

The AKS Primality Test

Ramprasad Saptharishi

1 Why am I doing this?

I do not have any excuses but my last talk on Primality was very fast, it might have been very hard on some of the students who were not used to the proof techniques or algebra used inside it. Hopefully this write-up should fill in the hand-wavings/voids in the talk.

2 Introduction

Primality has been a problem studied for centuries. It appears in numerous areas, and finding efficient algorithms to test for primality is fundamental techniques in complexity theory and algorithms.

It wasn't taken much by surprise when primality was shown to be in polynomial time since we already knew quite a lot about it.

1. Primality is in *co-NP*: A language L is in *co-NP* if for $x \notin L$, there exists a small witness for this. In the primality case, if a number n was not a prime number, then you have a divisor which is a small witness for n being composite.
2. Primality is in *NP*: Shown by Pratt, slightly non-trivial. Very clever idea that uses recursion.
3. **Very** efficient randomized algorithms.
4. Followed from the extended reimann hypothesis.

and a lot more.

For a long time people had been trying to solve the problem and Agrawal, Kayal and Saxena answered in the affirmative.

3 The Problem Statement

Primality: Input is a number n in *binary* (hence length of input is only $O(\log n)$). Decide whether n is a prime number or not.

Remember that if we are talking about polynomial time algorithms, it's polynomial in the length of the input. Hence our algorithm should run in time $O(\log^k n)$ and hence our resources have such bounds.

In general, all primality tests use a distinguisher test. There is some identity that works for primes but not for composites, use that identity to test for primality.

For example, we know by Fermat's little theorem that $a^{p-1} \equiv 1 \pmod p$ for p a prime. There are some moronic Carmichael numbers that satisfy this identity as well hence this immediately doesn't give a distinguisher.

The distinguisher used in this algorithm is the following:

Theorem 1. *A number n is prime if and only if*

$$(X - a)^n = X^n - a \pmod n$$

Hence all we need to do is check this identity for $a = 1$. Unfortunately, Godel prize winning papers are not that simple. There are some difficulties.

1. $(X - 1)^n$ takes (naively) $n - 1$ multiplications.

This however can be tackled in a clever way. For each $0 \leq k \leq \log n$, compute $(X - 1)^{2^k}$. Now look at the binary representation of n . If $n = b_0 + b_1 2 + b_2 2^2 + \dots + b_l 2^l$, then

$$(X - 1)^n = (X - 1)^{b_0} ((X - 1)^2)^{b_1} ((X - 1)^{2^2})^{b_2} \dots ((X - 1)^{2^l})^{b_l}$$

which is just a product of l terms among those you had computed earlier. This $(X - 1)^n$ can be computed efficiently.

2. There could potentially be n terms in $(X - 1)^n$, checking if it was equal to $X^n - a$ is clearly infeasible in polylog time.

Tackling this difficulty is the heart of the algorithm. The solution is to go modulo a small polynomial say $X^r - 1$. This essentially means you take $X^r = 1$ in the polynomial, thus forcing its degree to be bounded by $r - 1$.

For example:

$$(x^{12} + 23x^{10} + x^3 + x^2 + 5) = 1 + 23x + 1 + x^2 + 5 \pmod{x^3 - 1}$$

Hence instead of checking if $(X - 1)^n = X^n - a \pmod n$, we will check $(X - a)^n = X^n - a \pmod{X^r - 1, n}$ for some small r and for different a 's.

Thus our algorithm sketch is the following:

- 1: {some preliminary tests and choice for r , s and a }
- 2: **for** $a = 1, 2, \dots, s$ **do**
- 3: **if** $(X - a)^n \neq X^n - a \pmod{X^r - 1, p}$ **then**
- 4: **declare** COMPOSITE

5: **end if**
6: **end for**
7: **declare** PRIME

We still have to figure out the preliminary tests and the choice for r and s .

4 Preventing composite n from passing the test

Suppose n was composite and p was a prime divisor of n , we need to impose sufficient conditions on n to fail the test at some step. Here we have this nice idea of 'introspective numbers' as they call it.

Call a number m introspective (definition depends on the parameters) if for all $a = 1, 2, \dots, s$

$$(X - a)^m = X^m - a \pmod{X^r - 1, p}$$

Suppose some stupid composite n passed the test, then certainly n was introspective since if $(X - a)^n = X^n - a \pmod{X^r - 1, n}$ then certainly also \pmod{p} . And further, from Fermat's little theorem, we have $(X - a)^p = X^p - a$ for free.

Hence:

$$\begin{aligned} (X - a)^n &= X^n - a \pmod{X^r - 1, p} \\ (X - a)^p &= X^p - a \pmod{X^r - 1, p} \end{aligned}$$

And we have the nice claim, saying that introspective numbers are multiplicative.

Claim 2. *If m_1 and m_2 are introspective, that is for all $1 \leq a \leq s$*

$$\begin{aligned} (X - a)^{m_1} &= X^{m_1} - a \pmod{X^r - 1, p} \\ (X - a)^{m_2} &= X^{m_2} - a \pmod{X^r - 1, p} \end{aligned}$$

then so is $m_1 m_2$

$$(X - a)^{m_1 m_2} = X^{m_1 m_2} - a \pmod{X^r - 1, p}$$

Proof. I assume the proof was clear in the talk, you may revisit the slides if not. \square

From this we have that if n and p are introspective, then so is every number of the form $p^i n^j$.

Since $X^r = 1$, it makes sense to just look at the residues of $p^i n^j \pmod{r}$. Thus consider \mathbb{Z}_r^* , the multiplicative group modulo r and let its size be t .

Look at the set

$$L = \{p^i n^j : 0 \leq i, j \leq \sqrt{t}\}$$

The number of elements in the set is $(\sqrt{t} + 1)^2 > t$ and hence by pigeon hole principle, there exist two number $m_1 = p^{i_1} n^{j_1}$ and $m_2 = p^{i_2} n^{j_2}$ with $m_1 =$

$m_2 + kr$. This would hence force that $(X - a)^{m_1} = (X - a)^{m_2} \pmod{X^r - 1, p}$ as $X^r = 1 = X^{kr}$.

Suppose from $(X - a)^{m_1} = (X - a)^{m_2}$ we were to force that $m_1 = m_2$, then we have joy! If $m_1 = m_2$, n must be p^k for some k , and this can be checked easily. (for each $2 \leq k \leq \log n$, do a binary search on $a = 2, \dots, n$ to check if any $a^b = n$).

We now just need to ensure that $m_1 = m_2$ is forced.

5 Required to force $m_1 = m_2$

We have $(X - a)^{m_1} = (X - a)^{m_2} \pmod{X^r - a, p}$ for all a in the range. This is just saying the polynomial $h(z) = z^{m_1} - z^{m_2}$ has $(X - a)$ for different a as roots of it.

Note that the degree of h is at most $\max(m_1, m_2)$ and since $m_1, m_2 \in L$, $p^i n^j \leq n^{2i} \leq n^{2\sqrt{t}}$ since \sqrt{t} was the bound on i and j . We will now go on to show that h has more than $n^{2\sqrt{t}}$ many roots.

Look at the primitive r -th root of unity, η . Since X is essentially η since we are going mod $X^r - 1$, it just means that $(\eta - a)$ are roots of h . Now note that if $(\eta - a)$ and $(\eta - b)$ are roots of $h(z) = z^{m_1} - z^{m_2}$, then so is $(\eta - a)(\eta - b)$. Hence, every element of

$$S = \left\{ \prod_{a=1}^s (\eta - a)^{\delta_a} : \delta \in \{0, 1\} \right\}$$

Things would be fantastic if we show that every element of S is distinct, then we have 2^s roots of h . Then all we need to ensure is that $2^s > n^{2\sqrt{t}}$.

Recall that $t = |\mathbb{Z}_r^*| \leq r - 1$. With some playing around with the inequalities, it's easy to see that $s = 2\sqrt{r} \log n + 1$ works for the required inequality $2^s > n^{2\sqrt{t}}$. Since the polynomial h has more roots than its degree, it has to be zero, hence $m_1 = m_2$.

Now we need to show that the elements of S are distinct. Let us instead look at a different set,

$$S = \left\{ \prod_{a=1}^s (X - a)^{\delta_a} : \delta \in \{0, 1\} \right\}$$

Are these elements distinct? Just look at the $(X - a)$'s, when will $(X - a) = (X - b)$? This can happen only when $a = b \pmod{p}$ which means $(a - b) = 0 \pmod{p}$ or p divides $(a - b)$. Since we are only looking at $(X - a)$ where $1 \leq a \leq s$ with some small s , if p divides $a - b$, then p certainly has to be less than s . Hence if we ensure that n has no small prime divisors like p , we can be certain that each $(X - a)$ is a distinct element. Hence, just add to the preliminary tests to check for small divisors (for each $2 \leq a \leq s$, check if a divides n).

Now any polynomial in S_X factorizes as linear factors of the form $(X - a)$, and since each of these factors are distinct, and the factorization is unique, the polynomials as such are distinct. Hence S_X has 2^l elements. It's now left to show that for two different polynomials g_1 and g_2 in S_X , $g_1(\eta) \neq g_2(\eta)$.

Suppose for two different g_1 and g_2 , if $g_1(\eta) = g_2(\eta)$. Recall our earlier claim on introspective numbers, $g(X)^m = g(X^m)$ for all m of the form $p^i n^j$. Hence if η is a root of $g_1 - g_2$, then so is $\eta^{p^i n^j}$.

η^r is anyway 1, hence we need to look at the different roots that $\eta^{p^i n^j}$ generate for different i and j . Since η was chosen to be a primitive root, $\eta^{p^i n^j} = \eta^{p^i n^j \bmod r}$ and there are as many values of the exponent as there are residues of $p^i n^j$ modulo r . Hence equivalently we need to ask, how many residues modulo r do $p^i n^j$ generate? OK, we don't know anything about this prime number p , but if the order of n modulo r was k , then $p^i n^j$ has atleast k different residues.

$g_1 - g_2$ is a polynomial of degree atmost s , and if we choose $\text{ord}_r(n) > s$, then we have more roots than the degree, thus $g_1 = g_2$. Hence we just need to choose an r such that the degree of n modulo r is greater than s . Recall that $s = 2\sqrt{t} \log n + 1 \leq 2\sqrt{r} \log n + 1$ (since the size of the multiplicative group modulo r , which is equal to t , has to be less than r).

Since $t \leq \text{ord}_r(n)$, and we want $\text{ord}_r(n) \geq 2\sqrt{t} \log n + 1$, the inequalities can be solved easily. An r such that $\text{ord}_r(n) \geq 4 \log^2 n + 2$ is a safe choice.

Then there is some technical detail of saying that a such an r exists within a small range and can be found easily. I'm omitting that part, you could see the presentation for the details on that.

Finally, all the pieces put together.

- 1: {as needed after forcing $m_1 = m_2$ }
- 2: **if** $n = a^b$ for $a, b \geq 2$ **then**
- 3: **declare** COMPOSITE
- 4: **end if**
- 5: Choose r such that $\text{ord}_r(n) \geq 4 \log^2 n + 2$
- 6: Let $s = 2\sqrt{r} \log n + 1$
- 7: {to make sure there are no small divisors, preventing wrap arounds}
- 8: **if** any $2 \leq a \leq s$ divides n **then**
- 9: **declare** COMPOSITE
- 10: **end if**
- 11: **for** $a = 1, 2, \dots, s$ **do**
- 12: {the distinguisher test}
- 13: **if** $(X - a)^n \neq X^n - a \pmod{X^r - 1, p}$ **then**
- 14: **declare** COMPOSITE
- 15: **end if**
- 16: **end for**
- 17: **declare** PRIME

6 Conclusion

The other open problems part can be seen in the presentation. My apologies for going too fast in the talk, I realised only half way through that I had not been clear, it was already late by then. I tried to make up at the end of it, but wasn't convinced that I did a good job. I hope this write-up makes things a clearer, I'd be delighted to receive any comments/suggestions or to answer any doubts in this.

Thank you.