Toom-Cook

Sai Krishna Yedugani.
#991727920

Toom-Cook

Contents

1 Abstract

In the present scientific computation like cryptography and many applications multiplication of two polynomials is very big concern. There are only few algorithms that advance efficiency through this large integer multiplication. Toom cook-3, Toom-cook-4 way and Karatsuba area the well known techniques that help in improving the efficiency in cryptosystems. In this paper we work on the classical Toom-k way of dealing with the multiplication of large integers.

2 Introduction

Dealing with complex multiplications with our mind is completely out of bound. We use calculators or some computer applications to do calculations. With these applications we can work on upto 8 digits and the result we get is about 15 digits, and is accurate. Whereas, for the digits more than this rounding off is done. There comes the issue when one needs to work on large integers i.e more than 8 digits and getting the perfect result. For these complex computations scientist have devised some special routines, and this led to the invention of some algorithms. Few of these popular algorithms are Toom-Cook and Karatsuba Algorithm and Schonhage-Strassen algorithm(SSA).

The Toom-Cook algorithm follows the divide and conquer method for multiplying large integers. Just like Karatsuba it splits the given integer into n limbs of some fixed size. The division is then applied recursively with Toom-3 algorithm. This goes on until we are able to apply another algorithm on it for the last stage of recursion, or until the desired multiplier is reached.

The method works on the principles of polynomial multiplication. The input numbers are divided into limbs of a given size, and each in the form of polynomial, and the limb size is used as radix. Instead of multiplying the obtained polynomials directly, they are evaluated at a set of points, and the values multiplied together at those points. Based on the products obtained at those points the product polynomial is obtained. The final result is then obtained by substituting the radix.

2.1 Where did the method come from ?

The First algorithm that was developed for the univariate polynomials is Karatsuba. It was a stepping stone for other algorithms for multiplication, including the Toom-Cook algorithm. The Toom-Cook method is actually

based on the Karatsuba method by splitting each number to be multiplied into multiple parts. Toom -3 way reduces number of multiplications greatly depending on the number of splits done.

2.2 Problem Statement

The problem we are working on is to perform arithmetic operations on large integers, and especially multiplications on them. The reason for dealing with them is caused from cryptography. For ex, when we want to encrypt a string we first convert them into series of long integers, and the encryption keys are stored in the form of long integers. In order to make the encryption and decryption techniques efficient it depends in the arithmetic on long integers containing hundreds of digits. Additions and subtractions on these integers is easy. Typically it takes O(n) time for the algorithms to run these. Considering n is the number of digits. Whereas, it takes O(n2) running time for the algorithms to run on multiplications of these number. This running time for multiplication gets costly when it has to deal with many numbers. Scientist he worked on these many years in order to increase the efficiency of the algorithms. The results of their research are some of the algorithms like Karatsuba, Toom-Cook, Solaven-Strasen.

2.3 Multiplication Algorithms

Multiplication is a method of multiplying two numbers. Based on the length of the numbers different algorithms are used. More the size, more the time it takes for the multiplications. Sometimes it gets unacceptable if it takes more time for the result. Classical method of multiplication doesnt account for large numbers. Then, Scientist have worked on some methods to make the work done. This led to to the invention of some multiplication algorithms to accelerate the calculations. The concept known as complexity of computation is very much necessary in analyzing such algorithms. The efficiency of these algorithms is dependent on the complexity.

2.4 Divide And Conquer

When we have to deal with complicated mathematical issues the best method that comes into play is divide and conquer. In this the problem is divided into smaller pieces until certain level is reached and each of the smaller problems are solved recursively. Then each problem becomes easy to solve. All the sub problems are then combined to form a final solution. This Technique is used in some sorting techniques and multiplying large numbers like

Karatsuba and Toom-Cook. In multi-Processor machines that have shared memory systems, the sub-problems are executed on those processors thereby increasing the performance of the divide and conquer algorithm.

3 History

Toom Cook algorithm is developed by Andrei Toom in 1963 and is later improved and published by Stephen Cook in his Phd thesis. Toom Cook Algorithm is also referred as Toom 3 which is the collective name for all Toom Cook based algorithms. Toom Cook is the faster generalisation of the Karatsuba method. Unllike Karatsuba it deals with 3 parts rather than 2 parts which makes it even more complex.

4 Toom-Cook 3 Way Method

Toom cook algorithm is the advanced approach for splitting the numbers into parts. Toom cook n way reduces the product to 2*(n)-1 multiplications. Where n stands for 3. Let the operands considered are split into 3 pieces of equal length. The parts are written in terms of polynomials

$$X(t) = (X2)t^2 + X1(t) + X0$$

$$Y(t) = (Y2)t^2 + Y1(t) + Y0$$

base B = b is choosen , such that the number of digits of both x and m in base B is at most k (e.g., 3 in Toom-3). A typical choice for i is given by:

$$i = max\lfloor\lfloor log_b m\rfloor/k\rfloor, \lfloor\lfloor log_b n\rfloor/k\rfloor + 1$$

In our example we'll be choosing the value of

$$B = b^2 = 10^8$$

Then we will seperate x and y into base B digits x,y.: These 2 equations are multiplied to form w(t)=x(t)*y(t).

$$W(t) = w4 * t^4 + w3 * t^3 + w2 * t^2 + w1 * t + w0$$

The final w(t) is calculated through the value of t, although the final step is going to be the addition.

X(t) and Y(t) are calculated and multiplied by choosing some set of points, forming w(t).

Let the following points be (0,1,2,-1,inf)
t=0 x0 * y0,
t=1 (x2+x1+x0) * (y2+y1+y0)
t=-1 (x2-x1+x0) * (y2-y1+y0)

$$t = 2(4 * x^2 + 2 * x^1 + x0) * (4 * y^2 + 2 * y^1 + y0)$$

t=inf x2 * y2,
Then, the value of those combinations is calculated through
W(0) = w0
W(1) = w4 + w3 + w2 + w1 + w0
W(-1) = w4 - w3 + w2 - w1 + w0
W(2) = 16*w4 + 8*w3 + 4*w2 + 2*w1 + w0
W(inf) = w4
Toom-3 running time is significally

$$O(N^{1.465})$$

, the exponent log(5)/log(3), resembles 5 multiplies for 3 splits of each size.
This is an advance over Karatsuba algorithm which runs at

$$O(N^{1.585})$$

.

4.1 Algorithm

1. $$Input : TwointegersAandBaregivenwhere0 < A, B, < X^n$$

2. $$Output : AB = c0+c1x^k+c2x^{2k}+c3x^{3k}+c4x^{4k}whenk = n/(k = numberofsplits)$$

3. $$HereA = x0 + x1t + x2t^2, B = y0 + y1t + y2t^2heret = X^k$$

4. t=(0,1,2,-1,inf)

5. t=0 x0 * y0, which gives w0 immediately

6. t=1 (x2+x1+x0) * (y2+y1+y0)

7. t=-1 (x2-x1+x0) * (y2-y1+y0)

8.
$$t = 2(4 * x^2 + 2 * x1 + x0) * (4 * y^2 + 2 * y1 + y0)$$

9. t=inf x2 * y2, which gives w4 immediately

10. Each of the points are substituted to get

$$C(t) = c4 * t^4 + c3 * t^3 + c2 * t^2 + c1t^1 + c0$$

4.2 Complexity

Complexity is very essential to measure how long an algorithm is taking to run.. Complexity of a problem is defined using big O notation . These are further derivate into best case average and the worst cases. This helps to choose the best algorithm that suits our problem. The classical addition and multiplication method which e learned in our schools takes O(n2) for the multiplication, which is of no use when dealing with the large integers. The main task is to run our algorithm as fast as possible for efficient results.

Toom 3 splits the operand into n/2 parts

For ex: These 2 equations are multiplied to form w(t)=x(t)*y(t)

$$W(t) = w4 * t^4 + w3 * t^3 + w2 * t^2 + w1 * t + w0$$

The running time for the above equation takes T(n)=9T(n/3), T(c)=1 Where 9 is the total number of multiplications, this is of no use

The implemented algorithm takes 5 multiplication for 3 splits. Z0 = x0 y0 Z1 = (x0 + x1 + x2) (x0 + y1 + y2) Z2 = (x0 + 2 x1 + 4 x2) (y0 + 2 y1 + 4 y2) Z3 = (x0 x1 + x2) (y0 y1 + y2) Z4 =(x0 2 x1 + 4 x2) (y0 2 y1 + 4 y2)

Therefor we see that the for k-way split requires 2(k) - 1 multiplications.

Toom-3 running time is significally

$$O(N^{1.465})$$

, the exponent log(5)/log(3), resembles 5 multiplies for 3 splits of each size. This is an advance over Karatsuba algorithm which runs at

$$O(N^{1.585})$$

.

4.3 Implementation Of Toom-3 way Method

Toom-Cook is still one of the best techniques for multiplying big integers. Toom Cook-3 way is an improved method by a factor of 9/5 compared to Karatsuba methods where the number of multiplications have been reduced to 5 from 9. Toom Cook covers various cases under Toom Cook k where k equals to 3.

The implementation is shown in below example

Lets consider 2 numbers for the operation 831275469 by 897512436.

We can apply Toom-3 way on these numbers. Both the numbers are split into 3 limbs each of length 3 digits written into polynomials

$$P(x) = a2 * x^2 + a1 * x + a0(a2 = 831, a1 = 275, a0 = 469)$$

$$Q(x) = b2 * x^2 + b1 * x + b0(b2 = 897, b1 = 512, b0 = 436)$$

$$p(x) = 831x^2 + 275x + 469$$

$$q(x) = 897x^2 + 512x + 436$$

We write, p(x)q(x) = r(x).

$$(831x^2 + 275x + 469)(897x^2 + 512x + 436) = ax^4 + bx^3 + cx^2 + dx + e = r(x)$$

.

then we substitute the values of x for set of equations

Lets the set of points to be substituted be -2,-1,0,1,2

Now lets start with 0.

p(0)q(0) = r(0).

$$831(0)^2+275(0)+469)(897(0)^2+512(0)+436) = a(0)^4+b(0)^3+c(0)^2+dx+e = r(x)$$

Therefore,

e = 204484.

Now,

p(1)q(1) = r(1).

$$(831(1)^2+275(1)+469)(897(1)^2+512(1)+436) = a(1)^4+b(1)^3+c(1)^2+d(1)+e = r(x)$$

a+b+c+d+ e = 2905875 ——-equation 1

Now,

p(-1)q(-1) = r(-1).

$(831(-1)^2+275(-1)+469)(897(-1)^2+512(-1)+436) = a(-1)^4+b(-1)^3+c(-1)^2+d(-1)+e = r(x)$

.

a-b+c-d+e= 841525 ——equation 2
Now,
p(2)q(2) = r(2).

$(831(2)^2+275(2)+469)(897(2)^2+512(2)+436) = a(2)^4+b(2)^3+c(2)^2+d(2)+e = r(x)$

16a+8b+4c+2d+e= 21923464 ——equation 3
Now,
p(-2)q(-2) = r(-2).

$(831(-2)^2+275(-2)+469)(897(-2)^2+512(-2)+436) = a(-2)^4+b(-2)^3+c(-2)^2+d(-2)+e = r(x)$

16a-8b+4c-2d+e=1027116 ——-equation 4
The equations we got by those points are
e = 204484
a+ b+ c+ d+e= 2905875
a-b+ c- d+e= 841525
16a+8b+4c+2d+e = 21923464
16a-8b+4c-2d+e= 1027116
By solving the above equation we get
a=382828
c=-1286387
b=1397304
d=-365129
e=204484
The Fianl product is obtained by adding the numbers by shifting
382828
—-1397304
———1286387
————365129
——————-204484
————————————————-
38296772326152333484 —— This is the final product obtained.

5 Aplications

There are various areas where this algorithm application is done, which involves multiplication of large integers.

1. This method is used in McEliece Cryptosystems to overcome the drawbacks in terms of size of the encrypted key and transmission rate.

2. Big Number arithmetic.

3. In Cryptographic algorithms especially for reducing the complexity in encoding and decoding of the keys like ElGamal, RSA, Elliptical Curve-cryptosystems, and Diffie Hellman key exchange protocol.

4. Calculation of mathematical constants like Pi, e etc.

6 Conclusion

The Speed with which the operations are done not only depends on the type and the size of the numbers but also on the type of the multiplication algorithm. In this paper we analyzed how the operations have been done on the polynomials by performing Toom -3 way algorithm. Toom -3 way reduced the number of multiplication from 9 to 5, thereby reducing the complexity of the problem in solving it less time. In classical method it takes around On^2 multiplications whereas through toom-cook the running time would be $On^{1.465}$.

7 References

1.Bodrato, M., Zanoni, A.: Integer and Polynomial Multiplication: Towards Optimal Toom-Cook Matrices. Proceedings of the ISSAC 2007 Conference. ACM press, New York (2007)

2.D. Knuth. The Art of Computer Programming, Volume 2. Third Edition, Addison-Wesley, 1997. Section 4.3.3.A: Digital methods

3.Cook, Stephen A., On the Minimum Computation Time of Functions, 1966 PhD thesis, Harvard University Department of Mathematics.