



© 1997–2004, Millennium Mathematics Project, University of Cambridge.

Permission is granted to print and copy this page on paper for non-commercial use. For other uses, including electronic redistribution, please contact us.

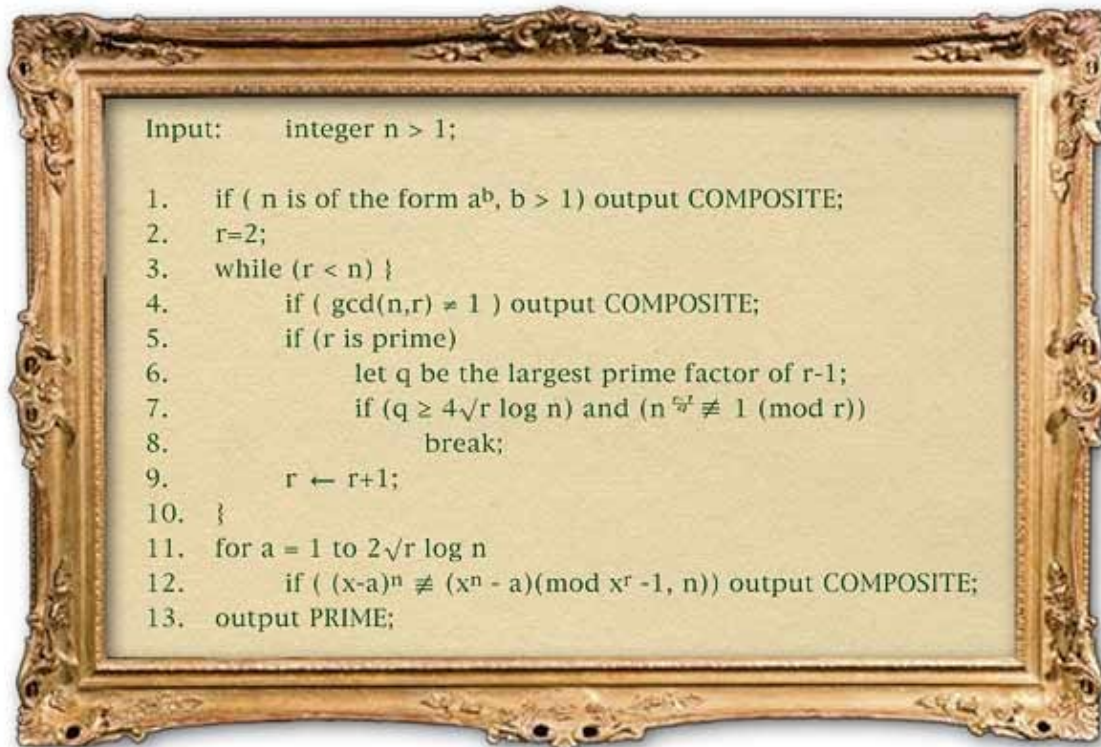
09/10/2002

News

Prime time



Despite struggling for centuries to find a simple and efficient way to test whether a number is prime, mathematicians have been gazumped by two computer science students and their professor. Manindra Agrawal of the Indian Institute of Technology, Kanpur, and his two postgraduate students, Neeraj Kayal and Nitin Saxena, have found a quick and simple algorithm (known as the AKS algorithm) that conclusively tests for primality.



The AKS Algorithm

Prime numbers are central to mathematics. They are the building blocks of the natural numbers – any integer can be expressed uniquely as a product of primes – and they play a crucial role in cryptography and

Prime time

e-commerce. The earliest known primality test is Eratosthenes' Sieve, which dates from around 240 BC. The Greek mathematician simply checked whether each integer less than the square root of n (the number to be tested) divided n . This method is straightforward but slow, as the time required increases exponentially as n increases.

With the rise of e-commerce and the resulting importance of secure communications, identifying primes has become vital to all of us (see Safety in Numbers in Issue 21 of *Plus* for how primes are used in cryptography). Mathematicians already have algorithms for primality testing that are fast and correct enough of the time for all practical purposes – but up till now the choice has been between algorithms that are fast but have a small chance of giving the wrong answer, or that are correct but have a small chance of being slow. In fact there are algorithms that run faster than AKS, but the lack of a primality algorithm *proven to run in polynomial time* has been a thorn in mathematicians' sides.

The *complexity* of an algorithm is the maximum time it takes to return an answer, expressed in terms of the number of steps required for the algorithm to run, as a function of the size of the input. A *polynomial time algorithm*, such as the AKS algorithm, is one where the complexity is some power of the input. In this case, the input (the number of bits needed to represent the number n being tested) is $\log n$ (base 2). The complexity of the AKS algorithm is $\log n^{12}$.

The result is a "superb theoretical breakthrough" according to mathematicians in the field. And not only does it finally prove that primality testing is possible in polynomial time, but it does so using relatively simple mathematics. In contrast to other work in the field, it takes just 9 pages to prove both the correctness and the speed of the AKS algorithm.

The elegance and simplicity of the result comes from an ingenious application of Fermat's Little Theorem:

$$\text{For any prime integer } n \text{ and any integer } a \text{ coprime to } n, a^{n-1} = 1 \pmod{n}.$$

What was new in this work was the way the researchers reduced the number of instances of this theorem that they needed to check (the loop in steps 3 to 10). Agrawal calls this process "*lifting the polynomial*" – extending correctness of the statement over a smaller number of cases to a proof which is valid for the whole domain.

Impressive though this theoretical achievement is, it will have little practical impact, as the existing algorithms are faster and sufficiently accurate. And, of course, the harder problem of *factorising* a number (in polynomial time) once it is known to be composite remains unsolved. However, as Kayal and Saxena have achieved a result significant enough to qualify for a Ph.D. after just one semester, perhaps they or another group of students will surprise the mathematics world with the solution to this problem as well.

Rachel Thomas



Plus is part of the family of activities in the Millennium Mathematics Project, which also includes the NRICH and MOTIVATE sites.