Re: JSH: Surrogate factoring, periodic behavior

Source: http://sci.tech−archive.net/Archive/sci.math/2007−09/msg00206.html

- *From*: JSH <jstevh@xxxxxxxxx>
- *Date*: Sun, 02 Sep 2007 00:03:28 −0000

On Sep 1, 3:34 am, rossum <rossu...@xxxxxxxxxxxx> wrote:

> On Sat, 01 Sep 2007 05:21:41 −0000, JSH <jst...@xxxxxxxxx> wrote:
>
> > Factored trying 5 surrogates on the fifth surrogate using k=1 where
> > the start was with n=7.
>
> I ran my usual tests again on 500 random composite odd numbers that
> are multiples of two different primes, each in the range 500 to 1000.
> The results are compared to Fermat's method, trial factorisation (both
> forward and reverse) and random picking.
>
> Fermat average = 7.58 probes.
> JSH average = 1635.83 probes.
> Probe ratio = 1 : 215.752
> Trial average = 118.52 probes.
> Reverse average = 12.12 probes.
> Random average = 727.79 probes.
>
> 500 trials, 0 misfactors found.

Here is output from my program and a run I just did testing out some
new theory.

Here for various reasons I won't explain I'm using k=30, starting with
n=1, and there is a lot wrong with this example from my standpoint as
by my theory I don't think it should have factored at this point.

java Factor 34587541
n_max=0
Number factored.
Total all combinations: 21
Time: 0
Time/combination: 0.0

Surrogate:

(11^2)(13)(19)(8101)
Product: 242114587

Surrogate combinations checked:
Initial Factorization:

f_1=307
f_2=112663
Now checking its factors...
Success!
Factors:
(307)(112663)
Product: 34587541

In coming is 34587541
Factored for k : 3
Not factored for k: 4
Factored fuel percentage: 42%

Factored fuel : 3
Fuel not factored: 4
Factored fuel percentage: 42%

Processing time: 78
Number of digits: 8
bitLength=26

The program I use loops through trying to factor the surrogate and if it can't fully factor it just keeps on going, where in this case it says it factored 3 fully and could not fully factor 4, so I guess it checked 7, but now I'm not sure what.

As it looks like to me it only tried one surrogate, so I'm not sure why it says it checked seven unless, um, the code is doing something I'm not sure about which wouldn't be a surprise.

I'm kind of half−heartedly hacking at it because I think it stupid that the world should suffer because you people lie.

How is that fair?

In any event, it's an example of what I mean of the kind of behavior I see all the time with dinky numbers.

Totally not random.

> Average k's tried per factorisation: 1.000
> Average n's tried per factorisation: 47.040
>
> k was fixed at 1 and n was 7, 8, 9, ...

> Part of the problem I've run into with comments about surrogate factoring have been repeated claims it works only as good as trial division, but hey, I programmed the damn thing.

Any chance of seeing your code?

No.

But you know I'm rather fascinated by this entire process as it reminds me of when I was working on prime counting, and eventually posters just went to lying about my research.

But this time if I figure out what's going on I can just factor big numbers, and see if you try to lie your way out of that, which makes the situation different.

Your only hope is that I'm bluffing.

Ok, I'm bluffing. Sleep ok tonite.

That's my present to you.

James Harris

.