

Yet Another Factoring Algorithm (yafa)

Source: <http://sci.tech-archive.net/Archive/sci.math/2006-05/msg01074.html>

- *From:* Marc Bogaerts <mbg.DELSPAMnimda@xxxxxxxxxx>
 - *Date:* Sat, 06 May 2006 06:29:25 +0200
-

I would like to have comments on the following algorithm.

It is written in gap, so it must be easily translated in other mathematica/maple lingo. It is based on an earlier thread I had a few weeks ago about the properties of an element in the residu mod n ring Z_n if its order has certain properties.

The algorithm starts with an arbitrary seed value ($\neq 1$) and takes successively value of the p_i th power of its previous value where p_i ranges through the set of prime numbers $\{2,3,5, \dots\}$ (until a certain give_up value is reached). The reason for this is to reduce the order of the seed (if it is in the multiplicative group Z_n^*). In other words, a certain seed value s is taken in Z_n , in the worst case it is a generator of Z_n^* . Then its successive values, $s_1=s^{p_1}$, $s_2=s_1^{p_2}$, ... (where p_i in $\{2,3,5,7,\dots\}$) are taken and tested if $\text{Gcd}(s_i-1, n) > 1$. At each step of the algorithm the order of s_i in Z_n^* gets knocked down by a factor of $\phi(n)$, until only one factor remains(*).

Here is the code (in GAP):

```
ContExp := function (seed, md, give_up) local exp;
exp:=2;
for ii in [1..give_up+1110] do
seed:=PowerMod (seed, exp, md);
if (ii = 10) then exp:=2; fi;
if (ii = 100) then exp:=2; fi;
if (ii = 1000) then exp:=2; fi;
if (ii mod 1000)=0 then Print(exp, "\n"); fi;
if (seed=1) then Print("result=1\n"); return(exp); fi;
if (Gcd(seed-1,md) > 1) then return (seed-1); fi;
exp:=NextPrimeInt(exp);
od;
Print("givin up\n");
return(seed);
end;
```

Example:

Yet Another Factoring Algorithm (yafa)

```
gap>p:=562458284892618761003429;  
562458284892618761003429
```

```
gap>q:=159577813696658854663811;  
159577813696658854663811
```

```
gap> n:=p*q;  
89755863398736586273160180973179889570813207919
```

Now let's apply the algorithm to this number n in order to find one of it's factors:

```
gap> Q:=ContExp(5, n, 10000000);  
2  
7927  
17393  
27457  
37831  
48619  
59369  
70663  
81817  
93187  
104743  
68920180065214556985830102631535182322010856271
```

```
gap> Q;  
68920180065214556985830102631535182322010856271
```

```
gap> Gcd(n,Q);  
159577813696658854663811
```

Another example, not far away from the chosen (p,q) pair:

```
gap> n:=p*q;  
89755863398736586274405356167570178617049343039  
gap> ContExp(5,n,10000000);  
2  
7927  
17393  
27457  
37831  
48619  
59369  
70663  
81817  
93187  
104743  
79195854690403543333663611308152993751906530365  
gap> Gcd(n,Q);  
159577813696658854665349
```

Yet Another Factoring Algorithm (yafa)

```
gap> n/last;  
562458284892618761005811
```

(*) this, actually puts a limit on the feasibilities of the algorithm.

Have a look at the factors of $\phi(n)$ and it's easy to find out.

.