

Re: JSH: Easy math, easy solution

Source: <http://sci.tech-archive.net/Archive/sci.math/2005-02/2931.html>

From: David McAnally (*D.McAnally_at_i'm_a_gnu.uq.net.au*)

Date: 02/09/05

Date: Wed, 9 Feb 2005 12:37:19 +0000 (UTC)

"Steven" <somewherenonexistant@yahoo.com> writes:

>Ok, so what if you came up with a new factorization algorithm? At
>least come up with one that is fast! When you do that, go to RSA's
>website and do their prime factorization challenge. After you won a
>few thousand dollars THEN you have the right to post what you did.
>(Besides, it is also easy to check for divisors!)

Factorization is quite simple. Here is an algorithm for it.

Input: a composite number n .

Output: an ordered pair (b,c) such that $1 < b < n$, $1 < c < n$, and $bc = n$.

1. If n is even, return $(2, n/2)$.
2. Let $k = 2$.
3. If n is not a perfect k -th power, go to Step 5.
4. Set x equal to the k -th root of n , and return (x, x^{k-1}) .
5. Increment k by 1 (i.e. $k := k+1$).
6. If $k < \text{floor}(\log n / \log 3) + 1$, go to Step 3.
7. Select x such that $1 < x < n-1$ at random.
8. If $\text{gcd}(x,n) > 1$, return $(\text{gcd}(x,n), n/\text{gcd}(x,n))$.
9. Determine the order, a , of x modulo n .
10. If a is odd, go to Step 7.
11. If $x^{a/2}$ is congruent to -1 modulo n , go to Step 7.
12. Return $(\text{gcd}(x^{a/2}-1, n), \text{gcd}(x^{a/2}+1, n))$.

sci.math: Re: JSH: Easy math, easy solution

All of these steps can be performed in polynomial time on a classical computer, except for Step 9. There is no known polynomial time algorithm for Step 9 on a classical computer. On the other hand, if quantum computers become a reality, then Step 9 can be performed in polynomial time using both a quantum computer and a classical computer.

So, after the invention of quantum computers, whenever that would be, the above gives a polynomial time algorithm for factorization.

David
