A Note On the Storage Requirement for AKS Primality Testing Algorithm

Zhengjun Cao

Abstract

We remark that AKS primality testing algorithm needs about 1,000,000,000 G (gigabyte) storage space for a number of 1024 bits. Such storage requirement is hard to meet in practice. To the best of our knowledge, it is impossible for current operating systems to write and read data in so huge storage space. Thus, the running time for AKS algorithm should not be simply estimated as usual in terms of the amount of arithmetic operations.

1 Introduction

The Agrawal-Kayal-Saxena primality test (AKS algorithm for short) is a deterministic primality-proving algorithm created by Manindra Agrawal, Neeraj Kayal, and Nitin Saxena [1]. The algorithm determines whether a number is prime or composite within polynomial time. The AKS algorithm is the first primality-proving algorithm to be simultaneously general, polynomial, deterministic, and unconditional. Many fast primality tests work only for numbers with certain properties. For example, the Lucas-Lehmer test works only for Mersenne numbers, The elliptic curve primality test and Adleman-Pomerance-Rumely primality test [2] prove or disprove that a given number is prime, but are not known to have polynomial time bounds for all inputs. Randomized tests, such as Miller-Rabin [4, 5], can test any given number for primality in polynomial time, but are known to produce only a probabilistic result. The correctness of AKS is not conditional on any subsidiary unproven hypothesis. In contrast, the Miller-Rabin test is fully deterministic and runs in polynomial time over all inputs, but its correctness depends on the truth of the yet-unproven generalized Riemann hypothesis.

[0]Department of Mathematics, Shanghai University, Shanghai, China. caozhj@shu.edu.cn

The AKS primality test has to find a suitable value r which will be $O(\log^5 n)$. It then checks that $(X+a)^n = X^n + a \,(\mathrm{mod}\, X^r - 1, n)$ for $a = 1$ to $\lfloor \sqrt{\phi(r)} \log n \rfloor$. Since X is a free variable which is never substituted by a number, it has to reduce $(X+a)^n$ in the ring $\mathbb{Z}_n[X]/(X^r - 1)$. In this note, we point out that the storage requirement for the AKS algorithm is hard to meet even in 2013.

2 Review of AKS algorithm

The AKS primality test is based upon the following theorem: an integer $n(\geq 2)$ is prime if and only if the polynomial congruence relation

$$(X+a)^n = X^n + a \,(\mathrm{mod}\, n)$$

holds for all integers a coprime to n. In 2002, the authors [1] proved that for appropriately chosen r if the following equation

$$(X+a)^n = X^n + a \,(\mathrm{mod}\, X^r - 1,\ n) \tag{1}$$

is satisfied for several a's then n must be a prime power. The number of a's and the appropriate r are both bounded by a polynomial in $\log n$.

AKS algorithm
Input: integer $n > 1$.
1. If ($n = a^b$ for $a \in \mathbb{N}$ and $b > 1$), Out COMPOSITE.
2. Find the smallest integer r such that $\mathrm{ord}_r(n) > \log^2 n$.
3. If $1 < \gcd(a, n) < n$ for some $a \leq r$, output COMPOSITE.
4. If $n \leq r$, output PRIME.
5. For $a = 1$ to $\lfloor \sqrt{\phi(r)} \log n \rfloor$ do if $((X+a)^n \neq X^n + a \,(\mathrm{mod}\, X^r - 1, n))$, output COMPOSITE.
6. Output PRIME.

Notice that r is essential to AKS algorithm. It has been proven that r will be $O(\log^5 n)$.

3 On the storage requirement for AKS algorithm

We note that Eq.(1) is a polynomial congruence relation where X is a free variable. It is never substituted by a number, instead it has to reduce $(X+a)^n$ in the ring $\mathbb{Z}_n[X]/(X^r-1)$ and compare the coefficients of the X powers.

Now suppose that n is of 1024 bits and $r = 1024^5 k$ where k is a positive number. In the reduction process of $(X+a)^n$ in the ring $\mathbb{Z}_n[X]/(X^r-1)$, the AKS algorithm has to compute

$$(c_{r-1}X^{r-1} + \cdots + c_1 X + c_0)^2 \pmod{X^r - 1, n}, \quad c_i \in \mathbb{Z}_n,\ i = 1, \cdots, r-1. \tag{2}$$

if it uses the repeated-squaring method, and store a polynomial of degree $2r-2$,

$$d_{2r-2}X^{2r-2} + \cdots + d_1 X + d_0, \quad d_i \in \mathbb{Z}_n,\ i = 1, \cdots, 2r-2. \tag{3}$$

Each of these coefficients takes 1024 bits storage space. Thus the algorithm takes almost $2 \times 1024^6 k$ bits storage space, i.e., about 1024^2 TG (terabyte).

As of 2013, the storage space for PC is less than 1 TG. That is to say, the AKS algorithm needs about 1,000,000 PC's storage space. To the best of our knowledge, it is impossible for current operating systems to write and read data in so huge storage space.

It has been shown that the best possible estimation for r is $O(\log^2 n)$ if Aritn's Conjecture holds. In such case, the storage space required is about 1 GB (gigabyte). In contrast, the Miller-Rabin test only requires several kilobyte (KB) since it is simply computing $\alpha^{2^s t} \bmod n$ for some randomly picked $\alpha \in \mathbb{Z}_n^+$, where $2^s t \,|\, n-1, \gcd(2,t) = 1$.

4 On some implementations of the AKS Algorithm

Some packages for the AKS algorithm can be found at

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

Some of them do not check that $\mathrm{ord}_r(n) > \log^2 n$ which is necessary for the correctness of AKS algorithm.

There is a latest implementation report for AKS algorithm [3]. For $n = 100000000000031$, the chosen number for r is 1024 (see Ref.[3], page 55). Note that the bit-length of 100000000000031 is 47, $\mathrm{ord}_r(n) = 32 < 47^2$. That is, the chosen number for r is not suitable. Thus, the estimated running time is not convincing.

5 Conclusion

The AKS algorithm may be the first algorithm in literatures that requires about 1,000,000,000 G storage space. In our opinion, the running time for AKS algorithm should not be simply estimated as usual in terms of the amount of arithmetic operations required.

References

[1] Agrawal, M.; Kayal, N.; Saxena, N.: PRIMES is in P. Annals of Mathematics 160 (2): 781-793. 2004.

[2] Adleman, M.; Pomerance, C.; Rumely, S.: On distinguishing prime numbers from composite numbers. Annals of Mathematics 117 (1): 173-206. 1983.

[3] Li H.: The Analysis and Implementation of the AKS Algorithm and Its Improvement Algorithms. Available at: `opus.bath.ac.uk/16757/1/CSBU-2007-09.pdf`

[4] Miller, G.: Riemann's hypothesis and tests for primality. J. Comput. Sys. Sci., 13:300-317, 1976.

[5] Rabin, M.: Probabilistic algorithm for testing primality. J. Number Theory, 12:128-138, 1980.