** Section 7 constructs a finite extension K of $\mathbb{F}_p$ with a primitive rth root ζ. It cannot lie in $\mathbb{F}_p$. Otherwise $\zeta^p = \zeta$, so $\zeta^{p-1} = 1$, which would imply $r \mid p-1$, contrary to (4.1). Therefore $\zeta + a \neq 0$ for every $a \in \mathbb{F}_p$.

The set G contains 1, is closed under multiplication, and lies in the finite group $K^\times$. It is itself a group: for $z \in G$, some positive power z^u equals 1, so z^{u-1} is an inverse in G. Together with the evaluation identity just proved, this establishes Lemma 4. □

6.5. **Why the uncompressed identity characterizes primes.** The next lemma completes the motivation from Section 2. Its converse is not needed elsewhere in the correctness proof.

Lemma 7. *For an integer $N \geq 2$, the coefficient identity $(X+1)^N = X^N + 1 \pmod N$ holds if and only if N is prime.*

Proof. The prime direction follows from the binomial calculation above. If N is composite, choose a prime $p \mid N$, and let p^e be the largest power of p dividing N. Then $p < N$, and

$$\binom{N}{p} = \frac{N}{p}\binom{N-1}{p-1}.$$

The second factor is $1 \pmod p$: its numerator is $(N-1)\cdots(N-p+1)$, whose factors reduce to $-1, \ldots, -(p-1)$; division by $(p-1)!$ is valid modulo p. Thus exactly p^{e-1} divides $\binom{N}{p}$, and p^e does not. This middle coefficient is not divisible by N, so the identity fails. □

7. Build the field used in the proof

There is only one purpose for the field K: it provides r distinct powers of ζ and lets us use the polynomial root bound. Here are complete proofs of the facts we need. These correspond to the standard results in [Sho08, Theorems 7.14, 7.29, and 16.25].

A *field extension* K of F is a field containing F with the same arithmetic on its elements. A nonconstant polynomial is *irreducible* if it cannot be written as a product of two polynomials of positive degree over the field in question.

Lemma 8 (Root bound)**.** *A nonzero polynomial of degree d over a field has at most d roots.*

Proof. If a is a root, polynomial division gives $f(X) = (X-a)g(X)$ with $\deg g = d-1$. Every other root is a root of g, since nonzero field elements can be cancelled. Induction on d proves the claim. □

Lemma 9 (A finite field large enough to contain the roots)**.** *Every nonzero polynomial over a finite field splits into linear factors over some finite extension field.*

Proof. Every nonconstant polynomial has an irreducible factor: repeatedly factor a reducible factor, decreasing its degree. If an irreducible factor $h(T)$ has degree $d > 1$, form $F[T]/(h)$. This is a field: for any nonzero representative g of degree less than d, irreducibility gives $\gcd(g, h) = 1$. Long division over a field gives the same Euclidean algorithm and Bézout identity as for integers. Thus $ug + vh = 1$ for some polynomials u, v, so u is an inverse of g modulo h. The quotient has $|F|^d$ elements, represented by coefficient arrays of length d, and the element represented by T is a root of h.

Thus we can adjoin a root of any irreducible factor that has not yet split. Divide out the resulting linear factor and repeat over the enlarged field. The remaining degree drops at each step, so after finitely many steps the whole polynomial splits. Each step preserves finiteness. □

Lemma 10 (Finite multiplicative subgroups are cyclic)**.** *If U is a finite subgroup of the nonzero elements of a field, some element of U has order $|U|$.*

Proof. Let e be the least common multiple of the orders of all elements of U. For each prime power q^a dividing e exactly, choose an element whose order has q-part q^a. If that order is $q^a c$ with $\gcd(q, c) = 1$, raising the element to the cth power produces an element of order q^a.

Multiply these chosen elements, one for each prime divisor of e. The product has order e. To justify this familiar fact, commuting elements of coprime orders u, v have product of order uv: if $(xy)^s = 1$, then $x^s = y^{-s}$ lies in both cyclic subgroups. An element in their intersection has order dividing both u and v, so is 1; hence $u \mid s$ and $v \mid s$. Repeating proves the assertion for all the chosen factors.

Every element of U satisfies $z^e = 1$, so the root bound gives $|U| \le e$. The element of order e has e distinct powers in U, giving $e \le |U|$. Thus $e = |U|$, as required. □

Lemma 11 (Primitive roots in a finite extension)**.** *If p is prime and $\gcd(p, r) = 1$, some finite extension of $\mathbb{F}_p$ contains an element of order exactly r.*

Proof. By Lemma 9, choose a finite field in which $X^r - 1$ splits. Its roots are distinct. Indeed, a repeated root α would also be a root of its formal derivative rX^{r-1}: this follows by differentiating a factorization containing $(X - \alpha)^2$. But $\alpha \neq 0$ and $r \neq 0$ in characteristic p.

There are therefore exactly r roots. They form a multiplicative subgroup: products and inverses of rth roots are again rth roots. Lemma 10 supplies an element of order r. □

This finishes the algebra used in the AKS argument. The field construction may be large, but it is an existence argument inside the proof. It contributes no operations to the algorithm.

8. Why a small parameter r exists

We turn to Theorem 3. The proof follows the corrected least-common-multiple approach to AKS: the original proof of its search lemma was corrected in 2019 [AKS19, pp. 317–318]. We include the whole argument, with constants adapted to our rounded n.

Proof roadmap

If every small modulus coprime to N had small order, then every small prime power would divide a single explicit integer Q. Consequently the least common multiple of $1, \dots, B$ would divide Q. But that least common multiple grows at least exponentially in B, while our formula for Q makes it too small. Set $B = 2n^5$ and compare the two bounds.

8.1. The elementary growth estimate.

Lemma 12. *For every integer $B \ge 1$,*

$$\operatorname{lcm}(1, \dots, B) \ge 2^{B-1}.$$

Proof. Put $M = \operatorname{lcm}(1, \dots, B)$. We first show that each of the integers

$$B\binom{B-1}{j} = \frac{B!}{j!(B-1-j)!}, \qquad 0 \le j \le B-1,$$

divides M. For a prime q, the exponent of q in $s!$ is $\sum_{a \ge 1} \lfloor s/q^a \rfloor$: count multiples of q, then the extra factors from multiples of q^2, and so on. Hence the exponent of q in the displayed integer is

$$\sum_{a \ge 1} \left(\left\lfloor \frac{B}{q^a} \right\rfloor - \left\lfloor \frac{j}{q^a} \right\rfloor - \left\lfloor \frac{B-1-j}{q^a} \right\rfloor \right).$$

Each summand is 0 or 1. To see this, write the two remainders of j and $B-1-j$ modulo q^a as u, v. The summand is $\lfloor (u+v+1)/q^a \rfloor$, and $0 \le u+v+1 < 2q^a$. Also the summand is 0 when $q^a > B$. Thus the exponent is at most $\lfloor \log_q B \rfloor$, exactly the exponent of q in M. This proves the divisibility claim.

The B binomial coefficients in row $B-1$ sum to 2^{B-1}, so one is at least $2^{B-1}/B$. Since M is at least B times that coefficient, $M \ge 2^{B-1}$. □

For another reference to this exact bound, see [Far09, Corollary 3, arXiv version 2, p. 4].

8.2. **Apply the growth estimate.**

Proof of Theorem 3. For $N \ge 3$ we have $n \ge 2$. Set

$$B = 2n^5, \qquad e = \lfloor \log B \rfloor, \qquad Q = N^e \prod_{i=1}^{n^2} (N^i - 1).$$

Suppose every $2 \le r \le B$ coprime to N had order at most n^2. We claim that every prime power $q^a \le B$ divides Q.

- If $q \mid N$, then $a \le \lfloor \log B \rfloor = e$, so $q^a \mid N^e$.
- If $q \nmid N$, the assumed order bound for the modulus q^a gives some $1 \le i \le n^2$ with $q^a \mid N^i - 1$.

It follows that $\operatorname{lcm}(1, \ldots, B) \mid Q$.

On the other hand, since $\log N \le n$ and $N^i - 1 < N^i$,

$$\begin{aligned} \log Q &< n \log B + n \sum_{i=1}^{n^2} i \\ &= n \log B + \frac{n^5 + n^3}{2} \\ &\le n^5 + \frac{5}{8} n^5 = \frac{13}{8} n^5 < 2n^5 - 1 = B - 1. \end{aligned}$$

Here are the numerical checks. For integer $n \ge 2$, $\log n \le n-1$, so $\log B = 1 + 5 \log n \le 5n - 4 \le 5n$. Therefore $n \log B \le 5n^2 \le n^5$. Also $n^3 \le n^5/4$, and $(3/8)n^5 > 1$.

We have proved $Q < 2^{B-1}$, contradicting

$$2^{B-1} \le \operatorname{lcm}(1, \ldots, B) \le Q$$

from Lemma 12. Thus some required $r \le 2n^5$ exists. □

Notice why N^e was included in Q. It absorbs *all* prime powers up to B whose primes divide N. A single factor of N would not be enough. This is the issue addressed by the AKS erratum [AKS19].

9. Running time and an implementation view

Proof roadmap

Count bits, not the numerical value of N. Every polynomial is stored as r residues with $O(n)$ bits each. The search bound gives $r = O(n^5)$, the number of tests is $O(n\sqrt{r})$, and repeated squaring uses $O(n)$ polynomial multiplications per test.

9.1. **The preliminary work is polynomial.** To test whether N is a perfect power, try each $2 \le v \le n$ and use binary search for an integer u with $u^v = N$. There is no need to try larger v, since $u \ge 2$ implies $2^v \le N$. In a comparison, multiply with an early stop as soon as the value exceeds N. The integers kept have $O(n)$ bits. Even using $O(n)$ multiplications per comparison, $O(n)$ comparisons per search, and $O(n)$ exponents gives $O(n^5)$ bit operations with schoolbook multiplication.

For each $r \le 2n^5$, compute $\gcd(N, r)$ and, if it is 1, check the first n^2 powers of $N \pmod r$. Each modulus has $O(\log n)$ bits. The loose bound $O(n^9)$ covers the whole search using ordinary integer arithmetic. The gcd checks through r cost at most $O(rn^3)$: Euclid's algorithm uses $O(n)$ divisions, each costing $O(n^2)$ by long division. For the division count, observe that every two remainders decrease by at least a factor of two. None of these steps requires factoring N or r.

9.2. **Polynomial multiplication is cyclic convolution.** For arrays $u = (u_0, \dots, u_{r-1})$ and $v = (v_0, \dots, v_{r-1})$, their product in $R_{N,r}$ has entries

$$w_k = \sum_{i+j \equiv k \pmod r} u_i v_j \pmod N.$$

The direct double loop uses r^2 coefficient multiplications. Reduce modulo N after each update, so all stored coefficients have $O(n)$ bits. One polynomial multiplication costs $O(r^2 n^2)$ bit operations. Binary exponentiation therefore checks one identity in $O(r^2 n^3)$ bit operations.

There are

$$\ell = \lceil n\sqrt{r} \rceil = O(n^{7/2})$$

tests. Their total cost is

$$O(\ell r^2 n^3) = O(n^{33/2}) \subseteq O(n^{17}).$$

The preliminary work is smaller. This proves the running-time statement used in Theorem 1. Compare the schoolbook analysis in [Sho08, Theorem 21.3, p. 551]; we make no claim that our loose exponent is an optimized AKS bound.

What is computed, and what appears only in the proof

Algorithm	Correctness proof
n, r, ℓ; gcds; modular powers; arrays of r coefficients	A prime divisor p; $b = N/p$; a finite field K; ζ; the groups H, G

10. Examples and a few useful checks

10.1. **A full small trace: $N = 31$.** Here $n = 5$, so the order threshold is 25. No $r \le 26$ can work, because a unit modulo r has order at most $r - 1 \le 25$. For the next candidates,

$$\operatorname{ord}_{27}(31) = 9, \qquad \operatorname{ord}_{28}(31) = 6, \qquad \operatorname{ord}_{29}(31) = 28.$$

Thus the search selects $r = 29$. For the last equality, $31 = 2 \pmod{29}$, $2^{14} = -1 \pmod{29}$, and $2^4 = 16 \not\equiv 1 \pmod{29}$; these rule out the proper divisors of 28 as the order.

The input is not a perfect power, and all gcds with $2, \dots, 29$ are 1. Since $31 > 29$, the algorithm reaches the polynomial tests, with $\ell = \lceil 5\sqrt{29} \rceil = 27$. For every tested a,

$$(X + a)^{31} = X^{31} + a = X^2 + a \quad \text{in } R_{31,29}.$$

All tests pass, and the output is PRIME.

For contrast, $N = 15$ has $n = 4$ and the search finds $r = 19$. Step 4 detects $\gcd(3, 15) = 3$ and returns COMPOSITE before any polynomial test. The input $N = 49$ is rejected even earlier, as 7^2.

10.2. **A small polynomial calculation.** For a concrete array example, work modulo 5 and $X^3 - 1$. Then

$$(X+1)^2 = X^2 + 2X + 1, \qquad (X+1)^4 = X^2 \quad \text{in } R_{5,3}.$$

Multiplying once more by $X + 1$ gives

$$(X+1)^5 = X^2 + 1 \quad \text{in } R_{5,3}.$$

For instance, the middle calculation starts from $X^4 + 4X^3 + 6X^2 + 4X + 1$ and wraps X^4 to X and X^3 to 1. This is an illustration of the arithmetic, not the parameter choice that AKS would use on input 5.

10.3. **Why the preliminary checks matter.**

- **Perfect powers.** The counting criterion concludes that N is a prime power. Step 2 converts that conclusion into primality. It also guarantees a second prime divisor when we need distinct integers in the exponent grid.
- **Small factors.** The gcd checks make $p > r$, so the short list $1, \ldots, D$ stays distinct modulo p.
- **Large order.** It gives $t > n^2$, hence $D = n\lfloor\sqrt{t}\rfloor < t$. That strict degree inequality is what makes the lower bound work.
- **A suitable prime divisor.** Choosing $p \not\equiv 1 \pmod{r}$ ensures $\zeta \notin \mathbb{F}_p$, so none of the factors $\zeta + a$ is zero.

10.4. **Exercises with short answers.**

(1) Why is testing all integers up to $\sqrt{N}$ not polynomial time in the input length? *Answer:* $\sqrt{N}$ is of size $2^{\Theta(n)}$.

(2) Show that $\ell \le r$ for our selected r. *Answer:* $r - 1 \ge \mathrm{ord}_r(N) > n^2$, so $n\sqrt{r} < r$; taking a ceiling gives $\ell \le r$.

(3) Where is the fact that K is a field used in the upper bound? *Answer:* to cancel $z^{m_2} \neq 0$ and to apply the root bound.

(4) Why is $(k+1)^2 > t$ even when t is a square? *Answer:* $k = \lfloor\sqrt{t}\rfloor$, so $k + 1 > \sqrt{t}$.

(5) Does this proof give a polynomial-time factoring algorithm? *Answer:* no. A failed polynomial identity certifies compositeness without necessarily revealing a factor.

11. Technical inputs and precise source locations

All the AKS-specific inputs below have been proved in this note. These boxes record the exact statements used and their source locations, so the dependency chain can be checked without guessing which version of a result is intended. Page numbers for Shoup refer to the printed numbering in the author's Version 2 PDF linked in the bibliography.

Input 1: elementary arithmetic

Every integer greater than 1 has a unique factorization into primes. For integers u, v, the extended Euclidean algorithm produces coefficients expressing $\gcd(u, v)$ as an integer linear combination of u, v. In particular, $\mathbb{Z}/p\mathbb{Z}$ is a field when p is prime. These are the elementary arithmetic prerequisites used throughout the note.

Precise references. Shoup, Theorem 1.3, p. 2 (proof completed on p. 8); Theorems 1.7–1.8, pp. 6–7; the extended Euclidean algorithm, Theorem 4.3, p. 78; and Example 7.11, p. 170 [Sho08].

Input 2: root counting and a field with primitive roots

A nonzero degree-d polynomial over a field has at most d roots. Every nonzero polynomial over a finite field splits in a finite extension. Every finite multiplicative subgroup of a field is cyclic. Consequently, if $p \nmid r$, a finite extension of $\mathbb{F}_p$ contains an element of order r.
Proofs here. Lemmas 8–11.
Precise references. Shoup: root bound, Theorem 7.14, p. 181; cyclicity, Theorem 7.29, p. 203; the irreducible quotient construction, §16.4, p. 437; splitting fields, Theorem 16.25, p. 443; the derivative criterion for repeated factors, Theorem 19.1, p. 509 [Sho08].

Input 3: the polynomial identities propagate

In $R = \mathbb{F}_p[X]/(X^r - 1)$ with $p \nmid r$, the map $f \mapsto f^p$ is injective. If a tested factor satisfies $f^N = f(X^N)$ and $p \mid N$, it also satisfies $f^{N/p} = f(X^{N/p})$. The property $f^m = f(X^m)$ is closed under products of polynomials and products of exponents.
Proof here. Section 6, using the explicit permutation $j \mapsto pj \bmod r$ of coefficient positions.
Precise references. AKS, Definition 4.4 and Lemmas 4.5–4.6, pp. 786–787 [AKS04]; Shoup, Lemmas 21.7–21.8, p. 555, and Frobenius in Example 7.48, p. 194, and Theorem 19.7, p. 512 [Sho08]. Injectivity in the quotient ring used here follows from our coefficient-permutation proof.

Input 4: the two counting estimates

With the parameters of Section 4, $k = \lfloor \sqrt{t} \rfloor$ and $D = nk$ give $|G| \geq 2^D \geq N^k$. If N is not a power of p, two distinct grid entries $b^i p^j$, $0 \leq i, j \leq k$, collide modulo r, and yield $|G| < N^k$.
Proofs here. Lemmas 5–6.
Precise references. AKS, Lemmas 4.7–4.8, pp. 787–788, and the concluding Lemma 4.9, pp. 788–789 [AKS04]. Shoup gives a related subset argument in Lemmas 21.9–21.11, pp. 556–558 [Sho08]. The numerical inequalities stated here use our rounded parameters; they are proved directly rather than quoted from either source.

Input 5: a small search parameter

For $B \geq 1$, $\operatorname{lcm}(1, \ldots, B) \geq 2^{B-1}$. For $N \geq 3$ and $n = \lfloor \log_2 N \rfloor + 1$, there is an $r \leq 2n^5$ coprime to N with $\operatorname{ord}_r(N) > n^2$.
Proofs here. Lemma 12 and Theorem 3.
Precise references. The lcm bound is Farhi, Corollary 3, arXiv:0906.2295v2, p. 4 [Far09]. The original AKS search lemma is Lemma 4.3, p. 785 [AKS04]; use its corrected proof in the 2019 erratum, pp. 317–318 [AKS19]. Our constant 2 and integer n are covered by the explicit calculation in Section 8.

The takeaway is short: compress a polynomial identity, keep enough distinct exponents, and count the same set in two incompatible ways. That is how the algorithm turns a small list of deterministic checks into a proof of primality.

References

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, September 2004.

[AKS19] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Errata: PRIMES is in P. *Annals of Mathematics*, 189(1):317–318, 2019. URL: https://annals.math.princeton.edu/wp-content/uploads/annals-v189-n1-p06-s.pdf, doi:10.4007/annals.2019.189.1.6.

[Far09] Bakir Farhi. An identity involving the least common multiple of binomial coefficients and its application. *The American Mathematical Monthly*, 116(9):836–839, 2009. See also arXiv:0906.2295v2, Corollary 3, p. 4. URL: https://arxiv.org/abs/0906.2295, doi:10.4169/000298909X474909.

[Sho08] Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, second edition, 2008. Author's Version 2, June 16, 2008. URL: https://www.shoup.net/ntb/ntb-v2.pdf.