

Research Statement: Factorization of Arbitrary Precision Integers in Real Time

Subhendra Basu

December 11, 2022

1 Abstract

I draw from no external reference to present a novel algorithm to factorize an integer into two factors, not necessarily prime. This short article has no Erdos Number as this cannot be traced to Erdos' original paper.

2 Algorithm Intfact

2.1 Introduction

Lower Factor is obtained by masking digits of π^x after the decimal point taking three digits at a time with the digits of e, the base of the natural logarithm, after the decimal point taking five digits at a time with an overlap of two digits between consecutive blocks of digits. Here, $x = (\text{middle digit}, 0 \text{ if null})(\text{decimal point}, .)$ (first half of number) strcat (second half of number) For $N = 30623$, $x = 6.3023$; For $N = 2189$, $x = 0.2189$. Higher Factor is obtained by masking digits of e^x in a similar fashion as above with the digits of π , the irrational number. x has an identical connotation as done in the case of the lower factor.

The testcases is $N = 30623$, $n = 5$ digits. We consider Factorization as complete when we have encountered five 8's in the residue set and the only digit in the residue set after masking should be 8. We take five 8's because $n=5$.

For the sake of this experiment, I used the following UNIX commands in (bc

```
-l). e^x scale=1000;e(6.3023);  
  pi^x scale=1000;e(6.3023*l(4*a(1))) Remarks pi=4*arctan(1)
```

2.2 Lower Factor, Thread 1

($e^{6.3023}$ 545.)	825	825 repeat extent = 2	867 extent = 1	016	016
(π 3.)	14159	59265	65358	58979	79323
(Residue Set)	2	NULL Fast Count = 1	NULL	1	1
(Remarks)	2 is not present in 14159	2 is present in 59265			

Further calculation is described in the link below. When we encounter the first

digit 8 in the residue set, we have 15 runs of intervals greater than 1. An interval is a run whence the residue set is non null for multiple stages greater than 1. The decimal number 15 is described as 1111 in binary number system. Since we have already encountered an 8, we have 4 8's remaining for the factorization to be completed. We skip the first interval following the first 8, and count the rest, we get 13 intervals of extent greater than 2 and 4 8's; which is $13 + (4-1) = 13 + 3 = 16$ or 10000 in binary number system. Higher Factor in Binary = Stage 2 strcat Stage1 = 10000 strcat 1111 or 100001111 or 271 in decimal.

2.3 Higher Factor, Thread 2

$(\pi^{6.3023} \ 1358.)$	900	900	900	900	450
$(e \ 2.)$	71828	28182	82845	45904	04523
(Residue Set)	0	0	0	NULL	NULL

Further Calculation is described in the link below. We encounter 4 8's and 4 intervals of extent greater than 1 in the first instance. This is decimal $4 + (4 - 1) = 7$ or 111 in reverse in binary. Balance is 1 8. In the second instance we get 4 intervals of extent greater than 1 and a remaining 8 which is substituted for a zero, 0. making the yield from the second instance as (0)(001) String Concatenating first stage with the second stage makes it 111 strcat 0001 or 1110001 in binary or 113 in decimal Hence, $30623 = 113 \times 271$.

3 Conclusion

Factorization can be carried out in a multi-threaded fashion in a programming language that supports multiple threads of execution. For the detailed computation, refer to the following hyperlink: Detailed Computation