Journal of Computer Science

Original Research Paper

The New Matrix Model of Computation Based Purely on Quite a New Concept of the Matrix Computations for Extremely Quick Web Pages Loading

Zulfia A. Chotchaeva

Moscow State University[*], Krasniy Kurgan 369387, Russia

Article history
Received: 12-06-2020
Revised: 03-10-2020
Accepted: 17-10-2020

Email: lesya.chotchaeva8@gmail.com

*The self-study

Abstract: We all dream of quick loading. Quicker than it is now. So that it loads immediately. What is needed for this? There are a lot of things to do. The most important things in it are computations. To speed up loading, we need to speed up computations. If we will find the way to multiply large numbers quicker than we have, the loading will be much quicker. How to do that? We need to multiply large numbers in time O(log n). How is that possible? A new model of computation may solve this problem. These are the algorithms that require considerably less amount of resources to perform them. The time complexity of the algorithm is the main (key) resource that we need to reduce to get the desired complexity. It seems incredible, but it is possible. We will get this through the sorting array. Best, worst, and average cases of a given algorithm could be considered for each particular input instance of the problem when analyzing algorithms. The worst-case complexity is the most used in algorithm analysis, it gives an upper bound on the resources required by the algorithm. Thus, the discovery of better algorithms brings the upper bound on the worst-case running time down. This paper presents the new matrix model of computation, which is based on the concept of the new matrix computations for advanced computing. The paper intends to prove the existence of better algorithms for any given input instance of the worst-case time complexity $M(n) = O(n^2)$ that take O(log n) and provide extremely quick web pages loading and create a new topic in complexity.

Keywords: Computational Complexity, Upper Bound, Web Load Time, Extremely Quick Web Loading, Large Numbers Multiplication

Introduction

Every time you engage in encrypted communication on the internet - for example, access your banking website or perform a web search – your device performs a head-spinning number of multiplications, involving numbers with hundreds or even thousands of digits. Very likely your device uses Karatsuba's trick for this arithmetic. This is all part of the amazing software ecosystem that keeps our web pages loading as snappily as possible (TCMGL, 2019).

> *"We have seen that sometimes computational problems have nonintuitive algorithms, which are quantifiably better (i.e., more efficient) than algorithms that were known for thousands of years. It would, therefore, be really interesting to prove for interesting computational tasks that the current algorithm is the best - in other words, no better algorithms exist. For instance, we could try to prove that the O(n log n)-step algorithm for multiplication can never be improved (thus implying that multiplication is inherently more difficult than addition, which does have an O(n)-step algorithm)."*
> *Sanjeev Arora and Boaz Barak, 2006.*

The web should be fast, Google says. A one-second delay in web page load time yields fewer page views, a decrease in customer satisfaction, and loss in conversions. The web users expect the web to load in two seconds, or less and will abandon a page that takes three or more seconds. A slow loading web is bad not only for the end-users but also for search engine, web pages are indexed and ranked based on the experience have they provided users. How to make web load quicker? Here are ways to keep webpages quick, it needs to delete a cache, to evaluate plugins, to reconsider the content delivery network, to review a web hosting package, to use adaptive images, to improve a webpage's design, to use the GZIP compression and AJAX requests, to use HTML, to minify CSS, JavaScript, and HTTP requests and other things that can make web slow. What makes web pages loading quicker? We need to speed up computations and for this, we need a faster way to multiply large numbers, as many computational problems may be reduced to integer multiplication. A new model of computation may solve this problem.

"In computer science and more specifically in computational complexity, a model of computation is a model that describes how an output of a mathematical function is computed given an input. A model describes how units of computations, memories, and communications are organized. The computational complexity of an algorithm can be measured given a model of computation (Wikipedia, 2020b)." "One major choice of the theory of computation which is currently taken for granted is the choice of a model of computation and corresponding complexity measures and classes (Arora and Barak, 2009)." "Using a model allows studying the performance of algorithms independently of the variations that are specific to particular implementations and specific technology (Wikipedia, 2020b)." "A central concept in algorithms is that of a reduction. A reduction is an algorithm that solves problem *A* by transforming any instance of *A* to an equivalent instance of a previously solved problem *B* (Du and Ko, 2000)." "The theory of computation is the branch that deals with how efficiently problems can be solved on a model of computation, using an algorithm… The field is divided into three major branches: Automata theory and languages, computability theory and computational complexity, which are linked by the question: "What are the fundamental capabilities and limitations of computers." (Wikipedia, 2020a)." The computational complexity or simply the complexity of an algorithm is the number of resources required for running it. "A complexity class is a set of functions that can be computed within a given resource (Goldreich, 2008)." As we know, the computational complexity of a problem is the minimum of the complexities of all possible algorithms for this problem, "Complexity theory considers not only whether a problem can be solved at all on a computer, but also how efficiently the problem can be solved, two major aspects are considered-time complexity and space complexity, which are respectively how many steps does it take to perform a computation and how much memory is required to perform that computation (Wikipedia, 2020a)." "The scheduling problem seems to be much harder than the sorting problem (Van Leeuwen and Leeuwen, 1990)." "To perform a rigorous study of computation, computer science work with a mathematical abstraction of computers called a model of computation. Models of computations can be classified into three categories: Sequential models, functional models and concurrent models (Wikipedia, 2020a)." The various branches of computer science and mathematics study and work with various models of computation (the graph theory, the group theory (matrix group), discrete mathematics, automata theory, cluster computing, grid computing, cloud computing, finite fields, parallel and distributed computing, the ring theory, the set theory and other fields). Different models of computation can do different tasks. In this study is presented a quite new model of computation. Therefore, the content of the paper is specific, including notions, notations, and presentation and the style of the paper does not correspond to the reference style, as the new requires a new style. This new matrix model of computation solves the problem of web load time that needs a faster way to multiply large numbers. When reading the paper, you need to drop template thinking to understand and assess how to create a new and how it works. Significance of this work and the main point – the major measure of computational complexity the time complexity has been reduced. The upper bound on the worst-case running time $M(n) = O(n^2)$ of an algorithm for any given input instance of a problem has been improved significantly. The purpose of this paper is to prove that the fast Fourier transform to compute integer products in time O(n log n log log n) and recently submitted O(n log n) are not optimal and perfect for quick web load, that there exist the short paths, the quicker and easier algorithms for this problem that takes O(log n) complexity and are not based on the FFT. The paper does not compare the methods and properties of the new matrix model of computation with the fast Fourier transform directly, since the concept of the new matrix model of computation is not based on the FFT and the paper does not provides multiple additional clarifications in lemmas and proofs, instead speaks an algorithms language. The fast Fourier transform is an algorithm that computes the DFT of a sequence, or its inverse (IDFT) and the new matrix model of computation uses completely other methods and means in working with a sequence and its inverse, factorization, recursion, approximation, sparse matrix and matrix transformations that allows reducing the complexity of computations with considerably lower resource usage. Thus, the new matrix model of computation represents the new type of matrix computations that will reduce web pages load time through speeding up computations. The paper offers a solution of the new computational tasks, including the problems in computational chemistry, computational biology and computational physics and allows to perform them rapidly, accurately and more easily (for instance, crystals, liquids, gases; atomic clusters, biomolecules, bio-nano systems; metallic, organic, inorganic and biological nanomaterials; atomic clusters and nanoparticles; classical molecular dynamics simulation: Nanoscale phase and structural transitions; physical and chemical phenomena with solids, liquids and gases; elastic and plastic deformations, nanoindentation, dislocations; collisional processes involving clusters, nanoparticles and biomolecules; relativistic molecular dynamics simulation: Nano-processes and technologies; photon emission by particles; random walk dynamics of atoms, molecules, clusters, particles; multiscale modelling: charged Simulation of chemical reactions in molecular systems; molecular structures; molecular dynamics; DNA computing; RNA computing; proteins and nucleic acids). The high performance of computations and the high computational efficiency and accuracy of the new matrix model of computation will make it an attractive method (the advanced computing approach) for extremely quick web load and large systems, like listed above.

Materials and Methods (Matrix Model Concept, Definition, Applications)

The growth of digital technologies has an exponential trend. As a consequence, the need for information security increases more than before and many modern scientific computations require the multiplication of large numbers. The large systems that need to multiply numbers in the range of several thousand digits may employ Karatsuba's multiplication algorithm and the fast Fourier transform, these are the most well-known algorithms for the multiplication of two large numbers. Karatsuba multiplication has a time complexity of $O\left(n^{\log_2 3}\right)$, making this algorithm significantly faster than long multiplication and the fast Fourier transform reduces the computational complexity to O(n log n log log n). Recently submitted an integer multiplication algorithm with a complexity of O(n log n) is not practically useful, as its advantages only appear when multiplying extremely large numbers, as the internet says.. Reducing the complexity of these multiplication algorithms to O[log n) will reduce the web pages loading time through speeding up computations and using fewer resources.

Matrix multiplication has been introduced for making easier and clarifying computations in linear algebra, historically. "A major application of matrices is to represent linear transformations, that are, generalizations of linear functions (Wikipedia, 2020c)." "Applications of matrices are found in most scientific fields… A major branch of numerical analysis is devoted to the development of efficient algorithms for matrix computations, a subject that is centuries old and is today an expanding area of research. Matrix decomposition methods simplify computations (Wikipedia, 2020c)." "Matrix computations are used in the treatment of nearly all large-scale models of computation (ScienceDirect, 2020)". "The proper organization of the matrix computations is becoming more and more important when high-speed computers with hierarchical memory are used in the treatment of large computational tasks (ScienceDirect, 2020)." The importance of the computational complexity of a matrix multiplication relies on the fact that many algorithmic problems may be solved using matrix computations.

Definition of the Matrix

Matrix elements multiplication. Fundamental definition of the new matrix 1. Let us consider the basic notions, properties, and applications of a reference matrix that are numerous.

Two numbers are given. Let the first be *L* and the second be *E* (denotes are taken conditionally). It is necessary to find the product of these two numbers (*L·E*) using the new matrix. We generate a matrix that creates an array of options of the new matrix where we have inserted the (*L·E*) option in an array to define the value of this target option and we will always consider this option as one of the array options, since all these options, working in one array, create the links we need that will sort the array and work to get the desired value faster and easier. Let us enter the corresponding variables from the initial pairwise: *I* to *L* and *T* to *E*. The values of *I* and *T* are 8 and 9, respectively. These values were selected to equal (10-*L*) and (10-*E*), that is (10-2) and (10-1), where 10 is the specific basis for the values of *L* and *E*. Here, the "sizes" of numbers *L* = 2 and *E* = 1 are equal, they are numbers of the same position (or rank, order, place), regardless of the rank to which these numbers belong it is right.

Given the matrix and the linear map of this matrix (1):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 8 & 9 \end{pmatrix} \to \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{1}$$

the new matrix that bears in itself steady properties and the linear map of the new matrix were formed, the elements of the new matrix are *L*, *E*, *I*, and *T* and the options of the new matrix are (*L·E*), (*I·E*), (*L·T*), and (*I·T*). Let *L* = 2, *E* = 1, *I* = 10-2 = 8, *T* = 10-1 = 9, the values of the new matrix elements *I* and *T* are the specific values, they were defined under the concept of the new matrix, where *I* = 10-*L* and *T* = 10-*E*, i.e., the *I* is the complement of *L* and the T is the complement of *E* and 10 is the specific basis to create the new matrix. The definition of the specific basis of the new matrix has been entered (injected) as a new notion. The specific basis of the matrix can be any of this infinite set 10, 100, 1000, 10000, 100000, 1000000, …, 10^n, where n is equal to the number of digits in *L*, these new matrices may have (or use) a restricted or unrestricted specific basis. More precisely, the size of the restricted specific basis is equal to the size of 10 *L*, i.e., the number of zeroes in the specific basis is the same as the number of digits in *L*, for example, if *L* = 3, the restricted sbasis = 10, if *L* = 17, the restricted sbasis = 100, if *L* = 334, the restricted sbasis = 1000, if *L* = 2458, the restricted sbasis = 10000, etc and if *L* = n_1, the restricted sbasis = 10^n. Here, L>E, *T*>*I*, *L*<*I*, *E*<*T*, *L* + *E*<*I* + *T*, *L*<sbasis:2, *I*>sbasis:2 and *L* + *I* = *E* + *T* = sbasis, the specific basis of the matrix is restricted that means that the sbasis is taken successively. The option (*L*·*E*) = (2·1), (*I*·*E*) = (8·1), (*L*·*T*) = (2·9), (*I*·*T*) = (8·9) and (*L*·*E*) + (*I*·*E*) + (*L*·*T*) + (*I*·*T*) = sbasis^2 = $(10^n)^2$. The specifics of symbolic notation of a matrix are new (e.g., the matrix elements are represented by capital letters). The size of the new matrix is defined. The linear map of the elements of the new matrix is new and specific too, we have a reference matrix with two rows and two columns. All rules of using the matrix have been not simply updated, these rules were completely transformed. The properties of the new matrix and their applications are new and numerous.

Let *L*-*E* = *T*-*I* = *J* = 1, *I*-*E* = *T*-*L* = *W* = 7, here, *L* + I = *E* + *T* = sbasis = 10, then Jsbasis = 10 and Wsbasis = 70, further:

1. (*L*·*E*) + Wsbasis = (*I*·*T*) = 72
2. (*I*·*E*) + Jsbasis = (*L*·*T*) = 18
3. (*L*·*E*) + (*I*·*E*) = Esbasis = 10
4. (*L*·*E*) + (*L*·*T*) = Lsbasis = 20
5. (*I*·*E*) + (*I*·*T*) = Isbasis = 80
6. (*L*·*T*) + (*I*·*T*) = Tsbasis = 90
7. (*L*·*T*)-(*I*·*E*) = Jsbasis = 18-8 = 10
8. (*I*·*T*)-(*L*·*E*) = Wsbasis = 72-2 = 70
9. (*L*·*E*) + (*I*·*T*) + (*L*·*T*) + (*I*·*E*) = $sbasis^2$
10. *W*-*J* = *I*-*L* = 6
11. *W* + *J* = *T*-*E* = 8

Matrix elements multiplication, or the matrix elements products of the new matrix model, is not an operation that produces a matrix from two matrices. The matrix elements' products of the new matrix model represent a new computational tool. Therefore, the definition of the matrix elements products for the new matrix model of computation requires multiplication of elements of the new matrix: (*L*·*E*), (*I*·*T*), (*I*·*E*), (*L*·*T*) and if one of these products is defined, the others are also defined, i.e., the matrix elements products are obtained by multiplying the elements of the new matrix, thus, computing matrix elements products is a central operation in all computational applications of the new matrix model of computation which provide the high performance (efficiency) and accuracy of computations and reduced significantly the computational complexity, since the total number of steps needed for multiplying has the new performing complexity.

Notice, Matrix elements addition and Matrix elements subtraction and their properties are embedded in the new matrix and take new running times that are less than O(n).

Matrix elements addition: *L* + *E* = sbasis-*W*, *I* + *T* = sbasis + *W*, *I* + *E* = sbasis-*J*, *L* + *T* = sbasis + *J*, *L* + *E* + *I* + *T* = 2sbasis, (I + *T*)-(*L* + *E*) = (*W* + *J*) + (*W*-*J*) = 2W, (*L* + *T*)-(*I* + *E*) = (*W* + *J*)-(*W*-*J*) = 2*J*, *L* + *E* + *I* + *T* = 2sbasis, at *E*<*L*<*I*<*T* and *L* + *I* = *E* + *T* = sbasis.

Matrix elements subtraction: *L*-*E* = *T*-*I* = *J*, *I*-*E* = *T*-*L* = *W*, *I*-*L* = *W*-*J*, *T*-*E* = *W* + *J*, at *E*<*L*<*I*<*T* and *L* + *I* = *E* + *T* = sbasis.

Lemma 1

Reference matrix with the unrestricted sbasis. It is sufficient to have the value of work of one of the options (*L*·*E*), (*L*·*T*), (*I*·*E*), or (*T*·*L*), so that other values define through subtraction and addition only. As a sbasis to an initial couple of elements, any sizes can be used.

Proof

Given the new matrix (2):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 9999999999998 & 9999999999999 \end{pmatrix}, \tag{2}$$

and the linear map of the new matrix:

$$\begin{pmatrix} L & E \\ I & T \end{pmatrix},$$

let us take the specific basis equal to 10000000000000, is taken randomly, *L* = 2, *E* = 1, *I* = 10000000000000-2 = 9999999999998,
T = 10000000000000-1 = 9999999999999, *J* = 1, *W* = 9999999999997, Jsbasis = 10000000000000, Wsbasis = 99999999999970000000000000, here, *L*<*I* and *E*<*T*, *E*<*L*<*I*<*T*, *L* + *I* = *E* + *T* = sbasis, *LE* + *IE* = Esbasis, *LE* + *LT* = Lsbasis, *IE* + *IT* = Isbasis, *LT* + *ET* = Tsbasis, *LE* + Wsbasis = *IT*, *IE* + Jsbasis = *LT*, *I*-*L* = *W*-*J*, *T*-*E* = *W* + *J*, *LE*<*IE*<*LT*<*IT*, then:

LE = 2·1 = 2,

IT = *LE* + Wsbasis = 2+99999999999970000000000000 = = 99999999999970000000000002,

IE = Esbasis-*LE* = 10000000000000-2 = 9999999999998,

LT = *IE* + Jsbasis = 9999999999998+10000000000000 = 19999999999998, etc., i.e., we have the sorted array that gives a short response time for a given piece of work, high throughput (rate of processing work), low utilization of computing resources, high availability of the computing system or application, fast (or highly compact) data compression and decompression, high bandwidth, short data transmission time. It follows from this that it is sufficient to have the value of work of one of the options LE, LT, IE, or IT so that other values define through subtraction and addition only that have a new and easy time complexity.

Lemma 2

Squaring of the new matrix. It is sufficient to have the value of the square of one of these elements *L*, *E*, *I*, or *T* so that other values of the squares of the elements define through this one value.

Proof

Given the new matrix (3):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 8 & 9 \end{pmatrix}, \tag{3}$$

and the linear map of the new matrix:

$$\begin{pmatrix} L & E \\ I & T \end{pmatrix},$$

where, $L = 2$, $E = 1$, $I = 10\text{-}2 = 8$, $T = 10\text{-}1 = 9$, sbasis = 10, $L + I = E + T$, $J = 1$, $W = 7$, Jsbasis = 10, Wsbasis = 70, $E<L<I<T$ and the specific properties (L^2-E^2) + (T^2-I^2) = (Wsbasis + Jsbasis)-(Wsbasis-Jsbasis), (I^2 + T^2)-(L^2 + E^2) = (Wsbasis-Jsbasis) + (Wsbasis+Jsbasis), (T^2 + L^2)-(I^2 + E^2) = (Wsbasis + Jsbasis)-(Wsbasis-Jsbasis), (T^2-L^2) + (I^2-E^2) = (Wsbasis + Jsbasis) + (Wsbasis-Jsbasis), I^2-L^2 = (Wsbasis-Jsbasis), T^2-E^2 = (Wsbasis + Jsbasis), let's define the values of the squares of the elements of the new matrix: L^2, E^2, I^2, T^2 using the properties of the squaring of the new matrix model of computation:

1. $L^2 = I^2$- (Wsbasis-Jsbasis) = 64-60 = 4
2. $I^2 = L^2$ + (Wsbasis-Jsbasis) = 4+60 = 64
3. $E^2 = T^2$ - (Wsbasis+Jsbasis) = 81-80 = 1
4. $T^2 = E^2$ + (Wsbasis+Jsbasis) = 1+80 = 81

Lemma 3

Fundamental definition of a reference matrix, 2. Let us divide all these options of pairs of numbers ($L{\cdot}E$), ($L{\cdot}T$), ($I{\cdot}E$), and ($T{\cdot}L$) into four main categories: (a), (A), (b), (B). Values of work of options of category (a)<(b) and (b)<(B), (B)<(A). It will be convenient to operate with the new matrix options and for perception.

Proof

Given the new matrix (4):

$$newMatrix = \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{4}$$

where, E<L<I<T, let ($L{\cdot}E$) = a, ($I{\cdot}T$) = A, ($I{\cdot}E$) = b, ($L{\cdot}T$) = B, $L + I = E + T$ = sbasis, then a + Wsbasis = A, b + Jsbasis = B, where $L + I = E + T$ = sbasis = 10^n and L-E = T-I = J, I-E = T-L = W, A>a, B>b, a<b<B<A.

It follows that:

1). A + a + B + b = $(10^n)^2$ = *sbasis*^{2} = a + (a + Wsbasis) + b + (b + Jsbasis)
2). a + b = Esbasis
3). A+ B = Tsbasis
4). A + b = Isbasis
5). a + B = Lsbasis
6). Aa = Bb
7). B/a = A/b
8). b/a = A/B
9). a/b = B/A

at $L>E$ and $I<T$, $T>I>L>E$, a<b<B<A.

Lemma 4

Proof of lemma 3. One of the ways of factorization of the new matrix that sidesteps the need for reference multiplication, it has polynomial running time, is presented as an example of operations of a new matrix only.

Proof

Given the new matrix (5):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 8 & 9 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{5}$$

where, $L = 2$, $E = 1$, $I = 10\text{-}2 = 8$, $T = 10\text{-}1 = 9$, sbasis = 10, $J = L\text{-}E = T\text{-}I = 1$, $W = I\text{-}E = T\text{-}L = 7$, Jsbasis = 10, Wsbasis = 70, $L + I = E + T$ = sbasis, ($L{\cdot}E$) = a = 2, ($I{\cdot}T$) = A = 72, ($I{\cdot}E$) = b = 8, ($L{\cdot}T$) = B = 18, a + A + b + B = *abasis*^{2} = $LE + IT + IE + LT$ = 100, a + Wsbasis = A, b + Jsbasis = B, Aa = Bb, a + b = Esbasis, a + B = Lsbasis, A + b = Isbasis, A + B = Tsbasis. A>B>b>a, $T>I>L>E$, let us define the value LE using these properties:

- $\begin{cases} Bb = Aa, \\ a + b = 10, \end{cases} \begin{cases} b(b + Jsbasis) = a(a + Wsbasis), \\ \qquad b = 10 - a, \end{cases}$
- $\begin{cases} b^2 - a^2 = 70a - 10b, \\ \quad b = 10 - a. \end{cases}$
- $(10-a)^2 - a^2$ = 70a-10(10-a)
- 100-20a + $a^2 - a^2$ = 70a-100+10a
- 100-20a = 80a-100

- 100a = 200
- a = 2
- A = a + Wsbasis = 2+70 = 72
- B = Tsbasis-A = 90-72 = 18
- b = Esbasis-a = 10-2 = 8

Lemma 5

Quick fractionation. This process demonstrates that all options, except options (1…·1…), cannot be studied at all because any option rather easily becomes a category option (1…·1…) – (a). For example, (2·1) will pass in (12·11). This means that we transferred (2·1) from the (a) category of units to the (a) category of tens.

Proof

Given the new matrix (6):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 8 & 9 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \qquad (6)$$

where, *L* = 2, *E* = 1, *I* = 10-2 = 8, *T* = 10-1 = 9, *J* = *L*-*E* = *T*-*I* = 2-1 = 9-8 = 1, *W* = *I*-*E* = *T*-*L* = 8-1 = 9-2 = 7, Jsbasis = 10, Wsbasis = 70, sbasis = 10 = *L* + *I* = *E* + *T*, *L*>*E*, *T*>*I*, *LE*- category (a) option, let us consider the set of the new matrix elements specific transformations:

Given the new matrix1 (6.1):

$$newMatrix1 = \begin{pmatrix} 12 & 11 \\ 88 & 89 \end{pmatrix} \rightarrow \begin{pmatrix} L1 & E1 \\ I1 & T1 \end{pmatrix}, \qquad (6.1)$$

where, *L*1 = 12, *E*1 = 11, *I*1 = 88, *T*1 = 89, sbasis1 = 100, i.e., the new matrix and the new matrix1 are linked directly since the values of the elements of these matrices are specific and were obtained with the purpose to define the most efficient approach (for maximum efficiency we need to minimize resource usage).

> *L*1 = *L* + sbasis = 2+10 = 12 and *E*1 = *E* + sbasis = 1+10 = 11, *I*1 = sbasis1-*L*1 = 100-12 = 88, *T*1 = sbasis1-*E*1 = 100-11 = 89, *J*1 = *L*1-*E*1 = *T*1-*I*1 = 12-11 = 89-88 = 1, *W*1 = *I*1-*E*1 = *T*1-*L*1 = 88-11 = 89-12 = 77, let us define the value *LE* - category (a) option using the value *L*1*E*1- category (a) option of the new matrix1:
>
> *LE* = *L*1*E*1-(2*sbasis*^{2}-Wsbasis) = 132-(200-70) = 132-130 = 2;
> given the new matrix2 (6.2):

$$newMatrix2 = \begin{pmatrix} 19 & 18 \\ 81 & 82 \end{pmatrix} \rightarrow \begin{pmatrix} L2 & E2 \\ I2 & T2 \end{pmatrix}, \qquad (6.2)$$

where, *L*2 = 19, *E*2 = 18, *I*2 = 81, *T*2 = 82, sbasis2 = 100, *L*2 = *T* + sbasis = 9+10 = 19, *E*2 = *I* + sbasis = 8+10 = 18, *I*2 = sbasis2-*L*2 = 100-19 = 81, *T*2 = sbasis2-*E*2 = 100-18 = 82, *J*2 = *L*2-*E*2 = *T*2-*I*2 = 19-18 = 82-81 = 1, *W*2 = *I*2-*E*2 = *T*2-*L*2 = 81-18 = 82-19 = 63, J2sbasis2 = 100, W2sbasis2 = 6300, *T*2>*I*2>*L*2>*E*2, let us define the value *IT*-category (A) option of the new matrix using the value *L*2*E*2 - category (a) option of the new matrix2:

> *IT* = *L*2*E*2-(2*sbasis*^{2} + Wsbasis) = 342-(200+70) = 342-270 = 72;
> given the new matrix3 (6.3):

$$newMatrix3 = \begin{pmatrix} 18 & 11 \\ 82 & 89 \end{pmatrix} \rightarrow \begin{pmatrix} L3 & E3 \\ I3 & T3 \end{pmatrix}, \qquad (6.3)$$

where, *L*3 = 18, *E*3 = 11, *I*3 = 82, *T*3 = 89, sbasis3 = 100, *L*3 = *I* + sbasis = 8+10 = 18, *E*3 = *E* + sbasis = 1+10 = 11, *I*3 = sbasis3-*L*3 = 100-18 = 82, *T*3 = sbasis3-*E*3 = 100-11 = 89, *J*3 = *L*3-*E*3 = *T*3-*I*3 = 18-11 = 89-82 = 7, *W*3 = *I*3-*E*3 = *T*3-*L*3 = 82-11 = 89-18 = 71, J3sbasis3 = 700, W3sbasis3 = 7100, *T*3>*I*3>*L*3>*E*3, let us define the value *IE*- category (b) option of the new matrix using the value *L*3*E*3 - category (a) option of the new matrix3:

> *IE* = *L*3*E*3-(2*sbasis*^{2}-Jsbasis) = 198-(200-10) = 198-190 = 8;
> Given the new matrix4 (6.4):

$$newMatrix4 = \begin{pmatrix} 19 & 12 \\ 81 & 88 \end{pmatrix} \rightarrow \begin{pmatrix} L4 & E4 \\ I4 & T4 \end{pmatrix}, \qquad (6.4)$$

where, *L*4 = 19, *E*4 = 12, *I*4 = 81, *T*4 = 88, sbasis4 = 100, *L*4 = *T* + sbasis = 9+10 = 19, *E*4 = *L* + sbasis = 2+10 = 12, *I*4 = sbasis4-*L*4 = 100-19 = 81, *T*4 = sbasis4-*E*4 = 100-12 = 88, *J*4 = *L*4-*E*4 = *T*4-*I*4 = 19-12 = 88-81 = 7, *W*4 = *I*4-*E*4 = *T*4-*L*4 = 81-12 = 88-19 = 69, J4sbasis4 = 700, W4sbasis4 = 6900, *T*4>*I*4>*L*4>*E*4, let us define the value *LT*-category (B) option of the new matrix using the value *L*4*E*4 - category (a) option of the new matrix4:

> *LT* = *L*4*E*4-(2*sbasis*^{2} + Jsbasis) = 228-(200+10) = 228-210 = 18,

i.e., the new matrix elements specific transformations are represented in the set of the new matrices 1, 2, 3, and 4 and the values of these new matrices elements products were obtained by expanding the new matrix elements transformations. It works on only 1% of all possible options, which are options of the least category to which certain options are carried, as it simple to define values of 99% of the other options.

Lemma 6

Recursion of the new matrix. One of the operations of the new matrix to define the value of *LE* = (a) through the recursion of the new matrix. This recursion has a polynomial running time.

Proof

Given the new matrix (7):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \to \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{7}$$

where, L = 2, E = 1, I = E = 1, T = L = 2, L + I = E + T = sbasis = 3, J = L-E = T-I = 2-1 = 1, W = I-E = T-L = 1-1 = 2-2 = 0, Jsbasis = 3·1 = 3, Wsbasis = 0·1 = 0, LE = a, *IT* = A, *IE* = b, *LT* = B, a + Wsbasis = A, b + Jsbasis = B, a + b = Esbasis = 1·3 = 3, a + B = Lsbasis = 2·3 = 6, A + b = Isbasis = 1·3 = 3, A + B = Tsbasis = 2·3 = 6, a + A + b + B = $sbasis^2$ = 3^2 = 9, here, L = T, I = E, a = A, therefore:

- $sbasis^2$-b-B = a + A
- 9-1^2-2^2 = 4 = a + A
- 4:2 = a = A = 2
- *LE* = ((L+E)^2-E^2-L^2):2 = (3^2-2^2-1^2):2 = 2 = a

It is sufficient to have the square of a given specific basis for each particular input instance of the new matrix and the values of the squares of elements L and E to define the value *LE*.

Lemma 7

A sequence of the new matrix. This specific process completely provides the quick fractionation of the new matrix model that allows defining the values of options *LE*, *LT*, *IE*, and *IT* at any point and any position of all these options in this sequence of the new matrix and its reverse. At each repeat iteration of the reverse of the sequence, we will divide by 2. It works like a looping, the complexity of a loop is considered as O(log n). The existence of such a new matrix model sequence allows reducing the volume of work that is necessary when searching for values of works for all possible options of categories (a), (b), (B), and (A) to a minimum of only 1% of these options.

Proof

Given the new matrix (8):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 8 & 9 \end{pmatrix} \to \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{8}$$

where, L = 2, E = 1, I = 10-2 = 8, T = 10-1 = 9, sbasis = 10, J = L-E = T-I = 2-1 = 9-8 = 1, W = I-E = T-L = 8-1 = 9-2 = 7, Jsbasis = 10, Wsbasis = 70, LE = a, *IT* = A, *IE* = b, *LT* = B, A>B>b>a, a + Wsbasis = A, b + Jsbasis = B, a + A + b + B = $sbasis^2$, L + I = E + T = sbasis (in the given data set, the ranks of the elements L and E are the same, this means that in this case we will not increase the one of these elements L or E on 10, 100,1000, etc. to balance L with E in ranks), let us create the sequence of the new matrix model, it is necessary to enter the new notions and therefore let (2·1) = a_1, (12·11) = a_2, (22·21) = a_3, (32·31) = a_4, etc., then:

- a_1 = a_2-(2$sbasis^2$-Wsbasis) = 132-(200-70) = 2
- a_2 = a_3-(4$sbasis^2$-Wsbasis) = 462-(400-70) = 132
- a_3 = a_4-(6$sbasis^2$-Wsbasis) = 992-(600-70) = 462, etc.

and let (8·1) = b_1, (18·11) = b_2, (28·21) = b_3, (38·31) = b_4, etc., then:

- b_1 = b_2-(2$sbasis^2$-Jsbasis) = 198-(200-10) = 8
- b_2 = b_3-(4$sbasis^2$-Jsbasis) = 588-(400-10) = 198
- b_3 = b_4-(6$sbasis^2$-Jsbasis) = 1178-(600-10) = 588, etc.

and let (8·9) = A_1, (18·19) = A_2, (28·29) = A_3, (38·39) = A_4, etc., then:

- A_1 = A_2-(2$sbasis^2$ + Wsbasis) = 342-(200+70) = 72
- A_2 = A_3-(4$sbasis^2$ + Wsbasis) = 812-(400+70) = 342
- A_3 = A_4-(6$sbasis^2$ + Wsbasis) = 1482-(600+70) = 812, etc.

and let (2·9) = B_1, (12·19) = B_2, (22·29) = B_3, (32·39) = B_4, etc., then:

- B_1 = B_2-(2$sbasis^2$ + Jsbasis) = 228-(200+10) = 18,
- B_2 = B_3-(4$sbasis^2$ + Jsbasis) = 638-(400+10) = 228,
- B_3 = B_4-(6$sbasis^2$ + Jsbasis) = 1248-(600+10) = 638, etc.

Note that, for example, (331·28) will pass in (1331·1280), in which case we will increase 28 on 10 before adding the sbasis equal 1000 to balance 28 with 331 in "sizes". This means that we transferred 28 from category ten to category hundred. After obtaining the result, we will simply divide what was received by 10, which will be the value of (321·28).

Lemma 8

Mixed matrix of positive and negative numbers 1. Let us take a pair of positive and negative numbers. All the properties of a reference matrix are saved. An example of using a matrix. This operation of the new matrix proves that we can define the product of (L·E), where one of these two numbers is a positive number and the other is a negative number, with the same efficiency as the product of two positive numbers.

Proof

Given the new matrix (9):

$$newMatrix = \begin{pmatrix} 1 & -2 \\ 9 & 12 \end{pmatrix} \to \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{9}$$

where, L = 1, E = -2, I = 10-1 = 9, T = 10-(-2) = 12, T>I>L>E, sbasis = 10, sbasis-L = I, sbasis-E = T, L + I =

$E + T$ = sbasis, $J = L-E = T-I$ = 1-(-2) = 12-9 = 3, W = 9-(-2) = 12-1 = 11, $I-L = W-J$, $T-E = W + J$, Jsbasis = 3·10 = 30, Wsbasis = 11·10 = 110, let us define the values LE = a, IT = A, IE = b and LT = B, a<b<B<A, a + A + b + B = $sbasis^2$ = 100, then:

- a = 1·(-2) = -2
- A = a + Wsbasis = -2+110 = 108
- b = Esbasis-a = -20-(-2) = -18
- B = b + Jsbasis = -18+30 = 12
- a + A + b + B = -2+108+(-18)+12 = 100 = sbsais^2

Lemma 9

Mixed matrix 2. Set of positive and negative numbers. It works like a Mixed matrix 1. Let us consider the option ($L·E$), where these numbers are negative. This operation of the new matrix proves that we can define the product of two negative numbers using the new matrix properties.

Proof

Given the new matrix (10):

$$newMatrix = \begin{pmatrix} -1 & -2 \\ 11 & 12 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{10}$$

where, L = -1, E = -2, I = 10-(-1) = 11, T = 10-(-2) = 12, $T>I>L>E$, sbasis = 10, I = sbasis-L, T = sbasis-E, $L + I = E + T$ = sbasis, $J = L-E = T-I$ = -1-(-2) = 12-11 = 1, $W = I-E = T-L$ = 11-(-2) = 12-(-1) = 13, Jsbasis = 1·10 = 10, Wsbasis = 13·10 = 130, let us define the values LE = a, IT = A, IE = b, LT = B, a + Wsbasis = A, b + Jsbasis = B, a + A + b + B = $sbasis^2$, a<b<B<A, then:

- a = -1·(-2) = 2
- A = a + Wsbasis = 2+130 = 132
- b = Esbasis-a = -20-2 = -22
- B = b + Jsbasis = -22+10 = -12
- a + A + b + B = 2+132+(-22)+(-12) = 100

Lemma 10

Matrix of non-integers. All rules the creation of a reference matrix were respected. Let us review an example of using. This operation of the new matrix proves that not only the products of integers but also the products of non-integers can be defined using the new matrix.

Proof

Given the new matrix (11):

$$newMatrix = \begin{pmatrix} 2{,}1 & 1{,}1 \\ 7{,}9 & 8{,}9 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{11}$$

where, L = 2,1, E = 1,1, I = 10-2, 1 = 7,9, T = 10-1,1 = 8,9, sbasis = 10, I = sbasis-L, T = sbasis-E, $T>I>L>E$, $J = L-E = T-I$ = 1, $W = I-E = T-L$ = 6,8, Jsbasis = 10, Wsbasis = 68, let us define the values LE = a, IT = A, IE = b, LT = B, a<b<B<A, a + A + b + B = $sbasis^2$, then:

- a = 2,1·1,1 = 2,31
- A = a + Wsbasis = 2,31+68 = 70,31
- b = Esbasis-a = 11-2,31 = 8,69
- B = b + Jsbasis = 8,69+10 = 18,69
- a + A + b + B = 2,31+70,31+8,69+18,69 = 100

Lemma 11

Matrix with the restricted sbasis. Clearly, a reference matrix, where the sbasis of a matrix is not taken randomly. Let us define the value of ($L·E$) using this matrix. All properties are the same. Irrespective of the specific basis, more exactly is the sbasis restricted or unrestricted, we can define the value of ($L·E$) using the new matrix properties and operations.

Proof

Given the new matrix (12):

$$newMatrix = \begin{pmatrix} 11 & 2 \\ 89 & 98 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{12}$$

where L = 11, E = 2, I = 100-11 = 89, T = 100-2 = 98, sbasis = 100, L + I = $E + T$ = sbasis, $T>I>L>E$, J = $L-E = T-I$ = 11-2 = 98-89 = 9, W = $I-E = T-L$ = 89-2 = 98-11 = 87, Jsbasis = 9·100 = 900, Wsbasis = 87·100 = 8700, let us define the values LE = a, IT = A, IE = b, LT = B, a + A + b + B = $sbasis^2$, a<b<B<A, then:

- a = 11·2 = 22
- A = a + Wsbasis = 22+8700 = 8722
- b = Esbasis-a = 200-22 = 178
- B = b + Jsbasis = 178+900 = 1078
- a + A + b + B = 22+8722+178+1078 = 10000 = sbasis^2

Lemma 12

Matrix with the separate sbasis for L and E. Let us consider matrices where we will take the sbasis for L and E separately. Here, we will take 100 for E and 10 for L. The separate sbasis is not optimal to define the value of LE, there are additional steps in this operation that are not useful, therefore, using the separate specific basis for the new matrix is not perfect as an optimal.

Proof

Given the new matrix (13):

$$newMatrix = \begin{pmatrix} 2 & 11 \\ 8 & 89 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{13}$$

where, $L = 2$, $E = 11$, $I = 10\text{-}2 = 8$, $T = 100\text{-}11 = 89$, $sbasis_1 = L + I = 10$, $sbasis_2 = E + T = 100$, $L + I \neq E + T$ and here, Jsbasis = (Tsbasis$_1$-Esbasis$_1$):2-(Isbasis$_2$-Lsbasis$_2$):2 = 90, Wsbasis = (Isbasis$_2$-Lsbasis$_2$):2 + (Tsbasis$_1$-Esbasis$_1$):2 = 690, a + Wsbasis = A, b + Jsbasis = B, let us define the values LE = a, IT = A, IE = b, LT = B at a<A, b<B, a + A + b + B = sbasis$_1$·sbasis$_2$ = 10·100 = 1000, then:

- a = 2·11 = 22
- b = Esbasis$_1$-a = 110-22 = 88
- A = Isbasis$_2$-b = 800-88 = 712
- a + A + b + B = 22+712+88+178 = 1000
- a + Wsbasis = A = 22+690 = 712
- b + Jsbasis = B = 88+90 = 178

Lemma 13

Options repositions. The position of the option of the new matrix is irrelevant. We can leave a matrix in its initial form or rearrange the new matrix elements to define the value of ($L·E$) option, regardless of the option position, the matrix properties are keeping and the efficiency of operations does not decrease.

Proof

Given the new matrix (14):

$$newMatrix = \begin{pmatrix} 6 & 9 \\ 4 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{14}$$

where, $L = 6$, $E = 9$, $I = 10\text{-}6 = 4$, $T = 10\text{-}9 = 1$, here $T<I<L<E$, therefore, $J = E\text{-}L = I\text{-}T$ and $W = E\text{-}I = L\text{-}T$, a>A, b>B, A = a-Wsbasis, B = b-Jsbasis, the matrix has a non-working appearance and has to be turned, we can leave a matrix or rearrange the new matrix elements so that $T>I>L>E$, where a<A, b<B:

$$newMatrix = \begin{pmatrix} 4 & 1 \\ 6 & 9 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix},$$

we received the worker matrix, where a<A, b<B, a<b<B<A and A = a + Wsbasis, B = b + Jsbasis, i.e., (6·9) is the (A) option of the worker matrix and the (a) option of the given matrix and (4·1) is the (a) option of the worker matrix and the (A) option of the given matrix. If we do not rearrange places, the (6·9) option remain in a position of the (a) option on an arrangement in a matrix that will overturn all of the functions and properties of a reference matrix, where (a)<(b)<(B)<(A). Therefore, it will be more convenient if we simply rearrange a matrix, having brought it into a working look, where $E<L<I<T$, because potentially (4·9) is the option of a category (B), remaining on a position of a category (b) on an arrangement in a matrix it forces down the algorithm for a matrix, where $E<L<I<T$. If we leave a matrix and do not rearrange, the matrix will work in a new way, where (A)<(B)<(b)<(a) and this is true both at $T<I<L<E$ and under other conditions, a position of the option is irrelevant.

Lemma 14

Equations in systems. Set 1. We define the value LE through (x) and (y) using the properties of the new matrix. This operation of the new matrix not only simplifies the multiplication process but practically abolishes it, has polynomial time complexity.

Proof

Given the new matrix (15):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 8 & 9 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \tag{15}$$

where, $L = 2$, $E = 1$, $I = 10\text{-}2 = 8$, $T = 10\text{-}1 = 9$, sbasis = 10, $L + I = E + T$ = sbasis, I = sbasis-L, T = sbasis-E, $J = L\text{-}E = T\text{-}I = 2\text{-}1 = 9\text{-}8 = 1$, $W = I\text{-}E = T\text{-}L = 8\text{-}1 = 9\text{-}2 = 7$, Jsbasis = 10, Wsbasis = 70, LE = a, IT = A, IE = b, LT = B, a<b<B<A, let Esbasis:2-x = a, Esbasis:2 + x = b, Tsbasis:2-y = B, Tsbasis:2+y = A and Lsbasis:2-x$_1$ = a, Lsbasis:2 + x$_1$ = B, Isbasis:2-y$_1$ = b, Isbasis:2 + y$_1$ = A, where x + y = (Wsbasis-Jsbasis):2 = 30, y-x = (W^2-J^2):2 = 24, x$_1$ + y$_1$ = (Wsbasis + Jsbasis):2 = 40, y$_1$-x$_1$ = y-x = 24, let us define the value LE = a using the following:

a} $\begin{cases} x + y = 30, \\ y - x = 24, \end{cases} \begin{cases} y = 30 - x, \\ y - x = 24. \end{cases}$

30-x-x = 24,
2x = 6,
x = 3.
LE = Esbasis:2-x = 10:2-3 = 2.

b} $\begin{cases} x_1 + y_1 = 40, \\ y_1 - x_1 = 24, \end{cases} \begin{cases} x_1 = 40 - y_1, \\ y_1 - x_1 = 24. \end{cases}$

y$_1$-(40-y$_1$) = 24,
2y$_1$ = 24+40,
y$_1$ = 64:2,
y$_1$ = 32.
LE = Lsbasis:2-x$_1$ = 20:2-8 = 2

Lemma 15

Set of equations in systems 2. To define the value LE through (x_2) and (y_2). This operation of the new matrix eliminates the need for multiplication too, has polynomial running time, is presented as an example of operations only.

Proof

Given the new matrix (16):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 8 & 9 \end{pmatrix} \to \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \quad (16)$$

where, $L = 2$, $E = 1$, $I = 10\text{-}2 = 8$, $T = 10\text{-}1 = 9$, sbasis = 10, $L+I = E + T$ = sbasis, I = sbasis-L, T = sbasis-E, $J = L\text{-}E = T\text{-}I = 2\text{-}1 = 9\text{-}8 = 1$, $W = I\text{-}E = T\text{-}L = 8\text{-}1 = 9\text{-}2 = 7$, Jsbasis = 10, Wsbasis = 70, $I\text{-}L = W\text{-}J$, $T\text{-}E = W + J$, LE = a, IT = A, IE = b, LT = B, a + Wsbasis = A, b + Jsbasis = B, a + A + b + B = sbasis*s*2, a<b<B<A, let us define the value LE = a using the properties of the new matrix a + b = Esbasis = 10, a + B = Lsbasis = 20, A + b = Isbasis = 80, A + B = Tsbasis = 90, let B-a = x_2, A-b = y_2 and x_2+y_2 = Wsbasis + Jsbasis = 80, y_2-x_2 = W^2-J^2 = 7^2-1^2 = 48, we received the following:

$$\begin{cases} x_2 + y_2 = Wsbasis + Jsbasis, \\ \quad y_2 - x_2 = W^2 - J^2 \end{cases} \begin{cases} x_2 + y_2 = 80, \\ y_2 - x_2 = 48, \end{cases} \begin{cases} x_2 + y_2 = 80, \\ y_2 = 48 + x_2. \end{cases}$$

$$x_2 + 48 + x_2 = 80$$

$$2x_2 = 80 - 48,$$

$$x_2 = 16;$$

further:

$$\begin{cases} B - a = x_2, \\ a + B = 20, \end{cases} \begin{cases} B - a = 16, \\ B = 20 - a. \end{cases}$$

$$20 - a - a = 16,$$

$$2a = 4,$$

$$a = 2$$

$$LE = a = 2$$

Lemma 16

Applications of properties of the new matrix in equations. Let us define the value LE through (x), (y), (x_1), and (y_1). This operation of the new matrix replaces the process of multiplication with a new path, it takes polynomial running time, is presented as an example of operations and properties of a matrix.

Proof

Given a new matrix (17):

$$newMatrix = \begin{pmatrix} 3 & 2 \\ 7 & 8 \end{pmatrix} \to \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \quad (17)$$

where, $L = 3$, $E = 2$, $I = 10\text{-}3 = 7$, $T = 10\text{-}2 = 8$, $J = L\text{-}E = T\text{-}I = 1$, $W = T\text{-}L = I\text{-}E = 5$, $L+I = E+T$ = sbasis = 10, Wsbasis = 5·10 = 50, Jsbasis = 1·10 = 10, let LE = a, IT = A, IE = b and TL = B, (a + b) = Esbasis, (a + B) = Lsbasis, (A + b) = Isbasis, (A + B) = Tsbasis, (Wsbasis-Jsbasis):2 = 20 = x + y, (Wsbasis+Jsbasis):2 = 30 = x_1 + y_1, x_1-x = y_1-y, x_1 + y = y_1 + x, y : x = T:E, y_1 : x_1 = I:L and Esbasis:2-x = a, Lsbasis:2-x_1 = a, Isbasis:2-y_1 = b, Tsbasis:2-y = B, Esbasis:2 + x = b, Lsbasis:2 + x_1 = B, Isbasis:2 + y_1 = A, Tsbasis:2 + y = A, a + A + b + B = sbasis^2, a<b<B<A, A:b = B:a, b:A = a:B, a:b = B:A, Aa = Bb, let us define the values of x, y, x_1, and y_1 using these properties of the new matrix:

a) x(A + B):2 = y(a + b):2,
x(80:2) = y(20:2),
x = y(20:2):(80:2),
x = 10y:40,
x = y:4;
further,
x + y = 20,
x = 20-y;
and
20-y = y:4,
-y = y:4-20,
y = (-y +80):4,
5y = 80,
y = 80:5,
y = 16;
x = 20-y,
x = 20-16,
x = 4

b) y_1(a + B):2 = x_1(A + b):2,
y_1(30:2) = x_1(70:2),
15y_1 = 35x_1,
y_1 = 35x_1:15,
y_1 = 7x_1:3;
further,
x_1 + y_1 = 30,
y_1 = 30-x_1;
and,
30-x_1 = 7x_1:3,
3(30-x_1) = 7x_1,
90-3x_1 = 7x_1,
10x_1 = 90,
x_1 = 90:10,
x_1 = 9;
y_1 = 30-x_1,
y_1 = 30-9,
y_1 = 21.

and further:

- Esbasis:2-x = a = LE = 20:2-4 = 10-4 = 6
- Tsbasis:2+y = A = IT = 80:2+16 = 40+16 = 56
- Lsbasis:2+x_1 = B = LT = 30:2+9 = 15+9 = 24
- Isbasis:2-y_1 = b = IE = 70:2-21 = 35-21 = 14, etc. (see the properties of x, y, x_1, and y_1 above)

Lemma 17

Defining the value LE through the decomposition of the new matrix. It is sufficient to divide. This operation of the new matrix sidesteps (replaces, eliminates) the need for multiplication, has a logarithmic time complexity O(log n) since at each step in an algorithm we remove half of the remaining options, hence it cuts the work in half at each step. As a specific basis of a reference matrix is always equal to 10^n, where $E<L<I<T$, $L + I = E + T$ = sbasis = 10^n and a<b<B<A, the specific basis of a matrix is taken successively, when n in 10^n is equal to the number of digits in the L, thus, the sbasis of this matrix is restricted, irrespective of the restricted sbasis or unrestricted, the complexity of 10^n is not considered as a complexity, we can multiply and divide by constant factors (the constant factors are always dropped from big O analysis, we can ignore them in the analysis of algorithm).

Proof

Given the new matrix (18):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 8 & 9 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \qquad (18)$$

where, $L = 2$, $E = 1$, $I = 10\text{-}2 = 8$, $T = 10\text{-}1 = 9$, $J = L\text{-}E = T\text{-}I = 1$, $W = T\text{-}L = I\text{-}E = 7$, $L{+}I = E{+}T$ = sbasis = 10, Esbasis = 10, Lsbasis = 20, Isbasis = 80, Tsbasis = 90, Jsbasis = 10, Wsbasis = 70, $I\text{-}L = W\text{-}J$, $T\text{-}E = W{+}J$, let LE = a, IT = A, IE = b, LT = B, a + A + b + B = sbasis^2, Esbasis:2-x = a, x + y = (Wsbasis-Jsbasis):2 = 30, let us define the value LE = a using these properties of a new matrix that are presented in the following algorithm, then:

> LE = Esbasis:2-(Esbasis:2:(sbasis^2:(Isbasis-Lsbasis))) = 10:2-(10:2:(100:(80-20))) = 5-(5:(100:60)) = 2 = LE = a

or we can define the value of LE using the simplified version of the above algorithm where we are inversing the exponent of a constant that is sbasis^2 = $(10^n)^2$:

> LE = (E:2-(E:2:(sbasis:(I-L))))·sbasis = (1:2-(1:2:(10:(8-2))))·10 = (0,5-(0,5:(10:6))·10 = 0,2·10 = 2 = LE = a

Keep in mind that we are not raising 10 to the n-th power every time to define the sbasis, we need only to add the L and I elements of a matrix that will be the value of 10^n.

Lemma 18

Sparse matrix. We will consider the option where, for definition LE, we will take a sbasis equal to L. This operation is presented as an example of a possible factorization path only. It works like a recursion matrix and has polynomial time.

Proof

Given the new matrix (19):

$$newMatrix = \begin{pmatrix} 3 & 1 \\ 0 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \qquad (19)$$

where, $L = 3$, $E = 1$, $I = 3\text{-}3 = 0$, $T = 3\text{-}1 = 2$, $L + I = E + T$, here, sbasis = $L + I = E + T = 3$, $J = L\text{-}E = T\text{-}I = 3\text{-}1 = 2\text{-}0 = 2$, $W = E\text{-}I = L\text{-}T = 1\text{-}0 = 3\text{-}2 = 1$, LE = a, IT = A, IE = b, LT = B, a + A + b + B = sbasis^2 = 3^2 = 9, a + b = Esbasis, A + b = Isbasis, a + B = Lsbasis, A + B = Tsbasis, $J = T$, $W = E$, L = sbasis, $E<L<10^n{:}2$, let us define the value LE using the following:

- a + A + b + B = sbasis^2 = 9
- A = 0
- b = 0
- a + B = sbasis^2-A-b = 9-0-0 = 9

then:

- a = sbasis^2-J(W + J) = 3^2-2(1+2) = 9-6 = 3 = LT
- B = sbasis^2-a = 3^2-3 = 6 = LT 27

It is necessary to have the values of the squares of L and T and the product of $JW = ET$ to define the value of LE and we have (or get) a reference multiplication.

Lemma 19

Approximation of the new matrix and a sorted array of options of a matrix. Let us take a reference matrix with a restricted sbasis, which is taken successively. This operation of the new matrix takes a logarithmic running time and will provide a short path to O(log n).

Proof

Given the new matrix (20):

$$newMatrix = \begin{pmatrix} 2 & 1 \\ 8 & 9 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix}, \qquad (20)$$

where, $L = 2$, $E = 1$, $I = 10\text{-}2 = 8$, $T = 10\text{-}1 = 9$, $L + I = E = T$ = sbasis = 10, $J = L\text{-}E = T\text{-}I = 1$, $W = I\text{-}E = T\text{-}L = 7$, $W\text{-}J = I\text{-}L$, $W + J = T\text{-}E$, Jsbasis = 1·10 = 10, Wsbasis = 7·10 = 70, LE = a, IE = b, LT = B, IT = A, $E<L<I<T$, a + A + b + B = sbasis^2 = 10^2, a<b<B<A, let us define the values of options (a), (b), (B), and (A) using the approximate values of these options that will sort the array of options of a new matrix (it is like to arrange a quick indices, or the bounds of limits, or the reference points), we will denote them as a≈, b≈, B≈, A≈, then:

- a≈ = Esbasis:4 = 10:4 = 2,5
- b≈ = 3Esbasis:4 = 30:4 = 7,5
- A≈ = 3Tsbasis:4 = 270:4 = 67,5
- B≈ = Tsbasis:4 = 90:4 = 22,5

and further, let us define the values of (a), (b), (A), (B) using these approximate values of these options and the properties of the new matrix:

- a = a≈ + x≈ = 2,5 + (-0,5) = 2
- b = b≈ - x≈ = 7,5 – (-0,5) = 8
- A = A≈ - y≈ = 67,5 – (-4,5) = 72
- B = B≈ + y≈ = 22,5 + (-4,5) = 18

where:

> (x≈ + y≈) = (sbasis^2: 2-(Isbasis – Lsbasis)): 2 = (100: 2-(80-20)): 2 = (50 - 60): 2 = - 5

and further we are targeting half of half to find x≈:

> x≈ = Esbasis: 2: (sbasis^2: 2: ((sbasis^2: 2-(Isbasis-Lsbasis)): 2) = 10: 2: (100: 2: (100: 2-(80-20): 2) = 5: (50: ((50-60): 2) = -0,5

$y\approx = -5-(-0,5) = -4,5$

and:

- $A\approx - a\approx = Wsbasis\approx$
- $B\approx - b\approx = Jsbasis\approx$
- $a\approx + A\approx + b\approx + B\approx = sbasis^2$

We received a sorted array of options of the new matrix therefore we don’t need to check all possible options of an array to define the target value *LE*.

Lemma 20

Big O notation of the new matrix. Analysis of algorithms of the new matrix. Asymptotic analysis of the runtime of an algorithm. A new definition of an upper bound on the worst-case running time of M{n} = O(n^2). The new matrix model multiplication has a time complexity of O(log n), making this algorithm significantly faster than O(n log n log log n) and O(n log n). Run-time analysis of the main algorithm of the new matrix model that takes a constant number of steps for any input of size n.

Proof

Suppose we need to estimate the complexity of input of size *n* in the asymptotic sense.

Given an array of positive integers in matrix form:

$$\begin{pmatrix} n_1 & n_2 \\ n_3 & n_4 \end{pmatrix},$$

where, n_1, n_2, n_3, n_4 are the elements of an array, $n_3 = 10^n - n_1$, $n_4 = 10^n - n_2$, i.e., the n_3 element is the complement of the n_1 element and the n_4 element is the complement of the n_2 element, $n_1 - n_2 = n_4 - n_3 = n_5$, $n_3 - n_2 = n_4 - n_1 = n_6$, $n_1 + n_3 = n_2 + n_4 = sbasis = 10^n$, where the exponent of 10^n is equal to the number of digits in the n_1, $n_2 < n_1 < n_3 < n_4$ and the options of an array are: (n_1n_2), (n_3n_4), (n_3n_2), (n_1n_4), $(n_1n_2) < (n_2n_3) < (n_1n_4) < (n_3n_4)$, $(n_1n_2) + (n_2n_3) + (n_1n_4) + (n_3n_4) = sbasis^2 = (10^n)^2$. Let us define an upper bound for the worst-case running time of an algorithm to define the value of $(n_1n_2) = n$ option.

Run-time analysis: Prove that Esbasis:2-(Esbasis:2:(sbasis^2:(Isbasis-Lsbasis) = O(log n), (see lemma 18 to clarify), let *T*(*n*) be the execution time for this input of size *n*, where $\lim\limits_{n\to\infty}$, there exist positive constants and lower order terms that are not considered and can be ignored (omitted), then:

- T(n) = Esbasis:2-(Esbasis:2:(sbasis^2:(Isbasis-Lsbasis))) = $n_2$sbasis:2-($n_2$sbasis:2:(sbasis^2:($n_3$sbasis-$n_1$sbasis))) = (n_2:2-(n_2:2:(sbasis:(n_3-n_1))))·sbasis
- T(n) = T_1(n)+ T_2(n) + T_3(n) + T_4(n) + T_5(n) + T_6(n) + T_7(n) + T_8(n) = *f*(n)
- T_1(n) = sbasis-n_1 = n_3 = 10^n- n_1⇒ O(n)
- T_2(n) = $n_2$sbasis = 10^n·n_2⇒can be ignored
- T_3(n) = $n_2$sbasis:2 = (10^n·n_2):2⇒ O(log n)
- T_4(n) = sbasis^2 = $(10^n)^2$⇒can be ignored
- T_5(n) = ($n_3$sbasis-$n_1$sbasis) = (10^n·n_3)-(10^n·n_1)⇒O(n)
- T_6(n) = sbasis^2:($n_3$sbasis-$n_1$sbasis) = $(10^n)^2$:((10^n·n_3)-(10^n·n_1))⇒O(log n)
- T_7(n) = $n_3$sbasis = 10^n·n_3⇒can be ignored
- T_8(n) = $n_1$sbasis = 10^n·n_1⇒can be ignored

Let *f* and *g* be functions from positive numbers to positive numbers, where *f*(n) = (n_2:2-(n_2:2:(10^n:(n_3-n_1))))·10^n = O(n) and *g*(n) = O(log n). Prove the claim that *f*(n) is O(*g*(n)) if there exist positive constants c>0 and n_o>0 such that:

$$f(n) \le c * g(n) \text{ for all } n \ge n_o.$$

To prove big-O, we choose values for c and n_0 and prove n>1 implies *f*(n)≤c∗g(n):

1. Choose n_0 = 1,
2. Assuming n>1, find/derive a c such that:

$$\frac{f(n)}{g(n)} \le \frac{cg(n)}{g(n)} = c$$

that proves that n>1 implies *f*(*n*)≤*c*∗*g*(*n*). This means that function *f*(*n*) does not grow faster than *g*(*n*), or that function *g*(*n*) is an upper bound for *f*(*n*) for all sufficiently large *n*→∞.

An algorithm asymptotic running time is O(log n), using Random Access Memory and on a Turing Machine, the complexity is O(n), concerning the rate of growth. The worst-case time complexity W(n) then defined as *W*(*n*) = max(*T*$_1$(*n*), *T*$_2$(*n*), …).

Notice. The complexity of addition is considered as O(log n) and the complexity of subtraction we also consider as O(log n), since we can use Random Access Memory addressing to speed up lookup. Addition can be executed very quickly, it runs in O(log n) if *n* is the value of the sum. Subtraction n_1 from 10^n to find n_3 that is the complement of n_1 has easy running time, it’s like we need to find the 10’s complement of a number and the time complexity of log10 is roughly constant time on most modern processors, it doesn’t matter the base of log. An algorithm takes a constant number of steps for any value of *n* larger than 1.

Comparing the asymptotic running time:

an algorithm that runs in O(n) time is better than one that runs in O(n log n) and O(log n) is better than O(n).

Lemma n

Verifying instance *n*. Defining the product of two large numbers that are taken arbitrarily. It is tedious enough to multiply large numbers by hand even using the new matrix when the sizes of the numbers are more than 10^{12}.

Proof

Given the new matrix:

$$newMatrix = \begin{pmatrix} 8724311093647731827 & 75683119214541796 \\ 1275688906352268173 & 9924316880785458204 \end{pmatrix} \rightarrow \begin{pmatrix} L & E \\ I & T \end{pmatrix},$$

where, L = 8724311093647731827, E = 75683119214541796, I = 1275688906352268173, T = 9924316880785458204, $J = L\text{-}E = T\text{-}I$ = 8648627984433190031, $W = T\text{-}L = I\text{-}E$ = 1200005787137726377, sbasis = L + I = E + T = 10000000000000000000, this sbasis is taken successively, therefore is restricted, Jsbasis = 864862798443319003100000000000000000000, Wsbasis = 120000578713772637700000000000000000000, here, $L>I$, $E<T$, $E<L>I<T$ and the matrix has to be turned, or we can leave a matrix, further, LE = a, IT = A, IE = b, LT = B, a + Wsbasis = A, b + Jsbasis = B, a + A + b + B = Asbasis^2 = 100000000000000000000000000000000000000, a<A, b<B, A<B, a + b = Esbasis, a + B = Lsbasis, b + A = Isbasis, A + B = Tsbasis, let us define the values of options (a), (b), and (B) using the value of (A) option and these properties of the new matrix, then:

- A = 1275688906352268173 · 9924316880785458204 = 12660340947942554572836836990490941292
- a = A-Wsbasis = 12660340947942554572836836990490941292-120000578713772637700000000000000000000 =
- = 660283076565290802836836990490941292
- b = Isbasis-A = 127568890635226817300000000000000000000-12660340947942554572836836990490941292 = 96548115580127157163163009509058708
- B = Tsbasis-A = 992431688078545820400000000000000000000-12660340947942554572836836990490941292 = 865828278599120274671631630095090587 08

Results

It is possible to multiply large numbers in time O(log n), with Random Access Memory.

Discussion

The paper has answered the following great thoughts and statements of complexity, refuting or confirming them that is obvious to all who read this work, since everything changes in complexity now: "Complexity theory is still an infant science. Thus, we still do not have complete answers for any of these questions (Papadimitriou, 1993)." "Computational complexity is one of the most beautiful fields of modern mathematics, and it is increasingly relevant to other sciences ranging from physics to biology, but this beauty is often buried underneath layers of unnecessary formalism, and exciting recent results…are usually considered too "advanced" (Moore and Mertens, 2011)." "…there never was an agreement over what kinds of topics should be included in the discipline, and it was very difficult to come up with a common understanding of how research in computing should ideally be done (Tedre, 2014)." "…practical computational tasks require the ironclad quantities that only mathematics provides (Lewis and Papadimitriou, 1998)." "…we are interested in finding the most "efficient" algorithm for solving a problem…However, by the "most efficient" one normally means the fastest (Garey and Johnson, 1979)." "For many problems, no polynomial-time algorithms are known (Arora and Safra, 1998)." "The advantage of having only a finite number of states is that we can implement the system with a fixed set of resources (Hopcroft and Ulman, 1979)." Algorithms whose running times are n^2 in the size of their inputs can be implemented to execute efficiently even for fairly large values of n, but algorithms that require an exponential running time can be executed only for small values of n (Homer and Selman, 2011)." "The most notable practical algorithm that has an exponential worst-case running time is the simplex algorithm for linear programming (Cook, 1983)." "Indeed, I believe that virtually every important aspect of programming arises somewhere in the context of sorting or searching (Knuth, 1997)!" "A search problem consists of a specification of a set of valid solutions (possibly an empty one) for each possible instance (Goldreich, 2010)." "…the set of sequences can be partitioned into computable and noncomputable sequences (Hartmanis and Stearns, 1965)." "…maturing understandings constantly lead to new insights and questions about existing models (Denning, 2012)." "We need a different theory: a theory of analog computation, where states and processes are inherently continuous, and which treats real numbers not as sequences of digits but as quantities in themselves (Moore, 1999)." "The reduction consists of computing $O(n^2)$ linear combinations of pairs of rows in the manner of standard Gaussian elimination except that now we get an additive term of the first kind at each step (Valiant, 1979)." "Computer problems come in different varieties; some are easy, and some are hard. For example, the sorting problem is an easy one (Sipser, 2012)." "Usually there are several possible algorithms for solving a problem such as evaluation of an algebraic expression, sorting a file, or parsing a string of symbols (Rabin, 2012).” “…one of the outstanding problems in computer science is determining whether questions exist whose answer can be quickly checked, but which require an impossibly long time to solve by any direct procedure (CMI, 2020).” “NP is in PP by adding a larger number of dummy accepting paths… PP is the prototypical counting class, classes defined in terms of the number of accepting and rejecting paths (Fortnow, 2002).”

Conclusion

A major measure of computational complexity the time complexity of $M(n) = O(n^2)$ has been reduced to $O(\log n)$, with Random Access Memory and not only this problem has been solved, but we have also solved the problem of redundant computations that means that the computations are only required on a quarter of the inputs instead of all the input set (the amount of resources needed for running an algorithm varies with the input and the size of the input, for example, when the input size *n* goes to infinity), i.e., there are at least three redundant computations at each step, which can increase the runtime of the code significantly that makes web pages loading slow, therefore these steps have been reduced.

Acknowledgment

I would like to express my special thanks of gratitude to all my teachers.

Funding Information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The author has NO conflicts of interest to disclose.

Ethics

This article is original and contains unpublished material.

References

Arora, S., & Barak, B. (2009). Computational complexity: a modern approach. Cambridge University Press.

Arora, S., & Safra, S. (1998). Probabilistic checking of proofs: A new characterization of NP. Journal of the ACM (JACM), 45(1), 70-122.

CMI. (2020). “P vs NP Problem | Clay Mathematics Institute”. www.claymath.org

Cook, S. A. (1983). An overview of computational complexity. Communications of the ACM, 26(6), 400-408.

Denning, P. J. (2012). Opening Statement: What is Computation? The Computer Journal 55.7 (2012): 805-810.

Du, D. Z., & Ko, K. L. (2000). Theory of Computational Complexity. John Wiley & Sons.

Fortnow, L. (2002). Computational Complexity Blog: Complexity Class of the Week. http://weblog.fortnow.com/2002/09/complexity-class-of-week.html

Garey, M. R., & Johnson, D. S. (1979). Computers and intractability (Vol. 174). San Francisco: freeman.

Goldreich, O. (2008). Computational complexity: a conceptual perspective. ACM Sigact News, 39(3), 35-39.

Goldreich, O. (2010). P, NP, and NP-Completeness: The basics of computational complexity. Cambridge University Press.

Hartmanis, J., & Stearns, R. E. (1965). On the computational complexity of algorithms. Transactions of the American Mathematical Society, 117, 285-306.

Homer, S., & Selman, A. (2011). Turing and the development of computational complexity. Department of Computer Science.

Hopcroft, J. E., & Ullman, J. D. (1979). Introduction to Automata Theory, Languages, and Computation. Adison-Wesley. Reading, Mass.

Knuth, D. E. (1997). The art of computer programming (Vol. 3). Pearson Education.

Lewis, H. R., & Papadimitriou, C. H. (1998). Elements of the Theory of Computation. ACM SIGACT News, 29(3), 62-78.

Moore, C. (1996). Recursion theory on the reals and continuous-time computation. Theoretical Computer Science, 162(1), 23-44.

Moore, C., & Mertens, S. (2011). The nature of computation. OUP Oxford.

Papadimitriou, C. (1993). Computational Complexity. Addison-Wesley.

Rabin, M. O. (2012). Turing, Church, Gödel, Computability. Complexity and Randomization: A Personal View.

ScienceDirect. (2020). Matrix Computation-an overview. ScienceDirect Topics. https://www.sciencedirect.com/topics/mathematics/matrix-computation

Sipser, M. (2012). Introduction to the Theory of Computation. Cengage learning.

TCMGL. (2019). We've found a quicker way to multiply really big numbers. The Conversation Media Group Ltd. http://theconversation.com/weve-found-a-quicker-way-to-multiply-really-big-numbers-114923

Tedre, M. (2014). The science of computing: shaping a discipline. CRC Press.

Valiant, L. G. (1979). The complexity of computing the permanent. Theoretical computer science, 8(2), 189-201.

Van Leeuwen, J., & Leeuwen, J. (Eds.). (1990). Handbook of theoretical computer science: Algorithms and complexity (Vol. 1). Elsevier.

Wikipedia. (2020a). Theory of computation. Wikipedia. https://en.m.wikipedia.org/wiki/Theory_of_computation

Wikipedia. (2020b). Model of computation. Wikipedia. https://en.wikipedia.org/wiki/Model_of_computation

Wikipedia. (2020c). Matrix (mathematics). Wikipedia. https://en.m.wikipedia.org/wiki/Matrix_(mathematics)