

Re: Proof factoring solution is closed form

Source: <http://sci.tech-archive.net/Archive/sci.math/2005-02/3092.html>

From: Mark Nudelman (markn_at_greenwoodsoftware.com)

Date: 02/09/05

Date: Wed, 9 Feb 2005 09:46:27 -0800

Larry Lard wrote:

> jstevh@msn.com wrote:

>> Rick Decker wrote:

>>> jstevh@msn.com wrote:

> [snip]

>>>> And that's a polynomial time problem, as the equation defining how

>>>> many combinations of those factors there are is a polynomial one.

>>>>

>>> That's not what we mean by a poly-time solution. In fact, one might

>>> have to search for factors in a space that could be about as large

>>> as M itself. If you're going to get a poly-time solution, you'll at

>>> least have to cut down the search to $O((\log M)^k)$, for some fixed k .

>>>

>>

>> Why?

>>

>> You aren't giving any information here, but just assertions without

>> explanation.

>

> No, he's giving you **definitions**. He's pointing out that the meaning

> you've ascribed to 'polynomial time' differs from the definition used

> by **EVERYONE ELSE IN THE WORLD**. You have in the past claimed to be a

> professional programmer; the fact that you appear not to know what

> 'polynomial time' means is... well, I was going to say surprising, but

> really, demonstrations of your ignorance no longer surprise.

James, if you don't know what "polynomial time algorithm" means in this context, you shouldn't claim your algorithm is polynomial time, or you just end up sounding ignorant. To be a polynomial time factoring algorithm, the number of operations required must be a polynomial IN THE NUMBER OF DIGITS OF M , not in M itself. Or, as Rick stated, the number of operations must be $O((\log M)^k)$. (If you don't know what the Big-Oh notation means, you should read about that too.) This is not just an arbitrary definition -- a poly-time algorithm MIGHT be able to factor a large integer in a reasonable amount of time (depending on the degree of the polynomial), but a non-poly-time algorithm (e.g. exponential time) simply will not.

--Mark