Primality Testing and the AKS Algorithm

Nicholas Mosier

December 10, 2019

Abstract

Integer factorization is an inherently difficult problem for which there is no known algorithm that produces an answer in polynomial time. Until recently, it was unknown whether the very similar but weaker problem of *primality testing* is equally difficult. Primality testing is the problem of determining whether an arbitrary positive integer $n \in \mathbb{N}$ is prime or composite. In 2002, Agrawal, Kayal, and Saxena presented a provably correct deterministic primality test that terminates in polynomial time over all $\mathbb{N}$. Although previous polynomial-time algorithms had been presented, they rely upon unproven claims such as the Riemann Hypothesis. We restrict our focus to algorithms that are unconditionally correct.

In this paper, we review principles common to existing deterministic, polynomial-time primality tests, specifically Fermat's Little Theorem (FlT) and the related topic of cyclotomic fields. We introduce a simple primality test based on FlT, and then discuss increasingly advanced algorithms (Pocklington primality test, APR primality test). In the second part of the paper, we prove the correctness of the AKS Primality Test, concluding that primality testing is a problem solvable in polynomial time.

Contents

1 Notation and Definitions

1.1 Divisibility

Definition 1. An integer a divides another integer b if $b = ka$ for some integer k. Divisibility is denoted by $a \mid b$.

Definition 2. For all $k, n \in \mathbb{N}$, let $\alpha_k(n)$ denote the exponent on the largest power of k that divides n. More precisely,

$$\alpha_k(n) = \max\left\{i \in \mathbb{N} \cup \{0\} : k^i \mid n\right\}.$$

If k is prime, then $\alpha_k(n)$ denotes the exponent on k in the prime factorization of n.

1.2 Greatest Common Divisor

Definition 3. The greatest common divisor of two positive integers a and b, denoted by $\gcd(a, b)$, is defined as the greatest positive integer n that divides both a and b, i.e.

$$\gcd(a, b) = \max\left\{n \in \mathbb{N} : n \mid a \text{ and } n \mid b\right\}.$$

One useful property of the greatest common divisor is that it divides both of its arguments, i.e. given integers a and b,

$$\gcd(a, b) \mid a$$
$$\gcd(a, b) \mid b.$$

Property 1 (Generalization of Euclid's Lemma)**.** Let $n, a, b \in \mathbb{N}$. If $n \mid ab$ and $\gcd(n, a) = 1$, then $n \mid b$.

1.3 Least Common Multiple

Definition 4. The least common multiple of two positive integers a and b, denoted by $\text{lcm}(a, b)$, is the least positive integer n such that $a \mid n$ and $b \mid n$.

Definition 5. The least common multiple of the first n positive integers, denoted by $\text{LCM}(n)$, is the least positive integer N such that for all $m \leq n$, $m \mid N$.

Example 1. $\text{LCM}(5) = 2^2 \cdot 3 \cdot 5 = 60$, since $2 \mid 60$, $3 \mid 60$, $4 \mid 60$, and $5 \mid 60$, but for any proper divisor $d \mid 60$, some positive integer less than or equal ot 5 does not divide d.

Property 2. For all $n \in \mathbb{N}$, $\text{LCM}(n) \mid n$.

1.4 Modular Congruence

Definition 6. An integer a is congruent to another integer b modulo some positive integer n, denoted with $a \equiv b \pmod{n}$, if $n \mid (a - b)$.

Property 3. If $a \equiv b \pmod{mn}$, then

$$\begin{aligned} a &\equiv b \pmod{m} \quad \text{and} \\ a &\equiv b \pmod{n}. \end{aligned}$$

Definition 7. The order of an integer $a \in \mathbb{Z}$ modulo n, denoted by $o_n(a)$, is the smallest positive integer i such that $a^i \equiv 1 \pmod{n}$. (If no such i exists, a is said to have infinite order.)

Example 2. The order of $a = 3$ modulo $n = 4$ is $o_a(n) = 2$, since

$$\begin{aligned} 3^1 &\not\equiv 1 \pmod{4} \quad \text{but} \\ 3^2 = 9 &\equiv 1 \pmod{4}. \end{aligned}$$

1.5 Group Theory

Definition 8. A group is a set G under a closed binary operation $\cdot : G \times G \to G$ that satisfies the following properties:

- **Associativity**. $(ab)c = a(bc)$ whenever $a, b, c \in G$.
- **Identity**. There exists an *identity element* $1 \in G$ such that for all $a \in G$, $a \cdot 1 = 1 \cdot a = a$.
- **Inverse**. For each $a \in G$, there exists an inverse $a^{-1} \in G$ such that $aa^{-1} = a^{-1}a = e$.

Example 3. Let $F_p = \{0, 1, 2, \ldots, p-1\}$ denote the finite field of the first p nonnegative integers, where p is prime. F_p is a group under integer multiplication modulo p.

Definition 9. Let a be an element of a group G. The set generated by a, denoted by $\langle a \rangle$, is equal to

$$\langle a \rangle = \left\{ a^i : i \geq 0 \right\}.$$

$\langle a \rangle$ is a subgroup of G.

Definition 10 (Roots)**.** Let a, b be elements of a group G. a is a nth root of b if $a^n = b$. If $b = 1$, then a is called a nth *root of unity*. If for all $0 \leq i < n$, $a^i \neq b$, then a is called a nth *primitive root* of b. Note that nth roots are not necessarily unique.

1.6 Polynomials

The following definitions are based on [1].

Definition 11 (Polynomial Rings)**.** Let R be a ring and let X be a symbolic variable. The *polynomial ring* of X over R, denoted by $R[X]$, is defined as

$$R[X] = \left\{ \sum_{i=0}^{n} r_i X^i \mid n \geq 0 \text{ and each } r_i \in R \right\}.$$

Definition 12 (Polynomial Divisibility)**.** Let polynomials $a(X), b(X) \in R[X]$. $a(X)$ divides $b(X)$, written as $a(X) \mid b(X)$, if $a(X) = b(X)c(X)$ for some polynomial $c(X) \in R[X]$.

Definition 13 (Polynomial Irreducibility)**.** A polynomial $a(X) \in R[X]$ is irreducible over R if no polynomial $b(X) \in R[X]$ of degree one or more (with $A(X) \neq B(X)$) divides $A(X)$. That is, if $b(X) \mid a(X)$, then $a(X) = b(X)$ or $b(X) = c$ for some $c \in R$.

Example 4. The polynomial $X^2 + 1 \in \mathbb{Z}[X]$ is irreducible over R, but the polynomial $X^2 - 1 \in \mathbb{Z}[X]$ is not irreducible, since $X^2 - 1 = (X - 1)(X + 1)$.

Definition 14 (Cyclotomic Polynomials over Polynomial Rings)**.** Let $F_p[X]$ be the polynomial ring over the finite field F of order p. Let n be a positive integer such that $p \nmid n$. Let z be be an nth primitive root of unity in F_p. The nth cyclotomic polynomial in $F[X]$ is defined as

$$Q_n(X) = \prod_{\substack{s=1 \\ \gcd(s,n)=1}} (X - z^s).$$

Note that $Q_n(X)$ is uniquely defined and at least one primitive root of unity z is guaranteed to exist [1].

Definition 15 (Polynomial Residue Class Rings)**.** The set $R[X]/(r(X))$ denotes the set of polynomial residues in $R[X]$ modulo $r(X)$. To be precise, $R[X]/r(X)$ is a set of congruence classes, where each class contains polynomials that are pairwise congruent to one another modulo $r(X)$. That is,

$$R[X]/(r(X)) = \{[a(X)] \mid a(X) \in R[X]\} \qquad \text{where}$$
$$[a(X)] = \{b(X) \in R[X] \mid a(X) \equiv b(X) \pmod{r(X)}\},$$

i.e. $[a(X)] \subseteq H[X]$ denotes the set of polynomials $b(X) \in R[X]$ congruent to $a(X)$ modulo $r(X)$.

Example 5.

$$F_2[X]/(X^2+1) = \{0, 1, X, X+1\}.$$

1.7 Combinatorics

Definition 16. The number of ways to choose a subset of k elements from a set of n elements is denoted by the choose function, $\binom{n}{k}$, where n and k are nonnegative integers and $k \leq n$. The choose function can be computed with the following formula:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}.$$

Property 4. Let n, k and r be nonnegative integers with $r \leq k \leq n$. Then

$$\binom{n}{k} \geq \binom{n-r}{k-r}.$$

2 Introduction to Primality Testing

A prime is a positive integer greater than 1 that is divisible by only 1 and itself. Prime numbers are the building blocks of number theory. For example, the Fundamental Theorem of Arithmetic states that each positive integer has a unique prime factorization, i.e. can be written as a unique product of primes [2]. As a result, two of the most fundamental questions about each positive integer n are (i) is n prime, and (ii) what is the prime factorization of n? In this paper, we will concern ourselves with answers to the first question. Procedures that determine whether n is prime or composite in finite time are called *primality tests*.

Definition 17. A *primality test* is a decision procedure that, given an $n \in \mathbb{N}$ $(n \geq 2)$ as input, returns `PRIME` if and only if n is prime and `COMPOSITE` if and only if n is composite.

In order for a primality test to be correct and useful, it needs to leverage a property that all prime numbers possess and no composite numbers possess (or vice versa). The most obvious property unique to prime numbers comes straight from their definition: every prime number is divisible by no positive integer greater than 1 and less than the prime number itself. This suggests the most obvious primality test:

Algorithm 1 Check the primality of a positive integer $n \geq 2$.

Require: $n \in \mathbb{N}$ with $n \geq 2$.
Ensure: Output `PRIME` if n is prime; otherwise output `COMPOSITE`.

```
for all 1 < k < n do
  if k | n then
    return COMPOSITE.
return PRIME.
```

The problem with this approach is that it takes up to $n-2$ divisibility checks! If n is a very large prime, then this algorithm takes unacceptably long to terminate.

Remark 1. Primality testing is not to be confused with integer factorization. Integer factorization is believed to be a more "difficult" problem than primality testing ("difficult" in a rigorous way – see Section ??? on algorithmic complexity classes.

One of the more common starting points for primality tests is the following observation made by Pierre de Fermat [2]:

Theorem 1 (Fermat's Little Theorem)**.** *Let p be prime and $a \in \mathbb{N}$ with $p \nmid a$. Then*

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

Proof. (Omitted; see [2].) □

However, note that Theorem 1 is *not* an if-and-only-if statement. As a result, it falls short of being a primality test. The following example demonstrates this.

Example 6. Take $a = 5$ and $p = 4$. Then

$$a^p = 5^{4-1} = 125 = 31 \cdot 4 + 1 \equiv 1 \pmod{4}.$$

Therefore, the unqualified converse of Theorem 1 is *not* true in general.

Strengthening our hypothesis brings us closer to a primality test.

Theorem 2. *Let $n \in \mathbb{N}$ and $a \in \mathbb{Z}$ with $\gcd(a, n) = 1$. If $a^{n-1} \equiv 1 \pmod{n}$ and for all $x \mid n-1$ with $x < n-1$, it is the case that $a^x \not\equiv 1 \pmod{n}$, then n is prime.*

Proof. Suppose $a^{n-1} \equiv 1 \pmod{n}$ and for all $x \mid n-1$ with $x < n-1$, $a^x \not\equiv 1 \pmod{n}$. Let $s = \min\left\{k \in \mathbb{N} \mid a^k \equiv 1 \pmod{n}\right\}$. Applying the Division Algorithm, we have that $n-1 = ks + r$ for some integers $k \in \mathbb{Z}$ and $0 \leq r < s$. Then

$$1 \equiv a^{n-1} = a^{ks+r} = a^{ks}a^r = (a^s)^k\, a^r \equiv 1^k a^r = a^r.$$

However, s is the minimum positive integer such that $a^s \equiv 1 \pmod{n}$, but $r < s$, so $r = 0$, and thus $s \mid n-1$. If $s < n-1$, then by the hypothesis, $a^s \not\equiv 1 \pmod{n}$, a contradiction. Therefore, $s = n-1$. But $o_n(a)$ divides $\phi(n)$ [2], but $o_n(a) = n-1$, so $\phi(n) = n-1$. It directly follows that n is prime. □

Theorem 2 is still not an if-and-only-if statement, however. This means that while the property described the hypothesis is not possessed by any composites, it need not be possessed by all primes.

Example 7. Consider $n = 5$, $a = 4$. Note that

$$a^{n-1} = 4^{5-1} = 4^4 = 256 \equiv 1 \pmod{5}.$$

However, when $x = 2$ so that $x \mid n-1 = 4$,

$$a^x = 4^2 = 16 \equiv 1 \pmod{5}.$$

Therefore, the hypothesis of Theorem 2 fails, so it says nothing about the primality of $n = 5$.

As we have seen in Example 7, applying Theorem 2 is sometimes inconclusive about the primality of a positive integer n. We can improve upon the strength of Theorem 2, though.

Theorem 3 (Lucas's Primality Test)**.** *Let $n \in \mathbb{N}$ and $a \in \mathbb{Z}$ with $\gcd(n, a)$. If for all primes $p \mid n-1$, $a^{(n-1)/p} \not\equiv 1 \pmod{n}$, then n is prime.*

Proof. Let $x \mid n-1$ with $x < n-1$ be given. Then $\frac{n-1}{x} \geq 2$, so there exists a prime $p \mid \frac{n-1}{x}$. It follows that $x \mid \frac{n-1}{p}$, i.e. $\frac{n-1}{p} = kx$ for some $k \in \mathbb{Z}$. Suppose for contradiction that $a^x \equiv 1 \pmod{n}$. Then

$$a^{(n-1)/p} = a^{kx} = (a^x)^k \equiv 1^k = 1 \pmod{n},$$

a contradiction of our hypothesis. Thus $a^x \not\equiv 1 \pmod{n}$, so we can apply Theorem 2 to get that n is prime.

□

The main advantage of Lucas's Primality Test over Theorem 2 is that fewer congruences must be tested to determine whether the hypothesis holds. However, both Theorem 3 (Lucas's Primality Test) and Theorem 2 fall short in that they *cannot* decide if positive integer n is *composite*. Another way to strengthen Lucas's Primality Test is to generalize the hypothesis to "if there exists some $a \in \mathbb{Z}$ with $\gcd(n, a) = 1 \ldots$". The problem with this stronger version is that it does not describe a decision procedure at all (at least as stated): there is an infinite number of choices for $a \in \mathbb{Z}$, but a decision procedure must terminate in a finite number of steps.

Now, we will present a successful primality test that satisfies Definition 17.

Theorem 4 (Generalization of Fermat's Little Theorem)**.** *Let $a \in \mathbb{Z}$ and $n \in \mathbb{N}$ with $\gcd(a, n) = 1$ and $n \geq 2$. n is prime if and only if $(X + a)^n \equiv X^n + a \pmod{n}$.*

Proof. First, we prove sufficiency. Suppose n is prime. Let $1 \leq i \leq n - 1$ be given. In the expansion of $(x + a)^n$, consider the term

$$\binom{n}{i} x^{n-i} a^i = \frac{n!}{(n-i)!\, i!} x^{n-i} a^i.$$

Since $\binom{n}{i}$ is a positive integer,

$$(n-i)!\, i! \mid n! = n \cdot (n-1)!, \tag{1}$$

Note that for all $1 \leq m < n$, $\gcd(n, m) = 1$, for n is prime. Since $1 \leq n - i < n$ and $1 \leq i < n$, it follows that $\gcd\left((n-i)!\, i!, n\right) = 1$. Applying Lemma 1 to equation 1, $(n-i)!\, i! \mid (n-1)!$. Therefore,

$$\binom{n}{i} = n \cdot \frac{(n-1)!}{(n-i)!\, i!} = nk.$$

We conclude that $n \mid \binom{n}{i}$, so

$$\binom{n}{i} x^{n-i} a^i \equiv 0 \pmod{n}.$$

Now, we prove necessity. Suppose n is composite. There exists a prime $p \mid n$. In the expansion of $(x + a)^n$, consider the term

$$\binom{n}{p} x^{n-p} a^p = \frac{n!}{(n-p)!\, p!} x^{n-p} a^p.$$

Define the multiplicative function $f : \mathbb{N} \to \mathbb{N}$ with $f(m) = \max\left\{k : p^k \mid m\right\}$. Note that since $p \mid n$, $p \nmid n - r$ for any $0 < r < p$. Therefore,

$$\begin{aligned} f\left((n-p)!\right) &= f(n!) - \left(f(n) + f(n-1) + f(n-2) + \cdots + f(n-p+1)\right) \\ &= f(n!) - \left(f(n) + 0 + 0 + \cdots + 0\right) \\ &= f(n!) - f(n). \end{aligned}$$

Also,

$$f(p!) = f(p) + f(p-1) + \cdots + f(1) = 1 + 0 + 0 + \cdots + 0 = 1.$$

Finally,

$$\begin{aligned} f\binom{n}{p} = f\left(\frac{n!}{(n-p)!\,p!}\right) &= f(n!) - f\left((n-p)!\right) - f(p!) \\ &= f(n!) - f(n!) + f(n) - 1 \\ &= f(n) - 1. \end{aligned}$$

Then by definition of f, $p^{f(n)-1} \mid \binom{n}{p}$ but $p^{f(n)} \nmid \binom{n}{p}$. However, $p^{f(n)} \mid n$, so $n \nmid \binom{n}{p}$. All that remains to be shown is that $n \nmid \binom{n}{p} a^p$. By hypothesis, $\gcd(a, n) = 1$, so $\gcd(a^p, n) = 1$ as well. By the contrapositive of Lemma 1, it follows that $n \nmid \binom{n}{p} a^p$. Therefore,

$$\binom{n}{p} x^{n-p} a^p \not\equiv 0 \pmod{n}.$$

□

Algorithm 2 Primality test based on Theorem 4.

Require: $n \in \mathbb{N}$ with $n \geq 2$.
Ensure: Output `PRIME` if n is prime; otherwise output `COMPOSITE`.

```
Find an a ∈ ℤ such that gcd(a, n) = 1.
if (X + a)^n ≡ X^n + a (mod n) then
    return PRIME.
else
    return COMPOSITE.
```

Theorem 4 describes a primality test because (i) it is an if-and-only-if statement and (ii) it describes a procedure that will terminate in a finite number of steps. For an example of such a decision procedure, see Algorithm 2. The downside of this primality test is that it extremely is inefficient to compute $(X+a)^n$ modulo n for large n, even with binary modular exponentiation [3]. Multiplying two polynomials of order $O(n)$ is a $O(n^2)$ operation in the worst case. Before we proceed with our search for more efficient primality tests, we need to give a concrete characterization of what we mean by "efficient" in the first place.

3 Introduction to Complexity Classes

As has become apparent so far, some algorithms are more efficient than others. However, to compare their efficiency, we need to define what "efficiency" in the context of an algorithm means in the first place. From now on, assume the domain of each problem is the positive

integers and the range is a single bit, i.e. a yes or no (or PRIME or COMPOSITE) answer. We can view problems as functions, where each input $n \in \mathbb{N}$ is mapped to the answer bit, yes or no. Now, we define the problem regarding primality testing.

Definition 18. Let $\texttt{PRIMES} : (\mathbb{N} + 1) \to \{\texttt{PRIME}, \texttt{COMPOSITE}\}$ denote the function that maps a positive integer $n \geq 2$ to PRIME if it is prime or COMPOSITE if it is composite. That is,

$$\texttt{PRIMES}(n) = \begin{cases} \texttt{PRIME} & \text{if } n \text{ is prime;} \\ \texttt{COMPOSITE} & \text{otherwise.} \end{cases}$$

Each problem has an intrinsic difficulty. While there are an infinite number of algorithms with varying efficiency that can solve a given problem, there is a theoretical lower bound on the runtime of any algorithm solving a given problem.

There are an infinite number of algorithms (i.e. decision procedures) that solve the problem PRIMES , but note that each algorithm is distinct from the problem PRIMES itself.

We can parameterize runtime of each of these algorithms in the input n to give us an estimate of how the number of steps taken by the algorithm grows with larger n, called the *asymptotic time complexity*. However, we don't care about the exact asymptotic function, just an upper bound written in big-O notation [4].

Definition 19. An algorithm $\mathcal{A}$ runs in polynomial time in the length of input n if the the number of steps taken by $\mathcal{A}$ on n is $O(\log^k n)$ for some $k \geq 0$.

Remark 2. Time complexity of algorithms is measured as a function of the input length and *not* a function of the input itself. The "length" of a positive integer n is $\lceil \log n \rceil$, where the logarithm base is generally taken to be two, since it takes $\lceil \log_2 \rceil$ bits to represent n in base two. We can omit the ceiling function in Definition 19 since $\lceil \log n \rceil < \log n + 1$ and big-O notation is forgiving.

Definition 20. P is the set of problems that are solvable by some algorithm in polynomial time in the length of the input n.[1]

Remark 3. Note that Definition 20 involves an existence statement; that is, one needs to only find one algorithm whose runtime is $O(\log^k n)$.

Both primality testing algorithms we have presented so far do have a polynomial runtime in $\log n$. For example, Algorithm 2 involves $n - 2$ divisibility checks in the worst case, and $n - 2$ is not $O(\log^k n)$. Also, Algorithm 2 involves a worst-case multiplication of two polynomials of order n, which involves reading $2n$ different coefficients, so we can quickly see it cannot be $O(\log^k n)$.

The question naturally arises, does there exist a primality test that determines the answer in polynomial time? Until recently, the answer to this was unknown (but believed to

[1] For a more rigorous definition of the complexity class P, see [5].

be yes). In 1976, Gary Miller presented a primality test and proved its polynomial runtime under the unproven assumption of the Extended Riemann Hypothesis [6]. Furthermore, probabilistic primality tests with polynomial runtime, such as the APR Primality Test [7], were presented in the late 20th century. However, it was not until 2002 when Manindra Agrawal, Neeraj Kayal, and Nitin Saxena presented the first deterministic primality test that provably runs in polynomial time with no additional assumptions. In the next section, we will present their algorithm, the AKS Primality Test, and prove its correctness. In the section thereafter, we will show its runtime is polynomial.

4 The AKS Primality Testing Algorithm

In this section, we present the AKS Primality Test and prove its correctness. From now on, assume $\log n$ is the base-2 logarithm of n.

4.1 The AKS Algorithm

Algorithm 3 Check the primality of a positive integer $n \geq 2$.

Input: $n \in \mathbb{N}$ with $n \geq 2$.

Output: Output `PRIME` when n is prime; output `COMPOSITE` when n is composite.

if $n = a^b$ for some $a, b \in \mathbb{N}$ with $b \geq 2$ **then**
  return `COMPOSITE`
Find the smallest $r \in \mathbb{N}$ such that $o_r(n) > \log^2(n)$.
if $1 < \gcd(a, n) < n$ for some $a \leq r$ **then**
  return `COMPOSITE`
if $n \leq r$ **then**
  return `PRIME`
Let p be a prime divisor of n such that $o_r(p) > 1$.
Let $l = \lfloor \sqrt{\phi(r)} \log n \rfloor$.
for all $0 \leq a \leq l$ **do**
  if $(X + a)^n \not\equiv X^n + a \pmod{X^r - 1, n}$ **then**
    return `COMPOSITE`
return `PRIME`

Theorem 5 (Correctness of the AKS Algorithm)**.** *Algorithm 3 returns* `PRIME` *if and only if the input n is prime.*

4.2 Proof of Correctness

We prove the correctness of the AKS primality test (Theorem 5) by proving that at each step of Algorithm 3, if it terminates, it outputs the correct answer.

Lemma 1. *If Algorithm 3 terminates by line 2, then it returns* `PRIME` *if and only if n is prime.*

Proof. If $n = a^b$ for some $a, b \in \mathbb{N}$ with $b \geq 2$, clearly n is composite. The algorithm correctly terminates, returning `COMPOSITE`. □

Before we proceed, we must first prove that finding such an r in line 3 of the AKS algorithm is possible.

Lemma 2. *In Algorithm 3, line 3 describes a valid computation. Furthermore, there exists an $r \in \mathbb{N}$ such that*

$$\begin{aligned} r &\leq \max\left\{3, \lceil \log^5 n \rceil\right\} && \textit{and} \\ o_r(n) &> \log^2 n && \textit{(Definition 7).} \end{aligned}$$

Proof. Let $n \geq 2$, be given. Let $B = \lceil \log^5 n \rceil$. Consider the product

$$P = n \cdot \prod_{i=1}^{\lfloor \log^2 n \rfloor} \left(n^i - 1\right).$$

We will now show $P < n^{\log^4 n} = 2^{\log^5 n}$. To show that $P < n^{\log^4 n}$,

$$\begin{aligned} P &= n \cdot \prod_{i=1}^{\lfloor \log^2 n \rfloor} \left(n^i - 1\right) \\ &< n \cdot \prod_{i=1}^{\lfloor \log^2 n \rfloor} n^i \\ &\leq n \cdot n^{\lfloor \log^2 n \rfloor - 1} \cdot \prod_{i=2}^{\lfloor \log^2 n \rfloor} n^{\lfloor \log^2 n \rfloor} && \text{since } \lfloor \log^2 n \rfloor - 1 \geq 1 \text{ when } n > 2; \\ &\leq n^{\lfloor \log^2 n \rfloor} \cdot n^{\lfloor \log^2 n \rfloor \left(\lfloor \log^2 n \rfloor - 1\right)} \\ &= n^{\lfloor \log^2 n \rfloor \cdot \lfloor \log^2 n \rfloor} \\ &\leq n^{\lfloor \log^4 n \rfloor} \\ &\leq n^{\log^4 n}. \end{aligned}$$

To show that $n^{\log^4 n} = 2^{\log^5 n}$, note that

$$\begin{aligned} n^{\log^4 n} &= 2^{\log n \log^4 n} \\ &= 2^{(\log n)\left(\log^4 n\right)} \\ &= 2^{\log^5 n}. \end{aligned}$$

Now, define the set

$$R = \left\{ r \in \mathbb{N} : r \mid n \text{ or } o_r(n) \leq \log^2 n \right\}.$$

The curious term $(n^i - 1)$ in product P comes from a straightforward property of group order:

$$\begin{aligned} & n^{o_r(n)} \equiv 1 \pmod{r} \\ \iff \quad & r \mid \left(n^{o_r(n)} - 1 \right). \end{aligned}$$

Therefore, if $r \in R$ and $o_r(n) \leq \log^2 n$, then $r \mid P$. It then follows that each $r \mid P$ for all $r \in R$.

Now, we find some positive integer $s \notin R$ with $s \leq \lceil \log^5 n \rceil$. Note that since $\lceil \log^5 n \rceil \geq 10 \geq 7$, it follows from Lemma 13 that

$$\text{LCM}\left(\lceil \log^5 n \rceil \right) \geq 2^{\lceil \log^5 n \rceil}.$$

However, $P < 2^{\lceil \log^5 n \rceil}$, so there exists some positive integer $s \leq \lceil \log^5 n \rceil$ such that $s \notin R$. (Otherwise, $\text{LCM}\left(\lceil \log^5 \rceil \right) \mid P$ but $P < \text{LCM}\left(\lceil \log^5 \rceil \right)$, a contradiction.) Therefore, $o_s(n) > \log^2 n$.[2]

We conclude that there exists a least positive integer r such that $o_r(n) > \log^2 n$.

□

Now, we can move on to proving the correctness of the next line in the algorithm.

Lemma 3. *If Algorithm 3 terminates by line 5, then it returns* `PRIME` *if and only if n is prime.*

Proof. Suppose there exists a positive integer $a \leq r$ such that $1 < \gcd(a, n) < n$. Since $\gcd(a, n) \mid n$, $n = k \gcd(a, n)$ for some $k \in \mathbb{N}$. Therefore n is composite, and the algorithm correctly proceeds to return `COMPOSITE` on line 5. □

Lemma 4. *If Algorithm 3 reaches line 6 and terminates by line 7, then it returns* `PRIME` *if and only if n is prime.*

Proof. Suppose $n \leq r$. First, consider the case when n is composite. There exists a positive integer $1 < a < n \leq r$ such that $a \mid n$. That is, $1 < a \leq \gcd(a, n) < n$, so the algorithm terminates at line 5, never reaching line 6. Otherwise, n is prime. In this case, the algorithm reaches 6 and proceeds to line 7 and correctly returns `PRIME`. □

Lemma 5. *If Algorithm 3 reaches line 8, there exists a prime p that divides n with $o_r(p) > 1$. Furthermore, $p > r$ and $\gcd(p, r) = 1$.*

[2] [8] considers when $\gcd(s, n) = 1$ and $\gcd(s, n) > 1$ separately, but I cannot figure out why that would be necessary.

Proof. Suppose the algorithm reaches line 8. Then the if-predicate on line 6 was not satisfied, so $n > r$. Suppose for contradiction that all primes p_i dividing n have order $o_r(p) = 1$. Then

$$n = p_1 p_2 \cdots p_t \equiv 1 \cdot 1 \cdot \cdots \cdot 1 = 1 \pmod{r},$$

so $o_r(n) = 1$ as well. However, r was computed to have the property that $o_r(n) > \log^2 n \geq 1$, a contradiction. Therefore, there exists a prime divisor p of n such that $o_r(p) > 1$. Now, suppose for contradiction that $p \leq r$. If $p = n$, then the algorithm terminated on line 7, a contradiction. Otherwise, $1 < p < n$, so $1 < \gcd(p, n) = p < n$. Then the algorithm terminated on line 5, a contradiction. We conclude that $p > r$. Finally, note that since $r < n$, and the if-predicate on line 4 was not satisfied, $\gcd(r, n) = 1$. But since $p \mid n$, $\gcd(r, p) \mid \gcd(r, n) = 1$, so $\gcd(r, p) = 1$ as well. □

Lemma 6. *If Algorithm 3 reaches line 10 and terminates by line 12, then it returns* `PRIME` *if and only if n is prime.*

Proof. If the algorithm reaches line 12, then the if-predicate of line 10 was satisfied for some $0 \leq a \leq \lfloor \phi(r) \log n \rfloor$. That is,

$$(X + a)^n \not\equiv X^n + a \pmod{X^r - 1, n}.$$

It directly follows that

$$(X + a)^n \not\equiv X^n + a \pmod{n}.$$

However, since the algorithm has reached line 10, the if-predicate of line 4 was not satisfied, so $\gcd(a, n) = 1$. Therefore, Lemma 4 applies, so we conclude that n is composite and thus the algorithm correctly returns `COMPOSITE`. □

Lemma 7. *If Algorithm 3 terminates at line 13, then n is prime.*

The proof of Lemma 7 is the most difficult. Before we can prove it, we will need to obtain a few results that characterize the behavior of the previous for-loop on line 10.

4.3 Introspective Numbers

We introduce the following useful concept for proving Lemma 7.

Definition 21. Let $A(X) \in F_p[X]$ be a polynomial and m be a positive integer. Call m *introspective*[3] for $A(X)$ if

$$(A(X))^m \equiv A(X^m) \pmod{X^r - 1, p}.$$

[3] [8] introduces this term for proving the correctness of the AKS algorithm.

Note that if the algorithm does not terminate until line 13, then n is introspective for all $X + a$ when $0 \leq a \leq l$, since

$$\begin{aligned} & (X+a)^n \equiv X^n + a \pmod{X^r - 1, n} \\ \implies \quad & (X+a)^n \equiv X^n + a \pmod{X^r - 1, p} \quad \text{since } p \mid n. \end{aligned}$$

Now, we'll introduce and prove a couple of properties about introspective numbers, namely (i) that the set of positive integers introspective for a given polynomial is closed under multiplication and (ii) that the set of polynomials for which a given positive integer is introspective is closed under multiplication.

Lemma 8. *If a and b are introspective for some $A(X) \in F_p[X]$, then their product ab is introspective for $A(X)$ as well.*

Proof. Let $A(X) \in F_p[X]$ and $a, b \in \mathbb{N}$ be given such that a, b are introspective for $A(X)$. Then

$$\begin{aligned} A\Big(x^{ab}\Big) = A\Big((X^a)^b\Big) = A\Big(Y^b\Big) & & & \text{where } Y = X^a; \\ \equiv (A(Y))^b & \pmod{Y^r - 1, p} & & \text{since } b \text{ is introspective for } A(Y) \\ \equiv (A(Y))^b & \pmod{X^{ar} - 1, p} & & \text{since } Y = X^a \\ \equiv (A(Y))^b & \pmod{X^r - 1, p} & & \text{since } X^{ar} - 1 = (X^r - 1) \cdot \sum_{i=0}^{a-1} X^i \\ = (A(X^a))^b & & & \text{since } Y = X^a \\ \equiv A(X)^{ab} & \pmod{X^r - 1, p} & & \text{since } a \text{ is introspective for } A(X). \end{aligned}$$

□

Lemma 9. *If m is introspective for polynomials $A(X), B(X) \in F_p[X]$, then m is introspective for $A(X) \cdot B(X)$ as well.*

Proof. Since m introspective for $A(X)$ and $B(X)$,

$$\begin{aligned} (A(X))^m \equiv A(X^m) \pmod{X^r - 1, p} \\ (B(X))^m \equiv B(X^m) \pmod{X^r - 1, p}. \end{aligned}$$

Multiplying the respective sides of each congruence together, we have

$$(A(X))^m (B(X))^m = (A(X)B(X))^m \equiv A(X^m)B(X^m) = (AB)(X^m) \pmod{X^r - 1, p}.$$

□

Now, we will define two sets, I and P, and then prove a couple of their properties related to introspectiveness.

Definition 22. Define the sets

$$I = \left\{ \left(\frac{n}{p}\right)^i p^j : i, j \geq 0 \right\}$$
$$P[X] = \left\{ \prod_{a=0}^{l} (X+a)^{s_a} : s_a \geq 0 \right\}.$$

Lemma 10. *If Algorithm 3 reaches line 13, then for every $m \in I$ and $A(X) \in P[X]$, m is introspective for $A(X)$.*

Proof. We first show that p and $\frac{n}{p}$ are introspective for all $X+a$ with $0 \leq a \leq l$ and then we apply Lemmas 8 and 9 to obtain the result.

Let $0 \leq a \leq l$ be given. First, note that since the algorithm reaches line 13,

$$\begin{aligned} & (X+a)^n \equiv X^n + a \pmod{X^r - 1, n} \\ \implies \quad & (X+a)^n \equiv X^n + a \pmod{X^r - 1, p} \quad \text{since } p \mid n. \end{aligned} \tag{2}$$

Since p is prime, by Lemma 4 we have that

$$\begin{aligned} & (X+a)^p \equiv X^p + a \pmod{p} \\ \implies \quad & (X+a)^p \equiv X^p + a \pmod{X^r - 1, p}. \end{aligned}$$

We claim that any pth root of $X^p + a$ is congruent to $X + a$ modulo $X^r - 1$ and p. To show this, let $C(X) = c_0 + c_1 X + c_2 X^2 + \cdots + c_t X^t$ such that $c_t \not\equiv 0 \pmod{p}$ and $t < r$, and suppose

$$(C(X))^p = \left(c_0 + c_1 X + \cdots + c_t X^t\right)^p \equiv X^p + a \pmod{X^r - 1, p}.$$

Note that since each $c_i \not\equiv 0 \pmod{p}$ and p is prime, $\gcd p, c_i = 1$. Let

$$Y = c_1 X + \cdots + c_t X^t = X(c_1 + \cdots + c_t X^{t-1}).$$

Then

$$\begin{aligned} C(X)^p &= \left(c_0 + c_1 X + \cdots + c_t X^t\right)^p \\ &= \left(c_0 + \left(c_1 X + \cdots + c_t X^t\right)\right)^p \\ &= (c_0 + Y)^p \\ &\equiv c_0 + Y^p \pmod{p} \qquad \text{since } c_0 \equiv 0 \pmod{p} \text{ or } \gcd(c_0, p) = 1, \text{ in which case Theorem 4 appl} \\ &= c_0 + \left(c_1 X + \cdots + c_t X^t\right)^p. \end{aligned}$$

This process can be repeated t times to obtain that

$$\begin{aligned} C(X)^p &\equiv c_0 + c_1X^p + c_2X^{2p} + \cdots + c_tX^{pt} \pmod{p}, && \text{so} \\ a + X^p &\equiv c_0 + c_1X^p + \cdots + c_tX^{pt} && \pmod{X^r - 1, p}. \end{aligned}$$

It must be the case that the order of c_tX^{pt} (pt) is congruent to the order of a (0) or X^p (p) modulo r. Suppose for contradiction that $pt \equiv 0 \pmod{r}$, so that $r \mid pt$. Since $\gcd(p, r) = 1$, $r \mid t$. However, $t < r$, so $t = 0$. But then $C(X) = c_0$, and then $C(X)^p = c_0^p \equiv c_0 \not\equiv a + X^p \pmod{X^r - 1, p}$, a contradiction. Therefore, $pt \equiv p \pmod{r}$, so that $r \mid p(t-1)$. However, $t - 1 < r$, so $t - 1 = 0$ and thus $t = 1$. It follows that $C(X) = c_0 + c_1X$, and

$$c_0 + c_1X^p \equiv a + X^p \pmod{X^r - 1, p}.$$

Clearly, $c_0 \equiv a \pmod{p}$ and $c_1 \equiv 1 \pmod{p}$. We conclude that any pth root $C(X)$ is congruent to

$$C(X) \equiv a + X \pmod{X^r - 1, p}.$$

Now, let $Y = X^{n/p}$. Then

$$\begin{aligned} & (Y + a)^p \equiv Y^p + a && \pmod{X^r - 1, p} \\ \iff \quad & (X^{n/p} + a)^p \equiv (X^{n/p})^p + a && \pmod{X^r - 1, p} \\ \iff \quad & (X^{n/p} + a)^p \equiv X^n + a && \pmod{X^r - 1, p}. \\ & \equiv (X + a)^n && \pmod{X^r - 1, p} \qquad \text{by Equation 2.} \end{aligned}$$

But any pth root of $Y^p + a = \left(X^{n/p}\right)^p + a$ is congruent to $Y + a = X^{n/p} + a$, so we conclude that

$$X^{n/p} + a \equiv (X + a)^{n/p} \pmod{X^r - 1, p}.$$

Therefore, n is introspective for $X + a$.

To recap, we have now shown that p and n/p are introspective for $X + a$ for all $0 \leq a \leq l$. Now, let an arbitrary polynomial $A(x) = \prod_{a=0}^{l}(X + a)^{s_a} \in P[X]$ be given. Since p and n/p are introspective for factor $X + a$, inductively applying Lemma 9 gives us that p and n/p are introspective for $A(x)$. Now, let an arbitrary integer $m = p^i(n/p)^j \in I$ be given. Since p and n/p are introspective for $A(x)$, by inductively applying Lemma 8, we have that m is also introspective for $A(x)$.

□

Definition 23. Let G be the group of residues in I modulo r, i.e.

$$G = \{s \in \mathbb{Z}_r \mid \exists\, m \in I \text{ such that } s \equiv m \pmod{r}\}.$$

Let $|G| = t$.

Definition 24. There exists a $Q'_r(X) \in F_p[X]$ congruent to the rth cyclotomic polynomial modulo p. Note that while the rth cyclotomic polynomial is irreducible in $\mathbb{Z}[X]$, $Q'_r(X)$ is not necessarily reducible in $F_p[X]$ due to the finiteness of the field. Let $h(X) \in F_p[X]$ be an irreducible polynomial divisor of $Q'_r(X)$ modulo p. Define the set $F = F_p[X]/h(X)$. Note that F is a field under polynomial addition and multiplication modulo $h(X)$ and p. Now, define the group

$$\mathcal{G} = \{A(X) \in F_p[X] \mid A(X) \equiv B(X) \pmod{h(X), p} \text{ where } B(X) \in P\}.$$

That is, $\mathcal{G}$ is the set of residues of polynomials in P modulo $h(X)$ and p.

A couple of remarks about the group $\mathcal{G}$: Note that since P is generated by polynomials of the form $X + a$ where $0 \leq a \leq l$, so is $\mathcal{G}$. Furthermore, we claim that $\mathcal{G}$ is indeed a group, but do not provide a proof.

This group $\mathcal{G}$ is the crux of the AKS algorithm. The upper bound on the size of group $\mathcal{G}$ indicates the maximum number of values for a you need to check in the congruence

$$(X + a)^n \stackrel{?}{\equiv} X^n + a \pmod{X^r - 1, n}$$

on line 10 of the algorithm. Now, we will prove a lower bound on the size of group $\mathcal{G}$. In order for the AKS algorithm to terminate in time polynomial on $\log n$, we cannot check more than $O(\log^k n)$ different values of a, so the size of $\mathcal{G}$ must also be $O(\log^k n)$. This is what we will prove next.

Lemma 11. *The group $\mathcal{G}$ contains at least $\binom{t+1}{t-1}$ elements, i.e.*

$$|\mathcal{G}| \geq \binom{t+1}{t-1}.$$

Proof. First, we show that X is a rth primitive root of unity in F. Note that

$$\begin{aligned} & X^r \equiv 1 \pmod{X^r - 1, p} & \\ \implies \quad & X^r \equiv 1 \pmod{Q_r(X), p} & \text{since } Q_r(X) \mid X^r - 1 \\ \implies \quad & X^r \equiv 1 \pmod{h(X), p} & \text{since } h(X) \mid Q_r(X). \end{aligned}$$

Therefore, X is an rth root of unity. Furthermore, X is primitive because for any $s \mid t$ and $1 \leq s < t$, $\gcd(Q_r(X), X^s - 1) = 1$, since any rth root of unity is not a sth root of unity (Definition 14).

We'll now show that distinct polynomials of degree less than $|G| = t$ are not congruent in F. Let distinct polynomials $f(X), g(X) \in P[X]$ be given such that $\deg f(X), \deg g(X) < t$. Suppose for contradiction that $f(X) \equiv g(X) \pmod{h(X), p}$. For each $m \in I$,

$$\begin{aligned} & f(X) \equiv g(X) \pmod{h(X), p} \\ \implies \quad & (f(X))^m \equiv (g(X))^m \pmod{h(X), p}. \end{aligned}$$

Furthermore, since m is introspective for $f(X)$ and $g(X)$ by Lemma 10,

$$\begin{aligned}(f(X))^m \equiv f(X^m) \pmod{X^r-1,p} \quad &\implies \quad (f(X))^m \equiv f(X^m) \pmod{h(X),p} \quad \text{and} \\ (g(X))^m \equiv g(X^m) \pmod{X^r-1,p} \quad &\implies \quad (g(X))^m \equiv g(X^m) \pmod{h(X),p},\end{aligned}$$

since $h(X) \mid X^r - 1$. Therefore, $f(X^m) \equiv g(X^m) \pmod{h(X),p}$. Furthermore, each $m \in I$ is relatively prime to p and since X is a pth root of unity, each X^m is also a pth root of unity. Therefore, the polynomial

$$f(X^m) - g(X^m) = (f-g)(X^m) \equiv 0 \pmod{h(X),p}$$

has $|G| = t$ distinct roots in F_p. It follows that $\deg(f-g)(X^m) \geq t$, so $\deg f(X^m) \geq t$ or $\deg g(X^m) \geq t$, a contradiction. We conclude that

$$f(X) \not\equiv g(X) \pmod{h(X),p}.$$

Now, we can supply a lower bound on the number of polynomials in group $\mathcal{G}$. First, we will need the following inequality:

$$o_r(n) \leq r. \tag{3}$$

To show this holds, note that the least positive integer $i = o_r(n)$ such that $n^i \equiv 1 \pmod{r}$ must be less than r by the Pigeonhole Principle. It then follows that

$$\begin{aligned}l &= \sqrt{\phi(r)}\log n \\ &< \sqrt{r}\log n && \text{since } \phi(k) < k \text{ for all } k \in \mathbb{N} \\ &< \sqrt{r}\cdot\sqrt{o_r(n)} && \text{by selection of } r \text{ in line 3 of Algorithm 3} \\ &\leq \sqrt{r}\cdot\sqrt{r} && \text{from Inequality 3} \\ &= r.\end{aligned}$$

Now, we'll show there are at least $l+1$ polynomials of degree one in $\mathcal{G}$. Note that for all $0 \leq i \leq l$, $X + i \in P$ and $0 \leq i < l < r < p$. Furthermore, $\deg h(X) > 1$, so $X + i \in \mathcal{G}$ by Definition 24. Further note that for all $0 \leq i, j \leq l$,

$$\begin{aligned}& X + i \equiv X + j \pmod{h(X),p} \\ \iff \quad & i \equiv j \pmod{p} \\ \iff \quad & i = j && \text{since } 0 \leq i, j < r < p.\end{aligned}$$

Furthermore, since $\deg h(X) > 1$, every $X + i \not\equiv 0 \pmod{h(X),p}$. Since there are $l+1$ choices for $0 \leq i \leq l$ in the polynomial $X + i$, there are at least $l+1$ polynomials of degree one in $\mathcal{G}$.

Finally, we will show a lower bound on the number of polynomials in $\mathcal{G}$. Consider a polynomial of the form

$$A(X) = \prod_{a=0}^{l} (X + a)^{s_a}$$

where each s_a is a nonnegative integer. In the first part of this proof, we found that there is a one-to-one correspondence between the set of polynomials of degree less than t and the set of their corresponding residues modulo $h(X)$ and p. Therefore, a lower bound on the number of polynomials of degree less than t also provides a lower bound on the number of polynomial residues in $\mathcal{G}$. To simplify things, we will just count the number of polynomials $A(X) \in P[X]$ with degreeexactly $t - 1$. Note that the number of polynomials $A(X) = \prod_{a=0}^{l} (X + a)^{s_a} \in P[X]$ with $\deg A(X) = t - 1$ is equal to the number of solutions to the equation

$$\sum_{a=0}^{l} s_a = t - 1 \qquad \text{where each } s_a \text{ is a nonnegative integer.} \tag{4}$$

We have $l + 1$ variables $s_0, s_1, \ldots, s_l$ over which to distribute a sum of $t - 1$. Applying the "stars and bars" technique from combinatorics [9], there are precisely

$$\binom{(l+1) + (t-1)}{t-1} = \binom{t+l}{t-1}$$

solutions to Equation 4. It follows that there are exactly $\binom{t+l}{t-1}$ polynomials of degree $t - 1$ in $P[X]$, each of which map to a distinct polynomial residue modulo $h(X)$ and p in $\mathcal{G}$. We conclude that

$$|\mathcal{G}| \geq \binom{t+l}{t-1}.$$

□

Now that we have proven a lower bound on the cardinality of $\mathcal{G}$, we will prove a conditional upper bound on the cardinality of $\mathcal{G}$.

Lemma 12. *If n is not a power of p, then $|\mathcal{G}| \leq n^{\sqrt{t}}$.*

Proof. Define the set

$$I' = \left\{ \left(\frac{n}{p}\right)^{i} \cdot p^{j} \mid 0 \leq i, j \leq \lfloor \sqrt{t} \rfloor \right\}.$$

Note that $I' \subset I$. Suppose n is not a power of p. Then there exists a prime $q \mid n$ distinct from p, and let $\alpha \in \mathbb{N}$ be the power of q in the prime factorization of n (that is, such that $q^{\alpha} \mid n$ but $q^{\alpha+1} \nmid n$). To show that each (i, j) pair produces a distinct element in I', let

$z_1 = \left(\frac{n}{p}\right)^{i_1} \cdot p^{j_1} \in I'$ and $z_2 = \left(\frac{n}{p}\right)^{i_2} \cdot p^{j_2} \in I'$ be given such that $z_1 = z_2$. It follows that $\alpha_p(z_1) = \alpha_p(z_2)$ and $\alpha_q(z_1) = \alpha_q(z_2)$. But

$$\begin{aligned}
\alpha_q(z_1) &= \alpha_q\left(\left(\frac{n}{p}\right)^{i_1} p^{j_1}\right) \\
&= \alpha_q\left(\left(\frac{n}{p}\right)^{i_1}\right) \\
&= \alpha_q\big(n^{i_1}\big) \\
&= i_1 \alpha_q(n).
\end{aligned}$$

Similarly, $\alpha_q(z_1) = i_2\alpha_q(n)$. Since $\alpha_q(z_1) = \alpha_q(z_2)$, $i_1\alpha_q(n) = i_2\alpha_q(n)$, so $i_1 = i_2$. Let $i = i_1 = i_2$. Now,

$$\begin{aligned}
& & z_1 &= z_2 & \\
\implies & & \left(\frac{n}{p}\right)^{i} p^{j_1} &= \left(\frac{n}{p}\right)^{i} p^{j_2} & \\
\implies & & p^{j_1} &= p^{j_2} & \\
\implies & & j_1 &= j_2 & \text{since } p > 1.
\end{aligned}$$

Therefore, $(i_1, j_1) = (i_2, j_2)$. Since there are $\lfloor\sqrt{t}\rfloor + 1$ choices for each i and j, each of which correspond to a distinct integer $z \in I'$, so there are a total of

$$\begin{aligned}
\big|I'\big| &= \Big(\lfloor\sqrt{t}\rfloor + 1\Big) \cdot \Big(\lfloor\sqrt{t}\rfloor + 1\Big) \\
&= \Big(\lfloor\sqrt{t}\rfloor + 1\Big)^2 \\
&> t && \text{since } \lfloor\sqrt{t}\rfloor + 1 > (t-1) + 1 = t
\end{aligned}$$

integers in I'. However, the group G contains only t elements, so there exists at least one pair of distinct integers $z_1, z_2 \in I'$ such that

$$z_1 \equiv z_2 \pmod{r},$$

where $z_1 > z_2$. By definition of modular congruence, $r \mid (z_1 - z_2)$, so $z_1 - z_2 = km$ for some nonnegative $k \in \mathbb{Z}$. Now, to show that $X^{z_1} \equiv X^{z_2} \pmod{X^r - 1}$, first note that

$$Y - 1 \mid Y^m - 1 = (Y-1)(Y^{m-1} + Y^{m-2} + \cdots + 1).$$

Substituting X^r in for Y gives

$$\begin{aligned}
& & X^r - 1 \mid X^{rm} - 1 \\
\implies & & X^r - 1 \mid X^{z_2}(X^{rm} - 1) \\
\implies & & X^r - 1 \mid X^{z_2}(X^{z_1 - z_2} - 1) \\
\implies & & X^r - 1 \mid X^{z_1} - X^{z_2} \\
\implies & & X^{z_1} \equiv X^{z_2} \pmod{X^r - 1}. \qquad (5)
\end{aligned}$$

Now, consider the polynomial $Q'(Y) = Y^{z_1} - Y^{z_2}]$. We will count the polynomial roots of $Q'(Y)$ in F, i.e. the number of symbolic substitutions $Y = A(X)$ such that $Q'(A(X)) \equiv 0 \pmod{h(X), p}$. This will give us an upper bound for $|\mathcal{G}|$. Let $A(X) \in \mathcal{G}$ be arbitrary. It follows that

$$\begin{aligned}
& (A(X))^{z_1} \equiv A(X^{z_1}) && \pmod{X^r - 1, p} && \\
\implies & \equiv A(X^{z_2}) && \pmod{X^r - 1, p} && \text{from Congruence 5} \\
\implies & \equiv (A(X))^{z_2} && \pmod{X^r - 1, p} && \\
\implies & (A(X))^{z_1} \equiv (A(X))^{z_2} && \pmod{h(X), p}. && (6)
\end{aligned}$$

We then have that

$$Q'(A(X)) = (A(X))^{z_1} - (A(X))^{z_2} \equiv 0 \pmod{h(X), p} \qquad \text{by Congruence 6.}$$

Therefore, $A(X)$ is a root of $Q'(Y)$ in F. However, $A(X)$ is an arbitrary element of $\mathcal{G}$, so $Q'(Y)$ has at least $|\mathcal{G}|$ distinct polynomial roots in F. It follows that

$$\deg Q'(Y) = \deg\left(Y^{z_1} - Y^{z_2}\right) = z_1 \geq |\mathcal{G}| \qquad (7)$$

since $z_1 > z_2$. However, since $z_1 \in I'$,

$$z_1 \leq \max I' = \left(\left(\frac{n}{p}\right)p\right)^{\lfloor\sqrt{t}\rfloor} = n^{\lfloor\sqrt{t}\rfloor}. \qquad (8)$$

Combining Inequalities 7 and 8, we have that

$$|\mathcal{G}| \leq n^{\lfloor\sqrt{t}\rfloor}.$$

□

4.4 Proof of Correctness: Part II

In the previous section, we spent a while developing necessary theory about how polynomial rings and the group of polynomial residues $\mathcal{G}$ behave when the algorithm reaches line 13 of the AKS Algorithm (Algorithm 3). Now, we will get to see our work pay off in the following quick proof of the algorithm's correctness when it reaches the last line, 13. (The following is a restatement of Lemma 7.)

Lemma 7. *If Algorithm 3 terminates at line 13, then n is prime.*

Proof. Suppose that Algorithm 3 reaches line 13, and suppose for contradiction that the input n is *not* a power of a prime. Then Lemmas 12 and 11 apply, so we have that

$$\binom{t+l}{t-1} \leq |\mathcal{G}| \leq n^{\lfloor t \rfloor}.$$

Let's now establish a couple of inequalities we'll need to use shortly. Firstly,

$$\begin{aligned}
& \log^2 n < o_r(n) && \text{by our choice of } r \\
& \leq t && \text{since there are } o_r(n) \text{ residues of } n^i \pmod r \text{ in } I \\
\implies \quad & \log n < \sqrt{t} \\
\implies \quad & \sqrt{t} \log n < t \\
\implies \quad & \lfloor \sqrt{t} \log n \rfloor < t \\
\implies \quad & \lfloor \sqrt{t} \log n \rfloor + 1 \leq t. && (9)
\end{aligned}$$

Then

$$\binom{t+l}{t-1} > \binom{\lfloor \sqrt{t} \log n \rfloor + l + 1}{(\lfloor \sqrt{t} \log n \rfloor)} \qquad \text{by Property 4 and Inequality 9}^4 \qquad (10)$$

Also, note that since each $i \in I$ is relatively prime to r, it follows that each residue $i' \in G$ is also relatively prime to r. Therefore,

$$\begin{aligned}
& l = \left\lfloor \sqrt{\phi(r)} \log n \right\rfloor \geq \left\lfloor \sqrt{t} \log n \right\rfloor \\
\implies \quad & \binom{\lfloor \sqrt{t} \log n \rfloor + l + 1}{\lfloor \sqrt{t} \log n \rfloor} \geq \binom{2\lfloor \sqrt{t} \log n \rfloor + 1}{\lfloor \sqrt{t} \log n \rfloor} \qquad \text{by Property 4.} \qquad (11)
\end{aligned}$$

For our last intermediate inequality, temporarily let $d = \lfloor \sqrt{t} \log n \rfloor$ for clarity. Then

$$\begin{aligned}
\binom{2d+1}{d} &= \frac{(2d+1)!}{d!\,(d+1)!} \\
&= \frac{(d+2)(d+3)\cdots(2d+1)}{1 \cdot 2 \cdot \cdots \cdot d} \\
&= \prod_{i=1}^{d} \frac{d+1+i}{i} \\
&= \prod_{i=1}^{d} \left(\frac{d+1}{i} + 1 \right) \\
&= \left(\frac{d+1}{1} + 1 \right) \prod_{i=2}^{d} \left(\frac{d+1}{i} + 1 \right) \\
&\geq \left(\frac{d+1}{1} + 1 \right) 2^{d-1} && \text{since } \frac{d+1}{i} + 1 \geq 2 \text{ for all } 1 \leq i \leq d \\
&\geq \left(\frac{2+1}{1} + 1 \right) 2^{d-1} && \text{since } d \geq 2^5 \\
&= 4 \cdot 2^{d-1} \\
&= 2^{d+1} = 2^{\lfloor \sqrt{t} \log n \rfloor}. && (12)
\end{aligned}$$

We can finally show that

$$\begin{aligned}
n^{\sqrt{t}} &\geq |\mathcal{G}| && \text{by Lemma 12} \\
&\geq \binom{t+l}{t-1} && \text{by Lemma 11} \\
&> \binom{l+1+\lfloor \sqrt{t} \log n \rfloor}{\lfloor \sqrt{t} \log n \rfloor} && \text{by Inequality 9.} \\
&\geq \binom{2\lfloor \sqrt{t} \log n \rfloor + 1}{\lfloor \sqrt{t} \log n \rfloor} && \text{by Inequality 11} \\
&\geq 2^{\lfloor \sqrt{t} \log n \rfloor + 1} && \text{by Inequality 12} \\
&\geq 2^{\sqrt{t} \log n} \\
&= \left(2^{\log n} \right)^{\sqrt{t}} \\
&= n^{\sqrt{t}}.
\end{aligned}$$

But then $n^{\sqrt{t}} \geq |\mathcal{G}| > n^{\sqrt{t}}$, a contradiction. We conclude that n must be a power of p, i.e. $n = p^k$ for some $k \in \mathbb{N}$. But since the Algorithm 3 reached line 13, the if-predicate on line

4 was not fulfilled, so for all $k \geq 2$, $p^k \neq n$. Therefore, $k = 1$ and thus $n = p$ is prime. We conclude the algorithm terminates with the correct output, `PRIME`, on line 13. □

4.5 Proof of Correctness: Conclusion

Lemmas 1–7 collectively prove that whenever Algorithm 3 terminates, it produces correct output. Theorem 5 directly follows.

4.6 Asymptotic Time Complexity

Now, we will demonstrate that the time complexity of Algorithm 3 is polynomial in $\log n$. Assume that addition, multiplication, and division of two integers $m \leq n$ can be done in $O(\log n)$ time [10].

Theorem 6. *The time complexity of Algorithm 3 is polynomial in* $\log n$.

Proof. Line 1 involves checking whether n is a perfect power, which can be done in $O(\log^3 n)$ time [11].

Line 3 involves finding the smallest $r \in \mathbb{N}$ such that $o_r(n) > \log^2 n$. However, from Lemma 2, we know that $r < \log^5 n$. We can check the smallest candidates for r first, starting at $r = 1$. For each candidate for r, we must compute $n^{\log^2 n}$ in the worst case, which with binary exponentiation takes $O\big(\log n \cdot \log\left(\log^2 n\right)\big) = O(\log n \log\log n)$ time. Performing this computation $r < \log^5 n$ times thus takes $O(\log^5 n \cdot \log n \cdot \log\log n)$ time, which is in turn less than $O(\log^7 n)$ time.

Line 4 involves computing $\gcd(a, n)$ for at most r different integers. Each gcd computation takes $O(\log n)$ [10], and $r < \log^5 n$, so the time taken in this step is $O(\log^5 n \cdot \log n) = O(\log^6 n)$.

Line 6 simply involves the comparison $n \leq r$, which take $O(\log n)$ time.

Now, consider the inner body of the for-loop on line 11, which involves computing $(X + a)^n \stackrel{?}{\equiv} X^n + a \pmod{X^r - 1, n}$ for a given integer a. Using binary exponentiation, $(X + a)^n$ can be evaluated in $\log n$ multiplications of polynomials of order less than $r < \log^5 n$, which takes $O(r^2)$ time. Therefore, the entire equation can be evaluated in $O(\log^1 0n \cdot \log n) = O(\log^1 1n)$ time.

The for-loop on line 10 runs at most

$$\begin{aligned} l + 1 &= \lfloor \sqrt{\phi(r)} \log n \rfloor + 1 \\ &< \lfloor \sqrt{r} \log n \rfloor + 1 \\ &< \lfloor \log^{5/2} \log n \rfloor + 1 \\ &< \log^{7/2} + 1 \end{aligned}$$

times. As explained above, each iteration of the loop takes $O(\log^{11} n)$ time, so the entire for-loop takes $O(\log^{7/2} \cdot \log^{11} n) = O(\log^{29/2} n)$ time.

The bottleneck step is the final one, so the entire algorithm runs in $O(\log^{29/2} n)$ time [6] We conclude that the algorithm runs in polynomial time in $\log n$.

□

5 PRIMES ∈ P

Theorem 7. *PRIMES ∈ P.*

Proof. Algorithm 3 correctly decides if a given input $n \in \mathbb{N}$ (with $n \geq 2$) is prime in polynomial time (Theorems 5 and 6). □

6 Conclusion

The introduction of the AKS Primality Test was a stunning breakthrough in number theory and theoretical computer science. Remarkably, its proof requires fairly elementary number theory – it is therefore surprising that this algorithm went undiscovered until recently. In this paper, we have shown primality testing is a fundamentally easy problem – P contains the easiest problems in algorithmic complexity theory.

Even if the AKS Primality Test improved the theoretical lower bound on the algorithmic time complexity of primality testing, it does *not* follow that the AKS algorithm is faster than existing algorithms for smaller n. In fact, the AKS algorithm is almost certainly not the fastest algorithm in most situations. This is not a failing of the AKS algorithm, but rather the difference between an *absolutely* "fast" algorithm versus an *asymptotically* "fast" algorithm.

7 Appendix

In this section, we give detailed proofs of some lemmas that are used in the main proof of correctness of the AKS Primality Test (Algorithm 3) but are not required for understanding the core of the proof.

Lemma 13. *For all $n \geq 9$,* $\text{LCM}(n) \geq 2^n$.

Proof. There are two parts to the proof, which follows closely to Nair's proof [12]: first, we show that $\text{LCM}(2n+1) \geq n(2n+1)\binom{2n}{n}$; thereafter, we show that $n(2n+1)\binom{2n}{n} \geq n4^n$.

To show $\text{LCM}(n) \geq n(2n+1)\binom{2n}{n}$, we will examine an integral that we will evaluate in two different ways to obtain an identity. Consider the definite integral

$$I = \int_0^1 x^{m-1}(1-x)^{n-m}\,dx.$$

[6] For a proof that the AKS algorithm runs in $O(\log^{21/2} n)$ time, see [8].

One way of evaluating this integral is to expand the integrand into a power series:

$$\begin{aligned} x^{m-1}(1-x)^{n-m} &= x^{m-1} \cdot \sum_{k=0}^{n-m} (-1)^k \binom{n-m}{k} x^k \\ &= \sum_{k=0}^{n-m} (-1)^k \binom{n-m}{k} x^{m+k-1}. \end{aligned}$$

Substituting this power series into the integral and evaluating, we have

$$\begin{aligned} I &= \int_0^1 \left(\sum_{k=0}^{n-m} (-1)^k \binom{n-m}{k} x^{m+k-1} \right) dx \\ &= \sum_{k=0}^{n-m} \left(\int_0^1 (-1)^k \binom{n-m}{k} x^{m+k-1}\, dx \right) \\ &= \sum_{k=0}^{n-m} \left[(-1)^k \binom{n-m}{k} \frac{1}{m+k} x^{m+k} \right]_{x=0}^{1} \\ &= \sum_{k=0}^{n-m} (-1)^k \binom{n-m}{k} \frac{1}{m+k} \\ &= \sum_{r=m}^{n} \frac{\alpha_r}{r} && \text{where each } \alpha_r \in \mathbb{Z} \\ &= \frac{0}{1} + \frac{0}{2} + \cdots + \frac{\alpha_m}{m} + \cdots + \frac{\alpha_n}{n} \\ &= \frac{\alpha}{\mathrm{LCM}(n)} && \text{for some } \alpha \in \mathbb{Z}. \end{aligned} \tag{13}$$

Therefore, $I \cdot \mathrm{LCM}(n) = \alpha$ is a positive integer. Another way of evaluating the integral I

is to repeatedly integrate by parts:

$$
\begin{aligned}
I &= \int_0^1 x^{m-1}(1-x)^{n-m}\,dx \\
&= \int_0^1 u\,dv && \text{where } u = (1-x)^{n-m} \text{ and } dv = x^{m-1}\,dx \\
&= \left[(1-x)^{n-m}\frac{x^m}{m}\right]_{x=0}^1 + \frac{n-m}{m}\int_0^1 x^m(1-x)^{n-m-1}\,dx \\
&= \frac{n-m}{m}\int_0^1 x^m(1-x)^{n-m-1}\,dx \\
&\cdots \\
&= \frac{n-m}{m}\cdot\frac{n-m-1}{m+1}\cdot\cdots\cdot\frac{1}{n-1}\cdot\int_0^1 x^{n-1}(1-x)^0\,dx \\
&= \frac{n-m}{m}\cdot\frac{n-m-1}{m+1}\cdot\cdots\cdot\frac{1}{n-1}\cdot\frac{1}{n} \\
&= \frac{(m-1)!(n-m)!}{n!} \\
&= \frac{1}{m}\cdot\frac{m!\,(n-m)!}{n!} \\
&= \left[m\binom{n}{m}\right]^{-1}.
\end{aligned}
$$

We know from Equation 13 that

$$I \cdot \mathrm{LCM}(n) = \frac{\mathrm{LCM}(n)}{m\binom{n}{m}} \in \mathbb{N},$$

so $m\binom{n}{m} \mid \mathrm{LCM}(n)$. Now, let $n = 2N$ and $m = N$, so that $N\binom{2N}{N} \mid \mathrm{LCM}(2N)$. Note that $\mathrm{LCM}(2N) \mid \mathrm{LCM}(2N+1)$, so

$$N\binom{2N}{N} \mid \mathrm{LCM}(2N+1). \tag{14}$$

Similarly, we can take $n = 2N+1$ and $m = N+1$ to obtain that

$$
\begin{aligned}
(N+1)\binom{2N+1}{N+1} &= (N+1)\frac{(2N+1)!}{(N+1)!N!} \\
&= (2N+1)\frac{(2N)!}{N!N!} \\
&= (2N+1)\binom{2N}{N}.
\end{aligned} \tag{15}
$$

divides $\mathrm{LCM}(2N+1)$.

To summarize what we have found so far, we know that $N\binom{2N}{N} \mid \mathrm{LCM}(2N+1)$ and $(2N+1)\binom{2N}{N} \mid \mathrm{LCM}(2N+1)$. But $\gcd(N, 2N+1) = 1$ since if $k \mid N$ and $k \mid 2N+1$, then $k \mid (2N+1) - 2N = 1$, so $k = 1$. Since N and $2N+1$ are relatively prime, we conclude that $N(2N+1) \mid \mathrm{LCM}(2N+1)$, so

$$\mathrm{LCM}(2N+1) \geq N(2N+1)\binom{2N}{N}. \tag{16}$$

Now, we show that $N(2N+1)\binom{2N}{N} \geq N4^N$. Consider the binomial expansion of $4^N = (1+1)^{2N}$,

$$\begin{aligned}(1+1)^{2N} &= \sum_{n=0}^{2N} \binom{2N}{n} \\ &\leq (2N+1)\binom{2N}{N}.\end{aligned}$$

Therefore,

$$N(2N+1)\binom{2N}{N} \geq N4^N. \tag{17}$$

Combining equations 16 and 17, we have that

$$\mathrm{LCM}(2N+1) \geq N(2N+1)\binom{2N}{N} \geq N4^N \geq 2 \cdot 4^N = 2^{2N+1}. \tag{18}$$

for $N \geq 2$. Furthermore, when $N \geq 4$, we have that

$$\mathrm{LCM}(2N+2) \geq \mathrm{LCM}(2N+1) \geq 4 \geq N4^N \geq 4 \cdot 4^N = 2^{2N+2}. \tag{19}$$

Combining equations (18) and (19), we have that for all $N \geq 4$, $\mathrm{LCM}(2N+1) \geq 2^{2N+1}$ and $\mathrm{LCM}(2N+2) \geq 2^{2N+2}$, thus we conclude

$$\mathrm{LCM}(n) \geq 2^n \qquad \text{for all } n \geq 2 \cdot 4 + 1 = 9.$$

□

References

[1] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*. Cambridge university press, 1994.

[2] P. D. Schumer, *Introduction to number theory*. Brooks/Cole Publishing Company, 1996.

[3] D.-Z. Sun, Z.-F. Cao, and Y. Sun, "How to compute modular exponentiation with large operators based on the right-to-left binary algorithm," *Applied mathematics and computation*, vol. 176, no. 1, pp. 280–292, 2006.

[4] J. Stopple, *A primer of analytic number theory: from Pythagoras to Riemann*. Cambridge University Press, 2003.

[5] D. C. Kozen, *Theory of computation*. Springer Science & Business Media, 2006.

[6] G. L. Miller, "Riemann's hypothesis and tests for primality," *Journal of computer and system sciences*, vol. 13, no. 3, pp. 300–317, 1976.

[7] L. M. Adleman, "On distinguishing prime numbers from composite numbers," in *21st Annual Symposium on Foundations of Computer Science (sfcs 1980)*, pp. 387–406, IEEE, 1980.

[8] M. Agrawal, N. Kayal, and N. Saxena, "Primes is in p," *Annals of mathematics*, pp. 781–793, 2004.

[9] R. A. Brualdi, *Introductory combinatorics*. Prentice Hall, 5 ed., 2018.

[10] J. Von Zur Gathen and J. Gerhard, *Modern computer algebra*. Cambridge university press, 2013.

[11] D. Bernstein, "Detecting perfect powers in essentially linear time," *Mathematics of computation*, vol. 67, no. 223, pp. 1253–1283, 1998.

[12] M. Nair, "On chebyshev-type inequalities for primes," *The American Mathematical Monthly*, vol. 89, no. 2, pp. 126–129, 1982.