Large integer multiplication in time O(n)

R. Rama Chander
(Recreational Math Enthusiast)
chander@pi-rim.com
Hyderabad, INDIA
July - 2020

Abstract

This paper attempts to disprove the asymptotic in time *O(n log n)* prediction of Schönhage-Strassen, claiming its 'best possible' result and remarking that no one will ever find a faster multiplication algorithm. Accordingly, this paper postulates that, the most desired complexity in time, i.e. *O(n)* is as achievable using only two basic arithmetic operations.

We present four algorithms for large integer multiplications. First algorithm is based on the place value approach and achieves the much desired complexity of *O(n)*, based on *Nearest Place Values (NPV)* approach. Second algorithm extends and improves Karatsuba algorithm for any ordered pairs greater than 2. It is important to remind that the present version of Karatsuba algorithm works only for ordered pairs of 2. Third algorithm called Addition and Subtraction *(AnS)* achieves time complexity of *O(n)* for very large integer multiplications using only *repeated additions and subtractions.* The fourth algorithm is called the Repeated Doubling Method *(RDM), which is an improvised version of AnS algorithm and achieves time complexity of O(n).*

Keywords: *computer algorithm, large integer multiplication algorithm, nearest place value algorithm, Karatsuba n-way algorithm, repeated addition and subtraction algorithm, repeated doubling algorithm.*

1. Introduction

Around 1956, Andrei Nikolajewitsch Kolmogorov conjectured that, the asymptotic time complexity of integer multiplication *is* $O(n^2)$, but during 1962 Anatoly Alexeevitch Karatsuba proved that the conjecture is false and showed that $O(n^{1.585})$ is possible.

In the year 1963 Andrei Leonovich Toom reduced it to $O(n^{1.46})$ & $O(n^{1.404})$ through Toom-3 and Toom 4. During 1966 Stephen Arthur Cook further refined the Toom-n algorithm and reduced it to $O(n^{1+\epsilon})$. *In the year* 1971, a breakthrough happened with Schönhage-Strassen algorithm running in time *O(n log n · log log n).*

For almost 35 years, Schönhage-Strassen's algorithm has been the fastest algorithm known for multiplying integers, with a time complexity *O(n log n · log log n)* for multiplying n-bit inputs .In 2007, Martin Fürer given asymptotically faster algorithm having time complexity $O(n \log n\, 2^{O(\log^* n)})$.

Arnold Schönhage and Volker Strassen predicted that, there should exist an algorithm that multiplies n digit numbers using essentially *n log n* basic operations**.** They also predicted that *O(n log n)* is the 'best possible' result and that no one will ever find a faster multiplication algorithm.

During last year April 2019, David Harvey, and Joris van der Hoeven have defined a new algorithm having asymptotic time complexity *O(n log n),* which has been predicted by Schönhage and Strassen.

Below *Table: 1a* provides chronological summary of the significant developments pertaining to integer multiplication and corresponding time complexity.

Year	*Algorithm*	*Time Complexity*
?!	Conventional Method	$O(n^2)$
1962	Karatsuba	$O(n^{\log 3/\log 2})$
1963	Toom	$O(\,n\, 2^{5\sqrt{\log n/\log 2}})$
1966	Schönhage	$O(n\, 2^{\sqrt{2\log n/\log 2}}(\log n)^{3/2})$
1969	Knuth	$O(n\, 2^{\sqrt{2\log n/\log 2}}\, \log n)$
1971	Schönhage - Strassen	$O(n\log n * \log n)$
2007	Fürer	$O(n \log n\, 2^{O(\log * n)})$
2019	Harvey - Hoeven	$O(n\log n)$
2020	In this paper	$O(n)$

Table: 1a-number of bit operations to multiply two n-bit integers

Our algorithm tries to achieve the much desired in time *O(n)* through two unique improvised approaches, and both algorithms having equal importance. The first algorithm is based on Vedic mathematics technique called the *Nearest Place values (NPV) method* and the second algorithm based on Egyptian, Babylonians & Average & Difference Square multiplication algorithm is called *Addition and Subtraction (AnS)* method.

2. The algorithms

We all know that, in arithmetic terms, repeated addition is called multiplication and addition & subtraction on large numbers is relatively having less time complexity than the multiplications. If 'n' is the order of the digits, we use only two basic arithmetic operations to achieve the asymptotic time complexity run in *O(n)* time.

The first algorithm we, will discuss about Integer multiplication using radix, place value, and face value to achieve the in time *O(n).* In the second algorithm we, will go through an improvised Karatsuba algorithm for ordered pair's like 3 way, 4 way and 5 way splits, to be used in divide and conquer approach for large integer multiplication. In the third algorithm we will discuss about Integer multiplication in time *O(n)* using *AnS* (Addition and Subtraction) approach, using repeated addition and subtractions. In the fourth algorithm using RDM (Repeated Doubling Method) approach is an improvised version of the third *(Ans)* algorithm to achieve in time *O(n).*

3. Vedic Mathematics

Before going through the first algorithm of this paper "The Nearest Place Values" approach, one must know the concepts of Vedic Mathematics and especially about the ***Nikhilam Navatashcaramam Dashatah (NND).***

Vedic mathematics is believed to be reconstructed by *Shri Bharathi Krishna Tirathaji (1884 -1960)* between 1911 and 1918. Vedic mathematics is divided into sixteen sutras (formulae) and thirteen upa-sutras (sub-formulae or corollaries), which can be applied to any branch of mathematics. Vedic mathematics sutras reduces the complexity of calculations using simple methods similar to the working of human minds.

The Sixteen Sutras and its meaning

Sl.No	Sutras	Meaning
1	*Ekadhikina Purvena*	By one more than the previous one
2	***Nikhilam Navatashcaramam Dashatah (NND)***	**All from 9 and the last from 10**
3	*Urdhva-Tiryagbyham*	Vertically and crosswise
4	*Paraavartya Yojayet*	Transpose and adjust
5	*Shunyam Saamyasamuccaye*	When the sum is the same that sum is zero
6	*(Anurupye) Shunyamanyat*	If one is in ratio, the other is zero
7	*Sankalana-vyavakalanabhyam*	By addition and by subtraction
8	*Puranapuranabyham*	By the completion or non-completion
9	*Chalana-Kalanabyham*	Differences and Similarities
10	*Yaavadunam*	Whatever the extent of its deficiency
11	*Vyashtisamanstih*	Part and Whole
12	*Shesanyankena Charamena*	The remainders by the last digit
13	*Sopaantyadvayamantyam*	The ultimate and twice the penultimate
14	*Ekanyunena Purvena*	By one less than the previous one
15	*Gunitasamuchyah*	The product of the sum is equal to the sum of the product
16	*Gunakasamuchyah*	The factors of the sum is equal to the sum of the factors

Table: 3a - The Sixteen Sutras of Vedic Mathematics

In this paper, we improve and use "*Nikhilam Navatashcaramam Dashatah (NND)*" sutra to achieve the desire*d O(n)* asymptotic time complexity. The literal meaning of sutra is *"All from 9 and the last from 10".* This sutra is used to convert the integer multiplication using few subtract, add and shift operations.

The steps for multiplication of 2 digit number, which is *less than* the nearest base using the NND sutra the base method when *'a'* and *'b'* < 100 are as shown below in *Table: 3b and Table: 3c:*

Step 1	*a * b*
Step 2	A = (Nearest Base) – *a*
Step 3	B = (Nearest Base) – *b*
Step 4	C = A * B
Step 5	D = (*a* – B) 'or' (*b* – A)
Step 6	Result = (100 * D) + C

Table: 3b - when 'a' and 'b' < 100

As an example, consider to find 94 * 97 which is less than and very much closer to the nearest base 100,

Step 1	94 * 97
Step 2	A = 100 – 94 = 6
Step 3	B = 100 – 97 = 3
Step 4	C = 6 * 3 = 18
Step 5	D = (94 – 3) 'or' (97 – 6) = 91
Step 6	Result = (100 * 91) + 18 = 9100 + 18 = **9118**

Table: 3c – worked example: when 'a' and 'b' < 100

The steps for multiplication of 2 digit number which is *greater than* the nearest base using the NND sutra, the base method when 'a' and 'b' > 100 are as shown below in Table: 3d and Table: 3e.

Step 1	*a * b*
Step 2	(Nearest Base) – *a* = – A
Step 3	(Nearest Base) – *b* = – B
Step 4	C = (– A * – B)
Step 5	D = *a* – (–B) 'or' *b* – (–A)
Step 6	Result = (100 * D) + C

Table: 3d - when 'a' and 'b' > 100

As an example, consider to find 103 * 108 which is greater than and very much closer to the nearest base 100.

Step 1	103 * 108
Step 2	A = 100 – 103 = – 3
Step 3	B = 100 – 108 = – 8
Step 4	C = (– 3 * – 8) = 24
Step 5	D = 103 – (– 8) ‘or’ 108 – (– 3) = 111
Step 6	Result = (100 * 111) + 18 = 11100 + 24 = **11124**

Table: 3e – worked example: when ‘a’ and ‘b’ > 10

The steps for multiplication of 2 digit number in which one number, which is a is greater than and the other number lesser than the nearest base 100, using the NND sutra. The base method when ‘a’ > 100 and ‘b’ < 100 are as shown below in Table: 3f and Table: 3g:

Step 1	*a * b*
Step 2	(Nearest Base) – *a* = – A
Step 3	B = (Nearest Base) – *b*
Step 4	C = (– A * B)
Step 5	D = *a* – (B) ‘or’ *b* – (– A)
Step 6	Result = (100 * D) – C

Table: 3f – when ‘a’ > 100 and ‘b’< 100

As an example, consider to find 104 * 97 in which 97 is less than and 100 is greater than the nearest base 100,

Step 1	104 * 97
Step 2	A = 100 – 104 = – 4
Step 3	B = 100 – 97 = 3
Step 4	C = (– 4 * 3) = – 12
Step 5	D = 104 – 3 ‘or’ 97 – (– 4) = 101
Step 6	Result = (100 * 101) – 12 = 11100 – 12 = **10088**

Table: 3g – worked example: when ‘a’ > 100 and ‘b’< 100

Using *NND sutra*, a 2 x 2 digit multiplication operation can be performed in 1 multiplication, whereas classical multiplication requires 4 multiplication operations.
In the first algorithm of this paper *“The Nearest Place Values”* approach, we will eliminate the 1 multiplication needed by the *NND sutra* and achieve in time *O(n).*

4. Karatsuba Algorithm

Karatsuba algorithm is based on divide and conquer strategy. Karatsuba algorithm can multiply a 2 digit multiplication in 3 multiplication steps instead of 4 steps. A generalization of two digit multiplication using Karatsuba algorithm for $a_1a_2 * b_1b_2$ is shown below:

1. A = $a_1 * b_1$
2. B = $a_2 * b_2$
3. C = $(a_1 – a_2) * (b_1 – b_2)$

   (Note in this paper we, opted the negative variation of the Karatsuba algorithm to find the value of C, which is an improvised version over the original algorithm and reduces the overall complexity when the opted numbers a_1, a_2, b_1, b_2 are greater than 5.)
4. D = A + B – (C) here D = $(a_1 * b_2) + (a_2 * b_1)$
5. Result = (100 * A) + (10 * D) + B

As an example, consider multiplication of 48 * 67, using Karatsuba algorithm. Let a_1 = 4, a_2 = 8 and b_1 = 6, b_2 = 7:

1. A = 4 * 6 = 24
2. B = 8 * 7 = 56
3. C = (4 – 8) * (6 – 7) = (– 4) * (– 1) = 4
4. D = 24 + 56 – (4) = 80 – 4 = 76
5. Result = (100 * 24) + (10 * 76) + 56 = 2400 + 760 + 56 = 3216

   So, **48*67 = 3216**

The shift, addition and subtraction operations can be ignored as multiplication operation is most costly. For a 2 x 2 digit multiplication, Karatsuba algorithm used only 3 multiplications steps instead of 4. The time complexity of Karatsuba algorithm is = $O(n^{\log 3/\log 2}) \approx O(n^{1.585})$.

5. Toom - 2 Algorithm

Toom – Cook algorithm is based on our realization that any integer can be written as a polynomial. Toom-2 algorithm can multiply a 2 digit multiplication in 3 multiplication steps instead of 4 steps. A generalization of two digit multiplication using Toom-2 for $a_1a_2 * b_1b_2$ is shown below:

$p(x) = (a_1x + a_2)$
$q(x) = (b_1x + b_2)$
$r(x) = ax^2 + bx + c$

$r(x) = p(x)*q(x)$
$r(x) = ax^2 + bx + c = (a_1x + a_2)\ (b_1x + b_2)$

Example we want to multiply 48 x 67
48 can be written as $p(x) = (a_1x + a_2) = (4x + 8)$ and $q(x) = (b_1x + b_2) = (6x + 7)$ where $x = 10$, which is the base value.
So r(x) = $ax^2 + bx + c = (4x + 8)\ (6x + 7)$
Substituting *'x'* value as 0, 1 and finding 'inf' (infimum) or 'GLB' (Greatest Lower Bound)

Now considering $x = 0$:
$r\ (0) = p\ (0)*q\ (0)$
So, a (02) + b (0) + c = [4(0) +8] [6(0) +7]
c = [8] [7]
Therefore, c = 56

Now considering $x = 1$: $r\ (1) = p\ (1)*q\ (1)$
So, a (12) + b (1) + c = [4(1) +8] [6(1) +7]
= $a + b$ + 56 = [4 +8] [6 +7]
= $a + b$ + 56 = 12 x 13
= $a + b$ = 156 – 56
= $a + b$ = 100 ……………………………………………………Equation 1

Now considering 'inf' (infimum) or 'GLB' (Greatest Lower Bound)
So, 'inf' of $ax^2 + bx + c = (4x + 8)\ (6x + 7)$
= $ax^2 = (4x)\ (6x)$
= a = (4) (6)
= a = 24

Now, we already know the value of 'c' which is equal to 56, and 'a' = 24
To find the value of 'b' by substituting the value of 'a' in the Equation 1
$a + b = 100$
$24 + b = 100$
$b = 100 – 24 = 76$

Solving the above equations we get a=24 and b=76. And we know the value of c, which is equal to 56
Substituting the values of a = 24, b = 76 and c = 56, in $ax2 + bx + c$, and considering the base value of x =10

24 (10^2) +76(10) +56
2400 +760 +56 =3216
So, **48 x 67 =3216**
A Toom -2 algorithm needs 3 multiplications. The time complexity of Toom-2 algorithm is = $O(n^{\log 3/\log 2}) \approx O(n^{1.585})$.

6. An improvised version of the *NND* method - the Nearest Place Values method *(NPV)*

A 2 x 2 digit multiplication needs the 3 multiplications to achieve the result as per Karatsuba as explained in (Section 4) and Toom-2 algorithms (Section 5).

We have also demonstrated that, for a 2 x 2 digit multiplication using the base method (as explained in Section 3), we need only 1 multiplication (when the two numbers are nearest to the base value 100).

In this section, we propose nearest place values approach for which we do not need any multiplications at all to multiply any two integers and absolutely achieves much desired *O(n)* time complexity.

The proposed nearest value approach is based on and combination of Vedic mathematics (discussed in section 3) and divide and conquer algorithm. The salient features of proposed approach are:

- Any given order can be multiplied
- The time complexity achieved is equal to *O(n)*, i.e., absolutely with *zero* number of multiplications.

There are *two unique methods to obtain O(n) time complexity* using proposed nearest place values approach. The two methods are namely: the nearest Ten's method and the nearest Hundred's method for a 2x2 multiplication. We can understand the concepts behind the two methods with a few examples in the remainder of this section.

The nearest Ten's method

***Approach1**: Nearest Ten's Method for 2x2 digits order.* To find multiplication of 48 * 67 and to achieve the *O(n)*.
Note: The place value of 4 and 6 is Ten's:

Ten's = $(10)^1$	*X*	*Y*	*(X+Y)-10*	*The steps to arrive the needed value of Ten's place*
T_1	48	67	105	T_1=(X+Y)-10=(48+67)-10 =115-10 =105
$T_2=T_1-10$	38	57	85	$T_2=(T_1-10)$=(48-10) & (67-10), (X+Y)-10=(38+57)-10=95-10=85
$T_3=T_2-10$	28	47	65	$T_3=(T_2-10)$=(38-10) & (57-10),(X+Y)-10=(28+47)-10=75-10=65
$T_4=T_3-10$	18	37	45	$T_4=(T_3-10)$=(28-10) & (47-10), (X+Y)-10=(18+37)-10=55-10=45
$T_5=T_4-10$	8	27	25	$T_5=(T_4-10)$=(18-10) & (37-10), (X+Y)-10=(8+27)-10=35-10=25
(X)*(Y) = +Ve, So to Add		***T =***	***+325***	***The final value of T = Sum of the values from T_1 to T_5 = +325***
One's = $(10)^0$	*-X'*	*Y'*	*(-X'+Y')+1*	*The steps to arrive the needed value of One's place*
$O_1=T_5-10$	-2	17	16	$O_1=T_5-10$=(8-10)& (27-10), (-X'+Y')+1=-2+17+1=-2+18=16
$O_2=O_1+1$	-1	18	18	$O_2=O_1+1$=(-2+1) & (17+1), (-X'+Y')+1=-1+18+1=-1+19=18
(-X')*(Y') = -Ve, So to Subtract		***O =***	***-34***	***The final value of O=Sum of the values from O_1 to O_2 = -34***

Table: 6a - Nearest Ten's Method for 2x2 digits

The procedures for the Nearest Place Values *(NPV)* Method for multiplication

- *The method to find the product of 'X' * 'Y' is to finally arrive to a unit digit multiplication (multiplicand to reach and end at (1* ≤ Y') or 'multiplier to reach and end at (X' ≥ *1)) in the least possible steps using the very Nearest Place Values multiplication approach appropriately.*
- *To derive the next row values, the value of 'X' the multiplicand and 'Y' the multiplier are to be added first, and then add or subtract the appropriate nearest place value to arrive to the unit digit multiplication in the shortest possible steps which is equal to the nearest place value. The method continues step by step lessening 'X' and 'Y' values till arriving and end at unit digit multiplication.*
- *All the rows final values of so arrived are to be grouped according to their place value and to be added together to get the total of that particular place value and to be multiplied with that place value.*
- *If the arrived row values of both 'X' and 'Y' are positive or negative, then the final value should be added because the product of (+X)*(+Y) (or) (-X)*(-Y) are = +XY: (+Ve). Similarly, if any one of the values of 'X' or 'Y' is negative, the final value should be subtracted because the product of (-X)*(+Y) (or) (+X)*(-Y) is = –XY: (-Ve).*

Using the following appropriate expansion for a 2 digit multiplication
$X * Y = O (10^0) + T (10^1) =$
$48 * 67 = (-34)(10^0) + 325 (10^1) = -34 + 3250 =$ ***3216***

Concept behind the *O(n)* using nearest Ten's algorithm for a two digit multiplication

2-digit integers and having Radix 10, can be expanded. i.e., *ab = 10a +b* and *cd = 10c +d*.
(10a +b) (10c +d), we will try to understand the concept behind the nearest Ten's approach to achieve *O(n)*:

Ten's = $(10)^1$	*X*	*Y*	*(X + Y)-10*
T_1	10a +b	10c +d	10a +b +10c +d -10
$T_2=T_1-10$	10a +b -10	10c +d -10	10a +b +10c +d -30
$T_3=T_2-10$	10a +b -20	10c +d -20	10a +b +10c +d -50
$T_4=T_3-10$	10a +b -30	10c +d -30	10a +b +10c +d -70
$T_5=T_4-10$	10a +b -40	10c +d -40	10a +b +10c +d -90
(X)*(Y) = +Ve, So to Add	***T = Sum all terms of (X +Y)-10 =***		***50a +5b +50c +5d -250***
One's = $(10)^0$	*-X'*	*Y'*	*(-X'+Y')+1*
$O_1=T_5-10$	10a +b -50	10c +d -50	10a +b +10c +d -99
$O_2=O_1-1$	10a +b -50 -1	10c +d -50 -1	10a +b +10c +d -97
(-X')*(Y') = -Ve, So to Subtract	***T = Sum all terms of (-X' +Y')+1 =***		***- (20a +2b +20c +2d -196)***

Table: 6b - Concept behind nearest Ten's algorithm for 2x2

Using the following expansion
$X * Y = (10a + b)*(10c + d) = O (10^0) + T (10^1) + H (10^2)$
$T = (50a + 5b + 50c + 5d - 250) (10) = 500a + 50b + 500c + 50d - 2500$
$O = -20a - 2b - 20c - 2d + 196$

$$
\begin{array}{rrrrrr}
 & 500a & +50b & +500c & +50d & -2500 \\
 & \underline{-20a} & \underline{-2b} & \underline{-20c} & \underline{-2d} & \underline{+196} \\
= & \underline{\mathbf{480a}} & \underline{\mathbf{+48b}} & \underline{\mathbf{+480c}} & \underline{\mathbf{+48d}} & \underline{\mathbf{-2304}}
\end{array}
$$

$= 480a + 48b + 480c + 48d - 230 = 48 (10a + b + 10c + d - 48)$
Substituting 48 = $10a + b$
$= (10a + b) (10a + b + 10c + d - (10a + b)) = (10a + b) (\cancel{10a+b} + 10c + d \cancel{-10a-b}) = (10a + b) (10c + d)$

Nearest Ten's algorithm for a two digit multiplication

To find the product of a two 2 digit numbers. The two 2 digit can be written as *(10a +b)* and *(10c +d)*
To find *(10a +b)* (10c +d)*
$(10a + b)*(10c + d) = 100ac + 10ad + 10bc + bd$
Also *(10a +b)*(10c +d)* can be expanded in two different ways

Expansion-1: $(a+1)(10c+d+b-10)(10^1)+(b-10)(10c+d-10a-10)(10^0)$

Or

Expansion-2: $(c+1)(10a+b+d-10)(10^1)+(d-10)(10a+b-10c-10)(10^0)$

Consider the example which we have seen: 48 x 67 from the *Table: 6a*, by observation the nearest Ten's method,
The total sum of ten's values, which we have arrived using nearest Ten's value method is 325 x 10. Which can be arrived using the first part of the *expansion-1*, which is $(a+1)(10c+d+b-10)(10)$
Let 48 = (4*10 +8) and a = 4 and b = 8 and 67 = (6*10+7) and c = 6 and d = 7
By substituting the values of *a, b, c* and *d,* in the first part of the expansion-1, which is $(a+1)(10c+d+b-10)(10)$
= (4 +1) (67 +8 –10) (10)
= (5) (65) (10) = 3250

The total sum of one's values, which we have arrived using nearest Ten's value method is –34. Which can be arrived using the second part of the expansion 1, which is $(b-10)(10c+d-10a-10)$
By substituting the values of *a, b, c* and *d*, in the second part of the expansion-1, which is $(b-10)(10c+d-10a-10)$
= (8 –10) (67 –40 –10) (10^0)
= (8 –10) (67 –50) (10^0)
= (–2) (17) (1) = –34

By adding the ten's value and one's value
3250 –34 = 3216
So, **48 x 67 = 3216**

So, technically the expansion-1: $(a+1)(10c+d+b-10)(10)+(b-10)(10c+d-10a-10)(10^0)$ has been used to find the product of a 48 x 67, using nearest Ten's method without using any single multiplication.

The nearest Hundred's method

Approach 2*: Nearest Hundreds Method for 2x2 digits order. To find 48*67. Note: The place value of 4 and 6 is Ten's*, and we consider the next base value, which is 100 for the nearest Hundred's method.

Hundred's = $(10)^2$	*X*	*Y*	*(X+Y)-100*	*The steps to arrive the needed value of Hundred's place*
H	48	67	15	*The place value of 4 in X and 6in Y is 10. The next place value after 10 is 100. So H=(X+Y)-100=(48+67)-100=115-100=15*
X*Y = +Ve, So to Add		***H =***	***+15***	***The final value of H=15***
Ten's = $(10)^1$	*-X'*	*-Y'*	*(-X'-Y')+10*	*The steps to arrive the needed values of Ten's place.*
T_1=H-100	-52	-33	-75	*T_1=(H-100)=(48-100)&(67-100), (-X'-Y')+10=(-52-33)+10=-85+10=-75*
T_2=T_1+10	-42	-23	-55	*T_2=(T_1+10)=(-52+10)&(-33+10), (-X'-Y')+10=(-42-23)+10=-65+10=-55*
T_3=T_2+10	-32	-13	-35	*T_3=(T_2+10)=(-42+10)&(-23+10), (-X'-Y')+10=(-32-13)+10=-45+10=-35*
(-X')*(-Y') = +Ve, So to Add		***T =***	***+165***	***The final value of T=Sum of the values from T_1 to T_3 = +165***
One's = $(10)^0$	*-X''*	*-Y''*	*(-X''-Y'')+1*	*The steps to arrive the needed value of One's place*
O_1=T_3+10	-22	-3	-24	*O_1=(T_3+10)=(-32+10)&(-13+10), (-X''-Y'')+1=(-22-3)+1=-25+1=-24*
O_2=O_1+1	-21	-2	-22	*O_2=(O_1+1)=(-22+1)&(-3+1), (-X''-Y'')+1=(-21-2)+1=-23+1=-22*
O_3=O_2+1	-20	-1	-20	*O_3=(O_2+1)=(-21+1)&(-2+1), (-X''-Y'')+1=(-20-1)+1=-21+1=-20*
(-X'')*(-Y'')= +Ve, So to Add		***O =***	***+66***	***The final value of T=Sum of the values from O_1 to O_3 = +66***

Table: 6c - The nearest Hundred's method for 2x2

Using the following appropriate expansion for a 2 digit multiplication
$X * Y = O(10^0) + T(10^1) + H(10^2)$
$48 * 67 = 66(10^0) + 165(10^1) + 15(10^2) = 66 + 1650 + 1500 =$ ***3216***

Concept behind the *O(n)* using nearest Hundred's algorithm for a two digit multiplication

Let the two numbers are (10a +b) and (10c +d)

Hundred's = $(10)^2$	*X*	*Y*	*(X + Y -100)*
H	*10a +b*	*10c +d*	*10a +b +10c +d -100*
X * Y = (+Ve) , So to Add		***H = Sum all terms of (X +Y -100) =***	***10a +b +10c +d -100***
Ten's = $(10)^1$	*-X'*	*-Y'*	*(-X' –Y') +10*
T_1=H-100	*10a +b -100*	*10c +d -100*	*10a +b +10c +d -190*
T_2=T_1+10	*10a +b -100 +10*	*10c +d -100 +10*	*10a +b +10c +d -170*
T_3=T_2+10	*10a +b -100 +20*	*10c +d -100 +20*	*10a +b +10c +d -150*
(-X') * (-Y') = +Ve, So to Add		***T = Sum all terms of (-X'-Y')+10 =***	***- (30a +3b +30c +3d -510)***
One's = $(10)^0$	*-X''*	*-Y''*	*(-X'' –Y'') +1*
O_1=T_3+10	*10a +b -100 +30*	*10c +d -100 +30*	*10a +b +10c +d - 139*
O_2=O_1+1	*10a +b -100 +31*	*10c +d -100 +31*	*10a +b +10c +d - 137*
O_3=O_2+1	*10a +b -100 +32*	*10c +d -100 +32*	*10a +b +10c +d - 135*
(-X'') * (-Y'') = +Ve, So to Add		***O = Sum all terms of (-X'' –Y'')+1 =***	***-(30a +3b +30c +3d -411)***

Table: 6d - Concept behind nearest Hundred's algorithm for 2x2

Using the following expansion
$X * Y = (10a + b)*(10c + d) = O(10^0) + T(10^1) + H(10^2)$
$H = (10a + b + 10c + d - 100)(10^2) = 1000a + 100b + 1000c + 100d - 10000$
$T = -(30a + 3b + 30c + 3d - 510)(10^1) = -300a - 30b - 300c - 30d + 5100$
$O = -(30a + 3b + 30c + 3d - 411) = -30a - 3b - 30c - 3d + 411$

$1000a$	$+100b$	$+1000c$	$+100d$	-10000
$-300a$	$-30b$	$-300c$	$-30d$	$+5100$
$-30a$	$-3b$	$-30c$	$-3d$	$+411$
$= 670a$	**$+67b$**	**$+670c$**	**$+67d$**	**-4489**

$= 670a + 67b + 670c + 67d - 4489 = 67(10a + b + 10c + d - 67)$
Substituting $67 = 10c + d$
$= (10c + d)(10a + b + 10c + d - (10c + d)) = (10c + d)(10a + b \cancel{+10c + d - 10c - d}) = (10c + d)(10a + b)$

Nearest Hundred's algorithm for a two digit multiplication

To find the product of a two 2 digit numbers. The two 2 digit can be written as $(10a + b)$ and $(10c + d)$

To find the product of $(10a + b)*(10c + d)$
$(10a + b)*(10c + d) = 100ac + 10ad + 10bc + bd$
Also $(10a + b)*(10c + d)$ can be expanded in two different ways

Expansion-3: $(10a + b + 10c + d - 100)(10^2) + (a - 9)(10c + d + b - 110)(10^1) + (b - 10)(10c + d - 10a - 10)(10^0)$

Or

Expansion-4: $(10a + b + 10c + d - 100)(10^2) + (c - 9)(10a + b + d - 110)(10^1) + (d - 10)(10a + b - 10c - 10)(10^0)$

Consider the example which we have seen: 48 x 67 from the *Table: 6c*, by observation the nearest Hundred's method, The total sum of hundred's value, which we have arrived using nearest Hundred's value method is 15 x 100. Which can be arrived using the first part of the *expansion-4*, which is $(10a + b + 10c + d - 100)(10^2)$

Let 48 = (4*10 +8) and $a = 4$ and $b = 8$ and 67 = (6*10 +7) and $c = 6$ and $d = 7$
By substituting the values of *a, b, c* and *d,* in the first part of the expansion 4, which is $(10a + b + 10c + d - 100)(10^2)$
$= (48 + 67 - 100)(10^2)$
$= (15)(100) = 1500$

The total sum of ten's values, which we have arrived using nearest Hundred's value method is 165 x 10. Which can be arrived using the middle part of the expansion 4, which is $(c - 9)(10a + b + d - 110)(10^1)$
By substituting the values of *a, b, c* and *d,* in the middle part of the expansion-4, which is $(c - 9)(10a + b + d - 110)(10^1)$
$= (6 - 9)(48 + 7 - 110)(10^1)$
$= (-3)(55 - 110)(10^0)$
$= (-3)(-55)(10) = 1650$

The total sum of one's values, which we have arrived using nearest Hundred's value method is 66 x 1. Which can be arrived, using the last part of the expansion 4, which is $(d - 10)(10a + b - 10c - 10)(10^0)$
By substituting the values of a, b, c and d in the last part of the expansion-4, which is $(d - 10)(10a + b - 10c - 10)(10^0)$
$= (7 - 10)(48 - 60 - 10)(10^0)$
$= (-3)(48 - 70)(10^0)$
$= (-3)(-22)(1) = 66$

By adding the hundred's, ten's value and one's values
1500 +1650 +66 = 3216
So, **48 x 67 = 3216**

So, technically the expansion-4: $(10a + b + 10c + d - 100)(10^2) + (c - 9)(10a + b + d - 110)(10^1) + (b - 10)(10a + b - 10c - 10)(10^0)$ has been used to find the product of a 48 x 67, using nearest Hundred's method without using any single multiplication.

3 x 3 Digit Multiplication

We can understand the *O(n)* concepts with one more worked out examples,
For a 3 x 3 digit multiplication
Example: 798 * 654 to achieve the *O(n)*

***Approach 1**: Nearest Hundreds Method for 3x3 digits. To find 798*654*

Hundred's = $(10)^2$	*X*	*Y*	*(X+Y)-100*	*The steps to arrive the needed value of Hundred's place*
H_1	798	654	1352	*The place value of 7in X and 6 in Y is 100.* *H_1=(X+Y)-100=(798+654)-100=1452-100=1352*
H_2=H_1-100	698	554	1152	*H_2=(H_1-100)=(798-100)&(654-100), (X+Y)-100=(698+554)-100=1252-100=1152*
H_3=H_2-100	598	454	952	*H_3=(H_2-100)=(698-100)&(554-100), (X+Y)-100=(598+454)-100=1052-100=952*
H_4=H_3-100	498	354	752	*H_4=(H_3-100)=(598-100)&(454-100), (X+Y)-100=(498+354)-100=852-100=752*
H_5=H_4-100	398	254	552	*H_5=(H_4-100)=(498-100)&(354-100), (X+Y)-100=(398+254)-100=652-100=552*
H_6=H_5-100	298	154	352	*H_6=(H_5-100)=(398-100)&(254-100), (X+Y)-100=(298+154)-100=452-100=352*
H_7=H_6-100	198	54	242	*H_7=(H_6-100)=(298-100)&(154-100), (X+Y)-100=(198+54)-100=252-100=152*
H_8=H_7-100	98	-46	-48	*H_8=(H_7-100)=(198-100)&(54-100), (X+Y)-100=(98-46)-100=52-100=-48*
X*Y = +Ve, So to Add		***H=***	***+5216***	***The final value of H=Sum of the values from H_1 to H_8 = +5216***
One's = $(10)^1$	*-X'*	*-Y'*	*(-X' –Y')+1*	*The steps to arrive the nearest value of Ten's place*
O_1=H_8-100	-2	-146	-147	*O_1=(H_8-100)=(98-100)&(-146-100), (-X'-Y')+1=(-2-146)+1=-148+1=-147*
O_2=O_1+1	-1	-145	-145	*O_2 =(O_1+1)=(-2+1)&(-146+1), (-X'-Y')+1=(-1-145)+1=-146+1 =-145*
(-X')*(-Y') = +Ve, So to Add		***T =***	***+292***	***The final value of H= Sum of the values from O_1 to O_2 = +292***

Table: 6e - Nearest Hundreds Method for 3x3

Using the following appropriate expansion for a 3 digit multiplication

*X * Y = O (10^0) + T (10^1) + H (10^2)*

798*654 = 292 (10^0) +0 (10^1) +5216 (10^2)

= 292 +0 +521600 = ***521892***

***Approach 2:** Nearest Thousands Method for 3x3 digits order. To find 798 * 654*

Thousand's = $(10)^3$	*X*	*Y*	*(X+Y)-1000*	*The steps to arrive the needed value of Thousand's place*
Th	798	654	452	*The place value of 7 in X and 6 in Y is 100. The next place value after 100 is 1000.So Th=(X+Y)-1000=(798+654)-1000=1452 -1000=452*
X*Y = +Ve, So to Add		***Th =***	***+452***	***The final value of Th=+452***
Hundred's = $(10)^2$	*-X'*	*-Y'*	*(-X'-Y')+100*	*The steps to arrive the needed value of Hundred's place*
H_1=Th-1000	-202	-346	-448	*H_1=(Th-1000)=(798-1000)&(654-1000), (-X'-Y')+100=(-202-346)+100 =-548+100=-448*
H_2=H_1+100	-102	-246	-248	*H_2=(H_1+100)=(-202+100)&(-346+100), (-X'-Y')+100=(-102-246)+100 =-348+100=-248*
(-X')*(-Y') = +Ve, So to Add		***H =***	***+696***	***The final value of H= Sum of the values from H_1 to H_2 = +696***
One's = $(10)^0$	*-X''*	*-Y''*	(-X''-Y'')+1	*The steps to arrive the needed value of One's place*
O_1=H_2+100	-2	-146	-147	*O_1=(H_2+100)=(-102+100)&(-246+100), (-X''-Y'')+1=(-2-146)+1=-148+1=-147*
O_2=O_1+1	-1	-145	-145	*O_2=(O_1+1)=(-2+1)&(-146+1), (-X''-Y'')+1=(-1-145)+1=-146+1=-145*
(-X'')*(-Y'') = +Ve, So to Add		***O =***	***+292***	***The final value of O = Sum of the values from O_1 to O_2 = +292***

Table: 6f - Nearest Thousands Method for 3x3

Using the following appropriate expansion for a 3 digit multiplication

*X * Y = O (10^0) + T (10^1) + H (10^2) + Th (10^3)*

798 * 654 = 292 (10^0) +0 (10^1) +696 (10^2) +452 (10^3) = 292 +69600 +452000 =***521892***

7. Improvised Karatsuba algorithm for n-way splits

We understand that the 2 x 2 Karatsuba Algorithm (Section 4) and Toom-2 (Section 5) needs total of 3 multiplication operations. In this Section, we present an improvised Karatsuba algorithm for n-way splits.

The algorithm for 3 x 3 (3 way algorithm)

*a b c * d e f*

*A = (a * d)*

*B = (b * e)*

*C = (c * f)*

D = A +B – ((a –b) (d –e))

E = A +B +C – ((a –c) (d –f))

F = B +C – ((b –c) (e –f))

G = C (10^0) +F (10^1) +E (10^2) +D (10^3) +A (10^4)

From the above example, it is clear that 6 single digit multiplications are required for the Karatsuba 3 x 3 multiplication, which is 1 more than the Toom-3 algorithm, and 1 less than conventional Karatsuba 2 way Algorithm. In addition, we can eliminate the interpolation and the evaluation needed by Toom-3 algorithm.

Let us consider an example to demonstrate improvised Karatsuba algorithm, to find the product of 368 * 897:

Here a = 3, b = 6, c =8 and d = 8, e = 9, f = 7,

A = *(a * d)* = (3 * 8) = 24

B = *(b * e)* = (6 * 9) = 54

C = *(c * f)* = (8 * 7) = 56

D = *A +B – ((a –b) (d –e))* = 24 +54 – ((3 –6) * (8 –9))

= 78 – ((–3) * (–1)) = 78 –3 = 75

E = *A +B +C – ((a –c) (d –f))* =24 +54 +56 – ((3 –8) * (8 –7))
= 134 – ((–5) * 1) = 134 +5 = 139
F = *B +C – ((b –c) (e –f))* = 54 +56 – ((6 –8)) * (9 –7))
= 110 – ((–2) * 2) = 110 +4 =114
Substituting the above five values in
G = C (10^0) *+F* (10^1) *+E* (10^2) *+D* (10^3) *+A* (10^4)
G = 56(10^0) +114(10^1) +139(10^2) +75(10^3) +24 (10^4)
= 56 +1140 +13900 +75000 +240000 = ***330096***
368 * 897 = 330096
The above 3 x 3 Karatsuba algorithm method can be extended to any number of equal pairs.
Improvised Karatsuba Algorithm for n-ordered pairs.
The algorithm for 4 x 4 (4 way algorithm)
4 x 4 digit multiplication
*a b c d *e f g h*
*A = (a * e)*
*B = (b * f)*
*C = (c * g)*
*D = (d * h)*
E = A +B – ((a –b) (e –f))
F = A +B +C – ((a –c) (e –g))
G = A +B +C +D – ((a –d) (e –h) – ((b –c) (f –g))
H = B +C +D – ((b –d) (f –h))
I = C +D – ((c –d) (g –h)
J = D (10^0) ***+I*** (10^1) ***+H*** (10^2) ***+G*** (10^3) ***+F ((***10^4***) +E*** (10^5) ***+A*** (10^6)
As demonstrated above, total of 10 single digit multiplications needed for the Karatsuba 4 x 4 multiplication.

Improvised Karatsuba Algorithm for 5 x 5 (5 way algorithm)
5 x 5 digit multiplication
*a b c d e * f g h i j*
*A = (a * f)*
*B = (b * g)*
*C = (c * h)*
*D = (d * i)*
*E = (e * h)*
F = A +B – ((a –b) (f –g))
G = A +B +C – ((a –c) (f –h))
H = A +B +C +D – ((a –d) (f –i)) – ((b –c) (g –h))
I = A +B +C +D +E – ((a –e) (f –j)) – ((b –d) (g –i))
J = B +C +D +E – ((b –e) (g –j)) – (c –d) (h –i))
K = C +D +E – ((c –e) (h –j))
L = D +E – ((d –e) (I –j))
M = E (10^0) ***+L*** (10^1) ***+K*** (10^2) ***+J*** (10^3) ***+I ((***10^4***) +H*** (10^5) ***+G*** (10^6) ***+ F*** (10^7) ***+A*** (10^8)
As demonstrated above, total of 15 single digit multiplications needed for the Karatsuba 5 x 5 multiplication.

For 6 x 6 multiplication, we need 21 single digit multiplications and similarly for 7 x 7 multiplication, we need 28, and for 8 x 8 multiplication, we need 36 single digit multiplications.
In general for an n ordered pair, the improvised Karatsuba algorithm requires, *(n) ((n+1)/2)* single digit multiplications.
It is important to note that, for other ordered pairs, using appropriate divide and conquer and Karatsuba improvised algorithm, the overall time complexity can be reduced.

8. Integer multiplication to achieve in time O(n) – using nearest place values and divide and conquer method

*To find 831 275 469 * 897 512 436 using divide and conquer through O(n) approach.*

Using the improvised 3x3 Karatsuba algorithm and *O(n) nearest place values* approach, one can find the product.

We can opt for any one of the *Nearest Hundreds Method or Nearest Thousands Method to find the product according to the merit.*

*To find the product of 831 275 469 * 897 512 436, using 3x3 Improvised Karatsuba algorithm. Let* a =831, b =275, c =469 and d =897, e =512, f =436.
A = 831* 897
B = 275 * 512
C = 469 * 436
D = A +B – [(831 –275) (897 –512)]
E = A +B +C – [(831 –469) (897 –436)]
F = B +C – [(275 –469) (512 –436)]
G = C $(10^0)^3$ *+F* $(10^1)^3$ *+E* $(10^2)^3$ *+D* $(10^3)^3$ *+A* (10^4)

A=831*897 using *Nearest Thousand's Method*

Thousand's=10^3	*X*	*Y*	*(X+Y)-1000*
Th	831	897	*728*
(X)*(Y) = +Ve, So to Add		***Th***	***+728***
Hundred's=10^2	*-X'*	*-Y'*	*(-X' –Y')+100*
H_1=Th-1000	-169	-103	*-172*
(-X')*(-Y') = +Ve, So to Add		***H =***	***+172***
One's=10^0	*-X''*	*-Y''*	*(-X'' –Y'')+1*
O_1=H_1+100	-69	-3	*-71*
O_2=O_1+1	-68	-2	*-69*
O_3=O_2+1	-67	-1	*-67*
(-X'')*(-Y'') = +Ve, So to Add		***O =***	***+207***

Table: 8a – worked example: 831*897

The 3 values Th =728, H =172, T =0 and O =207
*X * Y =O (10^0) +T (10^1) +H (10^2) +H (10^3)*
A = 831 * 897 =207 +0(10) +172(100) +728(1000)
= 207 +17200 +728000 =*745407. So,* ***A = 745407***

B=275*512 using *Nearest Hundred's Method*

Hundred's=10^2	*X*	*Y*	*(X+Y)-100*
H_1	275	512	*687*
H_2=H_1-100	175	412	*487*
H_3=H_2-100	75	312	*287*
X*Y = +Ve, So to Add		***H =***	***+ 1461***
Ten's=10^1	*-X'*	*Y'*	*(-X'+Y')+10*
H_4=H_3-100	-25	212	*197*
T_1=H_4+10	-15	222	*217*
(-X')*Y' = -Ve, So to Subtract		***T =***	***-414***
One's=10^0	*-X''*	*Y''*	*(-X''+Y'')+1*
O_1=T_2+10	-5	232	*228*
O_2=O_1+1	-4	233	*230*
O_3=O_2+1	-3	234	*232*
O_4=O_3+1	-2	235	*234*
O_5=O_4+1	-1	236	*236*
(-X'')*Y'' = -Ve, So to Subtract		***H =***	***-1160***

Table: 8b – worked example: 275*512

The needed three values are H=1461, T= -414 and O=-1160
*X * Y =O (10^0) +T (10^1) +H (10^2)*
B =275 * 512 = (-1160) +(-414)(10) +1461(100)
= -1160 -4140 +146100
= 146100 -5300=*140800*. *So,* ***B = 140800***

C=469*436 using *Nearest Hundred's Method*

Hundred's=10^2	*X*	*Y*	*(X+Y)-100*
H_1	469	436	*805*
H_2=H_1-100	369	336	*605*
H_3=H_2-100	269	236	*405*
H_4=H_3-100	169	136	*205*
H_5=H_4-100	69	36	*5*
X*Y = +Ve, So to Add		***H =***	***+2025***
Ten's=10^1	*-X'*	*-Y'*	*(-X' –Y')+10*
T_1=H_5-100	-31	-64	*-85*
T_2=T_1+10	-21	-54	*-65*
T_3=T_2+10	-11	-44	*-45*
(-X')*(-Y')= +Ve, So to Add		***T =***	***+195***
One's=10^0	*-X''*	*-Y''*	*(-X'' –Y'')+1*
O_1=T_3+10	-1	-34	*-34*
(-X'')*(-Y'')= +Ve, So to Add		***O =***	***+34***

Table: 8c – worked example: 469*436

The 3 values H =2025, T =195 and O =34
*X * Y = O (10^0) +T (10^1) +H (10^2)*
C = 469 * 436 = 34 +195(10) +2025(100)
= 34 +1950 +202500 = *204484. So,* ***C = 204484***

We have found the first three values needed for the improvised 3 way Karatsuba algorithm namely A, B, and C. By substituting the above three values, we can find the next three values needed, namely D, E and F.

D = A +B – [(831 –275) (897 –512)]
= 745407 +140800 – [(556) (385)]

556 * 385 using *Nearest Hundred's Method*

Hundred's=10^2	*X*	*Y*	*(X+Y)-100*
H_1	556	385	*841*
H_2=H_1-100	456	285	*641*
H_3=H_2-100	356	185	*441*
H_4=H_3-100	256	85	*241*
(X)*(Y)= +Ve, So to Add		***H =***	***+ 2164***
Ten's=10^1	*X'*	*-Y'*	*(X' –Y')+10*
T_1=H_4-100	156	-15	*151*
(X')*(-Y')= -Ve, So to Subtract		***T =***	***-151***
One's=10^0	*X''*	*-Y''*	*(X'' –Y'')+1*
O_1=T_1+10	166	-5	*162*
O_2=O_1+1	167	-4	*164*
O_3=O_2+1	168	-3	*166*
O_4=O_3+1	169	-2	*168*
O_5=O_4+1	170	-1	*170*
(X'')*(-Y'')= -Ve, So to Subtract		**O =**	***-830***

Table: 8d – worked example: 556*385

The 3 values H =2164, T = -151 and O = -830
*X * Y =O (10^0) +T (10^1) +H (10^2)*
556 * 385 = (-830) +(-151) (10) +2164(100)
= -830 -1510 +216400
= 216400-2340=*214060*
D = 745407 +140800 –[(556) (385)]
= 745407 +140800 –214060
= 886207 –214060 = *672147*. *So,* ***D = 672147***

E = A +B +C – [(831 –469) (897 –436)]
E **=** 745407 +140800 +204484 –[(362) (461)]

362 * 461 using *Nearest Hundred's Method*

Hundred's=10^2	*X*	*Y*	*(X+Y)-100*
H_1	362	461	*723*
H_2=H_1-100	262	361	*523*
H_3=H_2-100	162	261	*323*
H_4=H_3-100	62	161	*123*
(X)*(Y)= +Ve, So to Add		***H =***	***+ 1692***
Ten's=10^1	*-X'*	*Y'*	*(-X' +Y')+10*
T_1=H_4-100	-38	61	*33*
T_2=T_1+10	-28	71	*53*
T_3=T_2+10	-18	81	*73*
T_4=T_3+10	-8	91	*93*
(X')*(-Y')= -Ve, So to Subtract		***T =***	***- 252***
One's=10^0	*X''*	*Y''*	*(X +Y)-1*
O_1=T_4+10	2	101	*102*
O_2=O_1-1	1	100	*100*
(X'')*(Y'')= +Ve, So to Add		***O =***	***+ 202***

Table: 8e – worked example: 362*461

The 3 values H =1692, T = -252 and O = -202
*X * Y = O (10^0) +T (10^1) +H (10^2)*
362 * 461 = -202 +(-252)(10) +1692(100)
=202 -2520 +169200
= 169402-2520=*166882*

E **=** 745407 +140800 +204484 - [(362) (461)]
= 745407 +140800 +204484 -166882
= 1090691 -166882 =*923809.So,* ***E = 923809***

F = B +C – [(275 –469) (512 –436)]
= 140800 +204484 – [(275 –469) (512 –436)]
= 140800 +204484 – [(275 –469) (512 –436)]
= 140800 +204484 – [(–194) (76)]

194 * 76 using *Nearest Hundred's Method*

Hundred's=10^2	*X*	*Y*	*(X+Y)-100*
H_1	194	76	170
(X)*(Y)= +Ve, So to Add		***H =***	***+ 170***
Ten's=10^1	*X'*	*-Y'*	*(X' –Y')+10*
T_1=H_4-100	94	-24	80
T_2=T_1+10	104	-14	100
(X')*(-Y')= -Ve, So to Subtract		***T =***	***-180***
One's=10^0	*X''*	*-Y''*	*(X'' –Y'')+1*
O_1=T_2+10	114	-4	111
O=O_1-1	115	-3	113
O_3 =O_2 -2	116	-2	115
O_4=O_3-3	117	-1	117
(X'')*(-Y'')= -Ve, So to Subtract		***O =***	***- 456***

Table: 8f – worked example: 194*76

The 3 values H =170, T = -180 and O = -456

*X * Y = O (10^0) +T (10^1) +H (10^2)*

194 * 76 = (-456) +(-180)(10) +170(100)

= -456 -1800 +17000

= 17000 -2256 = 14744

F = 140800 +204484 -(-14744)

= 345284 +14744 = 360028. *So,* **F = 360028**

Now we found all the 5 values needed.

C =204484, F =360028, E =923809, D =672147 & A =745407

Substituting the above five values in the expansion

G = C $(10^0)^3$ +F $(10^1)^3$ +E $(10^2)^3$ +D $(10^3)^3$ +A $(10^4)^3$

= 204484 +360028$(10)^3$ +923809$(10)^6$ +672147$(10)^9$ +745407$(10)^{12}$

```
                 204 484
             360 028 000
         923 809 000 000
     672 147 000 000 000
(+)  745 407 000 000 000 000
     746 080 071 169 232 484
```

*831,275,469*897,512,436* ***= 746, 080, 071, 169, 232, 484***

Here it is important to highlight that in the above 9 x 9 digit multiplication, if we use conventional long multiplication method, it requires 81 multiplication steps, whereas using the nearest place values algorithm and using the improvised 3x3 Karatsuba Algorithm presented in this paper, we can achieve the much desired O(n) in time complexity without using a single multiplication.

9. Egyptian, Babylonians and Average & Difference Square multiplication methods

The two ancient civilizations - Egyptian and Babylonians developed mathematics that was similar in some ways and different in others. In this section, we provide brief summary of Egyptian and Babylonians methods. Furthermore, we present an improved called - Average & Difference Square multiplication algorithms, based on modifications to Egyptian and Babylonian algorithms.

Ancient Egyptian Multiplication method

Ancient Egyptians had developed a very interesting method for multiplying two numbers, which is called doubling method. Egyptians are aware of the Binary numbers, also called 'base-2' numbers. They also understand that any decimal number numbers can be represented using the sum of Binary numerals.

Considering 2 x 2 digit multiplication 48 * 67

LHS	RHS
48	67
1	67
(1+1) = 2	(67+67) = 134
(2+2) = 4	(134+134) = 268
(4+4) = 8	(268+268) = 536
(8+8) = 16*	(536+536) = 1072
(16+16) = 32*	(1072+1072) = 2144

Table: 9 - Egyptian Doubling Method: 48*67

From the above *Table: 9*, the number 48 on LHS column can be written as sum of 32 &16. The corresponding rows value of 32 and 16 from the RHS column is 2144 and 1072. By adding the corresponding values of 16 & 32, one can find the product of 48*36, which is equal to 1072 + 2144 = 3216.

Babylonians Multiplication method
Babylonians used square number tables in multiplication of two numbers.
To multiply two numbers say 'a' and 'b' Babylonians used the following algorithm.

$$a * b = \frac{(a+b)^2 - a^2 - b^2}{2}$$

The improvised version of the Babylonians algorithm, which is much simplified version.

$$a * b = \frac{a^2 + b^2 - (a-b)^2}{2}$$

We use the difference square $(a - b)^2$ instead of sum square $(a + b)^2$ in the improvised Babylonians algorithm

Before the use of calculators, for multiplications, during 18th century, they have referred predefined square number tables to do the multiplications using Quarter Square method.

As an example, consider 2 x 2 digit multiplication 48 * 67
Using the improvised Babylonians algorithm, let a = 48 and b = 67

$$48*67 = \frac{48^2 + 67^2 - (48-67)^2}{2} = \mathbf{3216}$$

Average & Difference Square (Quarter Square) multiplication method.
By further simplifying the Babylonians algorithm, we can achieve the Quarter Square multiplication Algorithm or simplify the Average & Difference Square Multiplication.

Refining the Babylonians algorithm to achieve Average & Difference Square Multiplication.

$$a* b = \frac{(a+b)^2 - (a-b)^2}{4}$$

Considering the previous example of 2 x 2 digit multiplication 48 * 67
Using the Quarter Square algorithm, let a = 48 and b = 67

$$48 * 67 = \frac{(48+67)^2 - (48-67)^2}{4} = \mathbf{3216}$$

To multiply any two integers and to achieve the *O(n)* time complexity, the needed strategy is

- By observing the above three algorithms one can see that, we need binary multiplication followed by the Egyptians (doubling technique) to multiply any two integers.
- Through repeated addition and subtraction of the two integers *'a' & 'b'*, we can achieve the much needed doubling at every alternate steps.

In the next chapter, which discusses and simplifies the above two needed strategy to achieve the *O(n) time complexity.*

10. Addition & Subtraction *(AnS)* approach to achieve *O(n)*

Let *'a'* and *'b'* be two integers.
In this Section, we propose Addition and Subtraction (AnS) based approach, which uses only addition and subtraction for multiplication of two integer numbers. The proposed approach relies and borrows some techniques that were discussed in Section. 9, through repeated addition and subtraction of the two numbers *'a' & 'b'.*

Steps	*Repeated Addition*	$a*2^{n-1}$	#	*Repeated Subtraction*	$b*2^{n-1}$
1	***a***	$\mathbf{a*2^0 = a}$	**#**	***b***	$\mathbf{b*2^0 = b}$
2	*a +b*		#	*a -b*	
3	***(a + b) + (a – b) = 2a***	$\mathbf{a*2^1 = 2a}$	**#**	***(a + b) – (a – b) = 2b***	$\mathbf{b*2^1 = 2b}$
4	*2a + 2b*		#	*2a -2b*	
5	***(2a + 2b) + (2a – 2b) = 4a***	$\mathbf{a*2^2 = 4a}$	**#**	***(2a + 2b) – (2a + 2b) = 4b***	$\mathbf{b*2^2 = 4b}$
6	*4a +4b*		#	*4a -4b*	
7	***(4a + 4b) + (4a – 4b) = 8a***	$\mathbf{a*2^3 = 8a}$	**#**	***(4a + 4b) – (4a + 4b) = 8b***	$\mathbf{b*2^3 = 8b}$

Table: 10a - The 7 steps of AnS algorithm – (a*b)

Write down *'a' & 'b'*, add *'a' & 'b'* and subtract *'a' & 'b'*. Now you have a two new numbers. Apply the same rules add & subtract to the two new numbers. As you keep repeating the process till we get five more new values and then to stop.

By observation from *Table: 10a*, we find that through repeated addition and subtractions, of the two integers *'a' and 'b'*, doubles itself, at every alternate steps i.e. at 1, 3, 5 and 7. Every odd numbered steps yields binary series. 2^0, 2^1, 2^2, and 2^3 which is 1, 2, 4, and 8. It's a combination of the Egyptian doubling multiplication method and Average & Difference Square Multiplication.

We collectively employ and improvise the Egyptians doubling logic and Babylonians algorithms to multiply the two integers only using the two basic arithmetic operations addition, and subtraction. Through repeated additions and subtractions to double any given two numbers using *(a +b)* and *(a –b).* It is very important to highlight that, the number of repeated steps needed to multiply any order of integers is to be limited to maximum of 7 steps Using this property and divide and conquer algorithm, we can achieve the much desired in time *O(n)* as demonstrated below..

Product of 69 * 87 using the AnS algorithm in time *O(n)*
To find 69 & 87 using repeated addition and subtraction (a+b) and (a-b), considering *a* = 69 & *b* = 87

2^n	*LHS = (a + b)*		*RHS = (a - b)*
1 =2⁰	***69***	**&**	***87***
	(69+87) =156	&	(69-87) = -18
2 =2¹	***(156-18) =138***	**&**	***(156+18) =174***
	(138+174) =312	&	(138-174) =-36
4 =2²	***(312-36) =276***	**&**	***(312+36) =348***
	(276+348) =624	&	(276-348) =-72
8 =2³	***(624-72) =552***	**&**	***(624+72) =696***

Table: 10b – (69*87)

In the above example, 69 * 87, the smallest number is 69 and the number '9' is having the biggest face value. We know that the number 9 can be written has 8 +1 = 2^3 +2^0. To arrive the needed values, we have to follow the AnS repeated addition and subtraction to reach the level of 2^3.
The product 69 * 87 can be written has (60 +9) * 87 = (60 * 87) +(9 * 87)
= (6 * 87)(10^1) + (9 * 87)(10^0)
= (4 +2)(87)(10^1) +(8 +1)(87)(10^0)
= (2^2 +2^1)(87)(10^1) + (2^3 +2^0)(87)(10^0)

Substituting corresponding values of 69 under 87 from *Table: 10b*
69*87 = (348 +174)(10) +(696 +87)(10^0)
= 522(10) +783(1) = 5220 +783 = ***6003***

The other way also it works 69 *87
= 69 * (80 +7)
= (69 * (8))(10^1) +(69) * (4 +2 +1)(10^0)
= (69 * (2^3))(10^1) + (69) * (2^2 +2^1 +2^0)(10^0)

Substituting corresponding values 87 under 69
= 552(10) +(276 +138 +69)(1) = 5520 +483 = ***6003***

*Now let us consider a three digit multiplication using an example 839 * 694.*
Let a = 839 and b = 694, and the smallest number is 694.

In 694 the number '9' is having biggest face value 9 and can be written as 8 +1 = 2^3 +2^0.
So to reach the 2^3 level we have to follow the AnS algorithm to get the desired answer.

2^n	*LHS = (a +b)*		*RHS = (a-b)*
1 =2⁰	***839***	**&**	***694***
	(839 +694) =1533	&	(839 -694) =145
2 =2¹	***(1533 +145) = 1678***	**&**	***(1533 -145) =1388***
	(1678 +1388) =3066	&	(1388 -1678) =290
4 =2²	***(3066 +290) =3356***	**&**	***(3066 -290) =2776***
	(2776 +3356) =6132	&	(2776 -3356) =580
8 =2³	***(6132 +580) =6712***	**&**	***(6132 -580) =5552***

Table: 10c – (839*694)

*839 * 694* = 839 * (600 +90 +4)
= 839 * (6(10^2) + 9(10) +4)
= (839)(4 +2)(100) + (839)(8+1)(10) + (839)(4)
= (839)(2^2 +2^1)(100) +(839)(2^3 +2^1)(10) +(839)(2^2)
Substituting corresponding values of694 under 839 from table 10c
= (3356 +1678)(100) +(6712 +839)(10) +3356 =(5034)(100) +(7551)(10) +3356 = 503400 +75510 +3356 =*582266*
*= 839 * 7694* ***=582266***
We can get the same answer the other way also
*839 * 694*= (800 +30 +9) * 694
= (8(10^2) +3(10) +9 * (694
= (8)(694)(100) +(2 +1)(694)(10) +(8 +1)(694)
= (2^3)(694)(100) + (2^1 +2^0)(694)(10) +(2^3 +2^0)(694)
Substituting corresponding values of 839 under 694 from table 10c
= (5552)(100) + (1388 +694)(10) + (5552 +694) = (5552)(100) + (2082)(10) +6246 = 555200 +20820 +6246 = *582266*
*= 839 * 694* ***= 582266***

11. Integer multiplication to achieve in time *O(n)* – *AnS* and divide & conquer method

In this section, we propose a new method called – Addition and Subtraction *(AnS)* approach, which is based on divide and conquer method to break large integer numbers into smaller blocks/chunks and then apply the *AnS* method discussed in Section 10, on each individual chunks simultaneously.

To find 759726896599588732731867 * 990593834999174568981846. Which is a 24 digit x 24 digit multiplication.
Using the 3 x 3 improvised Karatsuba algorithm the, 24 digits has been split into 3 equal parts each part having 8 digits, so value of 'n' = 24/3 = 8. Let a=75972689, b=65995887, c=32731867 and d=99059383, e=49991745, f = 68981846
A = (a * d) = 75972689 * 99059383
B = (b * e) = 65995887 * 49991745
C = (c * f) = 32731867 * 68981846
D = *A +B – [(a –b) (d –e)]* = A +B – [(75972689 –65995887) (99059383 –49991745)] = A +B – [(9976802) (49067638)]
E = *A +B +C – [(a –c) (d –f)]* = A +B +C – [(75972689 –32731867) (99059383 –68981846)] = A +B +C – [(43240822) (30077537)]
F = *B +C – [(b –c) (e –f)]* = B +C – [(65995887 –32731867) (49991745 –68981846)] = B +C – [(33264020) (–18990101)]
$G = C\,(10^0)^8 + F\,(10^1)^8 + E\,(10^2)^8 + D\,(10^3)^8 + A\,(10^4)^8$

To find *A = 75972689 * 99059383*
In 75972689, the number 9 is having the largest face value and in 99059383, the number 9 is having the largest face value. Any one of the number having the face value number 9 can be opted, to have least number of steps in *AnS* algorithm, which is up to level 2^3, because 9 =(8+1).

2^n	LHS = (a +b)		RHS = (a -b)
1 =2^0	***75972689***	**&**	***99059383***
	(*75972689* + 99059383) = 175032072	&	(*75972689* – 99059383) = – 23086694
2 =2^1	***(175032072 – 23086694) = 151945378***	**&**	***(175032072 + 23086694) = 198118766***
	(151945378 + 198118766) = 350064144	&	(151945378 – 198118766) = – 46173388
4 =2^2	***(350064144 – 46173388) = 303890756***	**&**	***(350064144 + 46173388) = 396237532***
	(303890756 + 3962375320) = 4266266076	&	*(303890756 – 3962375320) = – 3658484564*
8 =2^3	***(4266266076 – 3658484564) = 607781512***	**&**	***(4266266076 + 3658484564) = 7924750640***

Table: 11a – (75972689*99059383)

*75972689 * 99059383 =*
Place value of 9 =(8+1) is = 10000000 =$10^7$and the corresponding value at LHS is (607781512 + 75972689)(10^7) =6837542010000000
Place value of 9 =(8+1) is = 1000000 =$10^6$and the corresponding value at LHS is (607781512 + 75972689)(10^6) =683754201000000
Place value of 0 is =100000 =10^5 and the corresponding value at LHS is (0)(10^5) =0
Place value of 5 =(4+1) is =10000 =10^4 and the corresponding value at LHS is (303890756 + 75972689)(10^4) =3798634450000
Place value of 9 =(8+1) is = 1000 =$10^3$and the corresponding value at LHS is (607781512 + 75972689)(10^3) =683754201000
Place value of 3 =(2+1) is =100 =10^2 and the corresponding value at LHS is (151945378 + 75972689)(10^2) =22791806700
Place value of 8 is =10 =10^1 and the corresponding value at LHS is (607781512) (10^1) =6077815120
Place value of 3 =(2+1) is =1 =10^0 and the corresponding value at LHS is (151945378 + 75972689)(10^0) =227918067

Adding the above eight values

6837542010000000	
683754201000000	
3798634450000	
683754201000	
22791806700	
6077815120	
227918067	(+)
7525807697190887	

*A = 75972689 * 99059383* **= *7,525,807,697,190,887***

To find B = 65995887 * 49991745
In 65995887, the number 9 is having the largest face value and in 49991745, the number 9 is having the largest face value. Any one of the number having the face value number 9 can be opted, to have least number of steps in *AnS* algorithm, which is up to level 2^3, because 9 =(8+1).

2^n	LHS = (a +b)		RHS = (a -b)
1 =2^0	***65995887***	**&**	***49991745***
	(*65995887* + 49991745) = 115987632	&	(*65995887* – 49991745) = 16004142
2 =2^1	***(115987632 + 16004142) = 131991774***	**&**	***(115987632 – 16004142) = 99983490***
	(131991774 + 99983490) = 231975264	&	(131991774 – 99983490) = 32008284
4 =2^2	***(231975264 + 32008284) = 263983548***	**&**	***(231975264 – 32008284) = 199966980***
	(263983548 + 199966980) = 463950528	&	*(263983548 – 199966980) = 64016568*
8 =2^3	***(463950528 + 64016568) = 527967096***	**&**	***(463950528 – 64016568) = 399933960***

Table: 11b – (65995887*49991745)

65995887 * 49991745
Place value of 4 =10000000 =$10^7$and the corresponding value at LHS is (263983548)(10^7) =2639835480000000
Place value of 9 =(8+1) is = 1000000 =$10^6$and the corresponding value at LHS is (527967096 + 65995887)(10^6) =593962983000000
Place value of 9 =(8+1) is = 100000 =$10^5$and the corresponding value at LHS is (527967096 + 65995887)(10^5) =59396298300000
Place value of 9 =(8+1) is = 100000 =$10^4$and the corresponding value at LHS is (527967096 + 65995887)(10^4) =5939629830000
Place value of 1 is =1000 =$10^3$and the corresponding value at LHS is (65995887)(10^3) =65995887000
Place value of 7 =(4+2+1) is =100 =10^2 and the corresponding value at LHS is (263983548 + 131991774 + 65995887)(10^2) =46197120900
Place value of 4 is =10 =10^1 and the corresponding value at LHS is (263983548)(10^1) =2639835480
Place value of 5 =(4+1) is =1 =10^0 and the corresponding value at LHS is (263983548 + 65995887)(10^0) =329979435

Adding the above eight values

```
2639835480000000
 593962983000000
  59396298300000
   5939629830000
     65995887000
     46197120900
      2639835480
       329979435   (+)
----------------
3299249553952815
```

B = 65995887 * 49991745***= 3,299,249,553,952,815***

To find C = 32731867 * 68981846
In 32731867, the number 8 is having the largest face value and in 68981846, the number 9 is having the largest face value. Any one of the number having the face value number 8 or 9 can be opted, to have least number of steps in *AnS* algorithm, which is up to level 2^3,because 9 =(8+1).

2^n	*LHS = (a +b)*		*RHS = (a -b)*
1 =2^0	***32731867***	**&**	***68981846***
	(*32731867* + 68981846) = 101713713	&	(*32731867* – 68981846) = – 36249979
2 =2^1	***(101713713 – 36249979) = 65463734***	**&**	***(101713713 + 36249979) = 137963692***
	(65463734 + 137963692) = 203427426	&	(65463734 – 137963692) = – 72499958
4 =2^2	***(203427426 – 72499958) = 130927468***	**&**	***(203427426 + 72499958) = 275927384***
	(130927468 + 275927384) = 406854852	&	*(130927468 – 275927384) = – 144999916*
8 =2^3	***(406854852 – 144999916) = 261854936***	**&**	***(406854852 + 144999916) = 551854768***

Table: 11c – (32731867*68981846)

32731867 * 68981846 =
Place value of 6 =(4+2) is = 10000000 =$10^7$and the corresponding value at LHS is (130927468 + 65463734) (10^7) = 1963912020000000
Place value of 8 is =1000000 =$10^6$and the corresponding value at LHS is (261854936)(10^6) =261854936000000
Place value of 9 =(8+1) is =100000 =$10^5$and the corresponding value at LHS is (261854936 + 32731867)(10^5) = 29458680300000
Place value of 8 is =100000 =$10^4$and the corresponding value at LHS is (261854936)(10^4) =2618549360000
Place value of 1 is =1000 =$10^3$and the corresponding value at LHS is (32731867)(10^3) =32731867000
Place value of 8 is =100 =10^2 and the corresponding value at LHS is (261854936)(10^2) =26185493600
Place value of 4 is =10 =10^1 and the corresponding value at LHS is (130927468)(10^1) =1309274680
Place value of 6 =(4+2) is =1 =10^0 and the corresponding value at LHS is (130927468 + 65463734)(10^0) =196391202
Adding the above eight values

```
1963912020000000
 261854936000000
  29458680300000
   2618549360000
     32731867000
     26185493600
      1309274680
       196391202   (+)
----------------
2257904608686482
```

C = 32731867 * 68981846 ***= 2,257,904,608,686,482***
D = A +B – [(9976802) (49067638)].

To find 9976802 49067638*

In 9976802, the number 9 is having the largest face value and in 49067638, the number 9 is having the largest face value. The number 9976802 is the smallest and having the face value number 9 is opted, to have least number of steps in *AnS* algorithm, which is up to level 2^3, because 9 =(8+1).

2^n	LHS = (a +b)		RHS = (a -b)
1 =2^0	***9976802***	**&**	***49067638***
	(*9976802* + 49067638) = 59044440	&	(*9976802* – 49067638) = – 39090836
2 =2^1	***(59044440 – 39090836) = 19953604***	**&**	***(59044440 + 39090836) = 98135276***
	(19953604 + 98135276) = 118088880	&	(19953604 – 98135276) = – 78181672
4 =2^2	***(118088880 – 78181672) = 39907208***	**&**	***(118088880 + 78181672) = 196270552***
	(39907208 + 196270552) = 236177760	&	*(39907208 – 196270552) = – 156363344*
8 =2^3	***(236177760 – 156363344) = 79814416***	**&**	***(236177760 + 156363344) = 392541104***

Table: 11d – (9976802*49067638)

*9976802 * 49067638*

Place value of 9 =(8+1) is =1000000 =10^6 and the corresponding value at RHS is (392541104 + 49067638)(10^6) =441608742000000

Place value of 9 =(8+1) is =100000 =$10^5$and the corresponding value at RHS is (392541104 + 49067638)(10^5) =44160874200000

Place value of 7 =(4+2+1) is =100000 =$10^4$and the corresponding value at RHS is (196270552 + 98135276 + 49067638)(10^4) =3434734660000

Place value of 6 =(4+2) is =1000 =$10^3$and the corresponding value at RHS is (196270552 + 98135276)(10^3) =294405828000

Place value of 8 =100 =10^2 and the corresponding value at RHS is (392541104)(10^2) =39254110400

Place value of 0 is =10 =10^1 and the corresponding value at RHS is (0)(10^1) =0

Place value of 2 is =1 =10^0 and the corresponding value at RHS is (98135276)(10^0) =98135276

Adding the above seven values

```
 441608742000000
  44160874200000
   3434734660000
    294405828000
     39254110400
        98135276   (+)
 489538108933676
```

9976802 * 49067638 **=**489538108933676

D = A +B – [(9976802) (49067638)]. Substituting the values of A & B

D = 7525807697190887 +3299249553952815 – 489538108933676 =***10,335,519,142,210,026***

E = A +B +C –[(43240822) (30077537)]

To find 43240822 * 30077537

In 43240822, the number 8 is having the largest face value and in 30077537, the number 7 is having the largest face value. The number 30077537 having the smallest face value number 7 is opted, to have least number of steps in *AnS* algorithm, which is up to level 2^2, because 7 =(4+2+1).

2^n	LHS = (a +b)		RHS = (a -b)
1 =2^0	***43240822***	**&**	***30077537***
	(*43240822* + 30077537) = 73318359	&	(*43240822* – 30077537) = 13163285
2 =2^1	***(73318359 + 13163285) = 86481644***	**&**	***(73318359 – 13163285) = 60155074***
	(86481644 + 60155074) = 146636718	&	(86481644 – 60155074) = 26326570
4 =2^2	***(146636718 + 26326570) = 172963288***	**&**	***(146636718 – 26326570) = 120310148***

Table: 11e – (43240822*30077537)

43240822* 30077537

Place value of 3 =(2+1) =10000000 =$10^7$and the corresponding value at LHS is (86481644 + 43240822)(10^7) =1297224660000000

Place value of 0 is =1000000 =$10^6$and the corresponding value at LHS is (0)(10^6) =0

Place value of 0 is =100000 =$10^5$and the corresponding value at LHS is (0)(10^5) =0

Place value of 7 =(4+2+1) is =10000 =$10^4$and the corresponding value at LHS is (172963288 + 86481644 + 43240822)(10^4) =3026857540000

Place value of 7 =(4+2+1) is =1000 =$10^3$and the corresponding value at LHS is (172963288 + 86481644 + 43240822)(10^3) =302685754000

Place value of 5 =(4+1) is =100 =10^2 and the corresponding value at LHS is (172963288 + 43240822)(10^2) =21620411000

Place value of 3 =(2+1) =10 =$10^1$and the corresponding value at LHS is (86481644 + 43240822)(10^1) =1297224660

Place value of 7 =(4+2+1) is =1 =$10^0$and the corresponding value at LHS is (172963288 + 86481644 + 43240822)(10^0) = 302685754

Adding the above eight values

1297224660000000
3026857540000
302685754000
21620411000
1297224660
302685754 (+)
1300577423615414

43240822* 30077537 **=**1300577423615414

E = A +B +C –[(43240822) (30077537)]
E = A +B + C –1300577423615414 substituting the values of A, B & C
E = 7525807697190887 +3299249553952815 +2257904608686482 –1300577423615414 =***11,782,384,436,214,770***

F = B +C –[(33264020) (–18990101)] = B +C + [(33264020) (18990101)].
*To find 33264020 * 18990101*
In *33264020,* the number 6 is having the largest face value and in *18990101,* the number 9 is having the largest face value. The number 33264020 having the smallest face value number 6 is opted, to have least number of steps in AnS algorithm, which is up to level 2^2 ,because 6 =(4+2).

2^n	LHS = (a +b)		RHS = (a -b)
1 =2^0	***33264020***	**&**	***18990101***
	(*33264020* + 18990101) = 52254121	&	(*33264020* – 18990101) = 14273919
2 =2^1	***(52254121 + 14273919) = 66528040***	**&**	***(52254121 – 14273919) = 37980202***
	(66528040 + 37980202) = 104508242	&	(66528040 – 37980202) = 28547838
4 =2^2	***(104508242 + 28547838) = 133056080***	**&**	***(104508242 – 28547838) = 75960404***

Table: 11f – (33264020*18990101)

*33264020 * 18990101*
Place value of 3 =(2+1) =10000000 =$10^7$and the corresponding value at RHS is (37980202 + 18990101)(10^7) =569703030000000.
Place value of 3 =(2+1) =10000000 =$10^6$and the corresponding value at RHS is (37980202 + 18990101)(10^6) =56970303000000.
Place value of 2 is =100000 =$10^5$and the corresponding value at RHS is (37980202)(10^5) =3798020200000
Place value of 6 =(4+2) is =10000 =$10^4$and the corresponding value at RHS is (75960404 + 37980202)(10^4) =1139406060000
Place value of 4 is =1000 =$10^3$and the corresponding value at RHS is (75960404)(10^3) =75960404000
Place value of 0 is =100 =10^2 and the corresponding value at RHS is (0)(10^2) = 0
Place value of 2 is 10 =$10^1$and the corresponding value at RHS is (37980202) (10^1) = 379802020
Place value of 0 is =1 =$10^0$and the corresponding value at RHS is (0)(10^0) = 0
Adding the above eight values

569703030000000
56970303000000
3798020200000
1139406060000
75960404000
379802020 (+)
631687099466020

*33264020 * 18990101***=** 631687099466020

F = B +C +[(33264020) (18990101)].
F = B +C +631687099466020 substituting the values of A, & B
F = 3299249553952815 +2257904608686482 +631687099466020 = ***6,188,841,262,105,317***

We found the 6 values needed using the *AnS and Karatsuba 3 way algorithm*
A = 7,525,807,697,190,887
B = 3,299,249,553,952,815
C = 2,257,904,608,686,482
D = 10,335,519,142,210,026
E = 11,782,384,436,214,770
F = 6,188,841,262,105,317

But we need A, C, D, E, and F values to arrive the answer. Substituting the five values in the below expansion and adding: (considering 'n' = 24/3 = 8)

G = C $(10^0)^8$ +F $(10^1)^8$ +E $(10^2)^8$ +D $(10^3)^8$ + A $(10^4)^{8.}$

2,257,904,608,686,482	
618,884,126,210,531,700,000 000	
117,823,844,362,147,700,000,000,000,000,000	
10,335,519,142,210,026,000,000,000,000,000,000,000,000	
752,580,769,719,088,700,000,000,000,000,000,000,000,000,000,000	(+)
752,580,780,054,607,960,033,870,981,031,828,468,436,308,686,482	

*Using AnS algorithm and the improvised 3 way split Karatsuba algorithm, we found the product of a randomly selected 24 x 24 digit multiplication 759726896599588732731867 * 990593834999174568981846*
= 752,580,780,054,607,960,033,870,981,031,828,468,436,308,686,482.

Generalized *AnS* Algorithm to achieve in time *O(n)* for very large integer multiplication

*To find the product of 'n' digit * 'm' digit multiplication and following AnS method to arrive the needed 7 values, by substituting the values of multiplicand as 'n' and multiplier as 'm'.*

2^n		LHS = (a +b)		RHS = (a -b)	
1 =2⁰	**a =**	***n***	*&*	***m***	**= b**
	(a+b) =	*(n +m)*	&	*(n -m)*	= (a-b)
2 =2¹	**(a+b)+(a-b)= 2a =**	***2n***	*&*	***2m***	**= 2b =(a+b)-(a-b)**
	(2a+2b) =	*(2n +2m)*	&	*(2n -2m)*	= (2a-2b)
4 =2²	**(2a+2b)+(2a-2b)= 4a =**	***4n***	*&*	***4m***	**= 4b =(2a+2b)-(2a-2b)**
	(4a+4b) =	*(4n +4m)*	&	*(4n -4m)*	= (4a-4b)
8 =2³	**(4a+4b)+(4a-4b)= 8a =**	***8n***	*&*	***8m***	**= 8b =(4a+4b)-(4a-4b)**

Table: 11g – Generalized AnS algorithm for n*m

Through the *AnS* method we can arrive (without a single multiplication) the needed 8 = (4+4) *primary values* namely: *n, 2n, 4n, 8n* and *m, 2m, 4m, 8m*.

With the help of the any one set of above *4 primary values* one can easily calculate the other *5 secondary values* needed namely: *3n =(n+2n), 5n =(n+4n), 6n =(2n+4n), 7n =(n+2n+4n) or (n+6n), 9n =(n+8n)* 'or' *3m =(m+2m), 5m =(m+4m), 6m =(2m+4m), 7m =(m+2m+4m) or (m+6m), 9m =(m+8m).*

By substituting the corresponding values of *'LHS'* to *'RHS'* or *'RHS'* to *'LHS'* and multiplying the individual face value with respect to its corresponding place value and adding all those final values together to get the desired final answer in time *O(n).*

The salient features of the proposed *AnS* based algorithm is summarized here for convenience:

- In the *AnS* method, we used only the two basic arithmetic operations namely additions and subtractions repeatedly to double the two integers values and considered the alternate steps results to carry out the multiplication.
- The upper limit to repeat the repeated addition and subtraction to find product of two large integers using *AnS* method is maximum of 7 steps (6 addition and 6 subtraction operations) to find the 8 primary values needed and 6 more additions needed to find the 5 secondary values.
- To find a product of any two large integers, we need 18 basic arithmetic operations a total of 12 additions and 6 subtractions only.
- The unique advantage of the above *AnS* algorithm is that, we can multiply any two numbers having any two different orders in time *O(n).*

12. Repeated Doubling Method *(RDM)* to achieve in time *O(n).*

An improvised version of *AnS* algorithm is called Repeated Doubling Method *(RDM)*.

By doubling any one side value of *RHS* or *LHS* is much sufficient enough to carry out the multiplication and to achieve the much desired time complexity of *O(n).*

By following Repeated Doubling Method *(RDM)*, which is an improvised approach of *AnS* method, we need only four primary values and five secondary values.

We can understand the *RDM* through a worked example.

Suppose we have to multiply two numbers say 69 * 87

To find the product of 69 & 87 using Repeated Doubling method considering *a* = 69 & *b* = 87

2^n	*LHS = a*		*RHS = b*
1 =2⁰	*69*	***&***	***b = 87***
2 =2¹		***&***	*2b = (b+b) =* ***(87+87) =174***
4 =2²		***&***	*4b = (2b+2b) =* ***(174+174) =348***
8 =2³		***&***	*8b = (4b+4b) =* ***(348+348) =696***

Table: 12a – (69*87) – Repeated Doubling Method

In the above example, 69 * 87, the smallest number is 69 and the number '9' is having the biggest face value. We know that the number 9 can be written has 8 +1 = 2^3 +2^0. To arrive the three more needed values, we have to follow the repeated doubling to reach the level of 2^3 = 8.

If we double any one side of the *LHS* or *RHS* is very much sufficient to find the product.

From the Table: 12a, we only doubled the *RHS* value and followed repeated doubling approach at each level until to arrive the value of 2^3 = 8.

By substituting the corresponding values of *'LHS'* to *'RHS'* and multiplying the individual face value with respect to its corresponding place value and adding all those final values together to get the desired final answer .

The product 69 * 87 can be written has
= (60 +9) * 87
= (60 * 87) + (9 * 87)
= (6 * 87) (10^1) + (9 * 87) (10^0)
= (4 +2) (87) (10^1) + (8 +1) (87) (10^0)
= (2^2 +2^1) (87) (10^1) + (2^3 +2^0) (87) (10^0)

Substituting corresponding values of 69 under 87 from *Table: 12a*
69*87 = (348 +174) (10) + (696 +87) (10^0)
= 522 (10) +783 (1) = 5220 +783 = ***6003***

Through the *RDM* approach maximum values needed is 9 (4 primary and 5 secondary values) to multiply any two integers.

The main advantage of *RDM* over *ANS* is that we have reduced the maximum number of operation from 18 (12 additions + 6 subtractions) to 9 additions only. Technically we have eliminated 9 (3 additions + 6 subtractions needed by the *AnS* algorithm) arithmetic operations and improvised the *RDM* algorithm.

The *RDM* approach can be extended to any range of two numbers to find their product.

The salient features of the proposed *RDM* based algorithm is summarized here for convenience:

- In the *RDM* approach, we used only one arithmetic operation namely addition, repeatedly to double the any one of the *(LHS or RHS)* integer value to carry out the multiplication.
- The upper limit to repeat the repeated doubling to find product of two large integers using *RDM* approach is maximum of 3 steps to find the 3 primary values needed namely 2^1 =2, 2^2 =4, and 2^3 =8. Using the 4 primary values, we can calculate the 5 secondary values needed namely 3, 5, 6, 7 and 9.
- To find a product of any two large integers, we need 9 basic arithmetic operations (3 additions for primary and 6 additions for secondary values).
- The unique advantage of the above *RDM* algorithm is that, we can multiply any two numbers having any two different range, in time *O(n)*.
- We can also implement the *RDM* algorithm along with divide and conquer method or directly without utilizing the divide and conquer method, using big integer concept to multiply any two integers (multiplicand and multiplier, having any range).
- *RDM* algorithm is much faster than the *proposed NPV* and *AnS algorithms.*
- *Repeated doubling* is also called multiplication.

Conclusion

This paper proposed three novel and unique methods to achieve the much desired Integer multiplication in time *O(n).* The first method called - *Nearest Place Values (NPV) method,* modifies and improves the Vedic multiplication technique *"All from 9 and the last from 10 (NND)".* Second method called - *Repeated Addition & Subtraction method (AnS),* borrows and improves ancient Babylonians multiplication algorithms & Egyptian multiplication of doubling technique. The third method is called - *Repeated Doubling Method (RDM),* which is an improvised version *of AnS algorithm.*

In all the three proposed methods only two basic arithmetic operations namely addition and subtraction has been used. We also disproved the prediction made by Schönhage-Strassen that the best possible result of asymptotic in time is *O(n log n).* The proposed methods indeed achieve *O(n)* time complexity, by borrowing and improving appropriate ancient techniques and combining them with improvised divide and conquer Karatsuba Algorithm.

Based on the proposed methods and observations, we can say that not only the "Repeated Addition is called Multiplication" also *"Systematic Addition and Subtraction (using NPV & AnS) and Repeated Doubling (using RDM) is also called multiplication".*

Furthermore, we also extended the proposed *RDM* technique for very large integer multiplication. The proposed repeated *RDM* approach, is the most efficient and achieves much desired *O(n)* complexity than the *Nearest Place Values (NPV) and Addition and Subtraction (AnS) method.*

References

1. *Integer multiplication in time O(n log n) David Harvey, Joris van der Hoeven https://hal.archives-ouvertes.fr/hal-02070778/document*

2. *Faster integer multiplication. Martin Fürer https://www.math.tamu.edu/~rojas/furerfastmult.pdf*

3. *Vedic mathematics -'vedic' or 'mathematics': a fuzzy & neutrosophic analysis, w. B. Vasantha kandasamy Florentin smarandache. https://arxiv.org/ftp/math/papers/0611/0611347.pdf*

4. *Vedic mathematics or sixteen simple mathematical formulae from the Vedas http://www.ms.uky.edu/~sohum/ma330/files/manuscripts/Tirthaji_S.B.K.,_Agarwala_V.S.-Vedic_mathematics_or_sixteen_simple_mathematical_formulae_from_the_Vedas-Orient_Book_Distributors_1981.pdf*

5. *Egyptian multiplication and some of its ramifications, M.H. van Emden https://arxiv.org/pdf/1901.10961.pdf*

6. *Ancient numerals and arithmetic, by paul j. Renne http://www.chatham.edu/pti/curriculum/units/2004/Renne.pdf*

7. *Calculation Across Cultures and History, by Carl R. Seaquist, Padmanabhan Seshaiyer, and Dianne Crowley http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.537.3224&rep=rep1&type=pdf*

8. *Study of Vedic Multiplier Algorithms using Nikhilam Method, by Prof S. B. Somani, Nikhil R. Mistri https://www.ijareeie.com/upload/2016/may/25_Study.pdf*

9. *An Efficient Multiplication Algorithm Using Nikhilam Method, by Shri Prakash Dwivedi https://arxiv.org/pdf/1307.2735.pdf*

10. *Maharishi's Vedic Mathematics in Elementary Education Developing All Knowingness to Improve Affect, Achievement, and Mental Computation. John M. Muehlman. https://www.miu.edu/pdf_msvs/v08/muehlman.pdf*

11. *Multiplication of Long Integers (Faster than Long Multiplication). Arno Eigenwillig und Kurt Mehlhorn https://people.mpi-inf.mpg.de/~mehlhorn/ftp/chapter2A-en.pdf*

12. *A. Karatsuba, Yu. Ofman, Multiplication of manydigital numbers by automatic computers, Dokl. Akad. Nauk SSSR, 1962, Volume 145, Number 2, 293–294. http://www.mathnet.ru/links/37aa58fa15f04a46fa174e9b6ec452fe/dan26729.pdf*

13. *Multiplication of Multidigit Numbers on Automata. https://www.researchgate.net/profile/Anatolii_Karatsuba/publication/234346907_Multiplication_of_Multidigit_Numbers_on_Automata/links/00b495357e64391356000000/Multiplication-of-Multidigit-Numbers-on-Automata.pdf*

14. *A. A. Karatsuba (1995). "The Complexity of Computations" (PDF). Proceedings of the Steklov Institute of Mathematics. 211: 169–183. Translation from Trudy Mat. Inst. Steklova, 211, 186–202 (1995). http://www.ccas.ru/personal/karatsuba/divcen.pdf*