Karatsuba algorithm

From Wikipedia, the free encyclopedia

The **Karatsuba algorithm** is a fast multiplication algorithm. It was discovered by Anatolii Alexeevitch Karatsuba in 1960 and published in 1962.[1][2][3] It reduces the multiplication of two *n*-digit numbers to at most $n^{\log_2 3} \approx n^{1.585}$ single-digit multiplications in general (and exactly $n^{\log_2 3}$ when *n* is a power of 2). It is therefore faster than the classical algorithm, which requires n^2 single-digit products. For example, the Karatsuba algorithm requires $3^{10} = 59{,}049$ single-digit multiplications to multiply two 1024-digit numbers ($n = 1024 = 2^{10}$), whereas the classical algorithm requires $(2^{10})^2 = 1{,}048{,}576$.

The Karatsuba algorithm was the first multiplication algorithm asymptotically faster than the quadratic "grade school" algorithm. The Toom–Cook algorithm is a faster generalization of Karatsuba's method, and the Schönhage–Strassen algorithm is even faster, for sufficiently large *n*.

Contents

History

The standard procedure for multiplication of two *n*-digit numbers requires a number of elementary operations proportional to n^2, or $\Theta(n^2)$ in the big-O notation. In 1952, Andrey Kolmogorov conjectured that the classical algorithm was *asymptotically optimal,* meaning that any algorithm for that task would require $\Omega(n^2)$ elementary operations.

In 1960, Kolmogorov organized a seminar on mathematical problems in cybernetics at the Moscow State University, where he stated the $\Omega(n^2)$ conjecture and other problems in the complexity of computation. Within a week, Karatsuba, then a 23-year-old student, found an algorithm (later it was called "divide and conquer") that multiplies two *n*-digit numbers in $\Theta(n^{\log_2 3})$ elementary steps, thus disproving the conjecture. Kolmogorov was very agitated about the discovery; he communicated it at the next meeting of the seminar, which was then terminated. Kolmogorov did some lectures on the Karatsuba result at the conferences all over the world (see, for example, "Proceedings of the international congress of mathematicians 1962", pp. 351--356, and also "6 Lectures delivered at the International Congress of Mathematicians in Stockholm, 1962") and published the method in 1962, in the Proceedings of the USSR

Academy of Sciences. The article had been written by Kolmogorov and contained two results on multiplication, Karatsuba's algorithm and a separate result by Yuri Ofman; it listed "A. Karatsuba and Yu. Ofman" as the authors. Karatsuba only became aware of the paper when he received the reprints from the publisher.[2]

Algorithm

The basic step

The basic step of Karatsuba's algorithm is a formula that allows us to compute the product of two large numbers x and y using three multiplications of smaller numbers, each with about half as many digits as x or y, plus some additions and digit shifts.

Let x and y be represented as n-digit strings in some base B. For any positive integer m less than n, one can write the two given numbers as

$$x = x_1 B^m + x_0$$
$$y = y_1 B^m + y_0,$$

where x_0 and y_0 are less than B^m. The product is then

$$xy = (x_1 B^m + x_0)(y_1 B^m + y_0)$$
$$xy = z_2 B^{2m} + z_1 B^m + z_0$$

where

$$z_2 = x_1 y_1$$
$$z_1 = x_1 y_0 + x_0 y_1$$
$$z_0 = x_0 y_0$$

These formulae require four multiplications, and were known to Charles Babbage.[4] Karatsuba observed that xy can be computed in only three multiplications, at the cost of a few extra additions. With z_0 and z_2 as before we can calculate

$$z_1 = (x_1 + x_0)(y_1 + y_0) - z_2 - z_0$$

which holds since

$$z_1 = x_1 y_0 + x_0 y_1$$
$$z_1 = (x_1 + x_0)(y_1 + y_0) - x_1 y_1 - x_0 y_0$$

A more efficient implementation of Karatsuba multiplication can be set as [5] $xy = (b^2 + b)x_1 y_1 - b(x_1 - x_0)(y_1 - y_0) + (b + 1)x_0 y_0$, where b is the power where the split occurs of x_1.

Example

To compute the product of 12345 and 6789, choose $B = 10$ and $m = 3$. Then we decompose the input operands using the resulting base ($B^m = \mathit{1000}$), as:

$12345 = \mathbf{12} \cdot \mathit{1000} + \mathbf{345}$
$6789 = \mathbf{6} \cdot \mathit{1000} + \mathbf{789}$

Only three multiplications, which operate on smaller integers, are used to compute three partial results:

$z_2 = \mathbf{12} \times \mathbf{6} = 72$
$z_0 = \mathbf{345} \times \mathbf{789} = 272205$
$z_1 = (\mathbf{12} + \mathbf{345}) \times (\mathbf{6} + \mathbf{789}) - z_2 - z_0 = 357 \times 795 - 72 - 272205 = 283815 - 72 - 272205 = 11538$

We get the result by just adding these three partial results, shifted accordingly (and then taking carries into account by decomposing these three inputs in base *1000* like for the input operands):

result $= z_2 \cdot B^{2m} + z_1 \cdot B^m + z_0$, i.e.
result $= 72 \cdot \mathit{1000}^2 + 11538 \cdot \mathit{1000} + 272205 = \mathbf{83810205}$.

Note that the intermediate third multiplication operates on an input domain which is less than twice larger than for the two first multiplications, its output domain is less than four times larger, and base-*1000* carries computed from the first two multiplications must be taken into account when computing these two subtractions; but note also that this partial result z_1 cannot be negative: to compute these subtractions, equivalent additions using complements to $\mathit{1000}^2$ can also be used, keeping only the two least significant base-*1000* digits for each number:

$z_1 = 283815 - 72 - 272205 = (283815 + 999928 + 727795)$ **mod** $\mathit{1000}^2 = 2011538$ **mod** $\mathit{1000}^2 = 11538$.

Recursive application

If n is four or more, the three multiplications in Karatsuba's basic step involve operands with fewer than n digits. Therefore, those products can be computed by recursive calls of the Karatsuba algorithm. The recursion can be applied until the numbers are so small that they can (or must) be computed directly.

In a computer with a full 32-bit by 32-bit multiplier, for example, one could choose $B = 2^{31} =$ 2,147,483,648 or $B = 10^9 =$ 1,000,000,000, and store each digit as a separate 32-bit binary word. Then the sums $x_1 + x_0$ and $y_1 + y_0$ will not need an extra binary word for storing the carry-over digit (as in carry-save adder), and the Karatsuba recursion can be applied until the numbers to multiply are only 1-digit long.

Efficiency analysis

Karatsuba's basic step works for any base B and any m, but the recursive algorithm is most efficient when m is equal to $n/2$, rounded up. In particular, if n is 2^k, for some integer k, and the recursion stops only when n is 1, then the number of single-digit multiplications is 3^k, which is n^c where $c = \log_2 3$.

Since one can extend any inputs with zero digits until their length is a power of two, it follows that the number of elementary multiplications, for any *n*, is at most $3^{\lceil \log_2 n \rceil} \leq 3n^{\log_2 3}$.

Since the additions, subtractions, and digit shifts (multiplications by powers of *B*) in Karatsuba's basic step take time proportional to *n*, their cost becomes negligible as *n* increases. More precisely, if *t*(*n*) denotes the total number of elementary operations that the algorithm performs when multiplying two *n*-digit numbers, then

$$T(n) = 3t(\lceil n/2 \rceil) + cn + d$$

for some constants *c* and *d*. For this recurrence relation, the master theorem gives the asymptotic bound $T(n) = \Theta(n^{\log_2 3})$.

It follows that, for sufficiently large *n*, Karatsuba's algorithm will perform fewer shifts and single-digit additions than longhand multiplication, even though its basic step uses more additions and shifts than the straightforward formula. For small values of *n*, however, the extra shift and add operations may make it run slower than the longhand method. The point of positive return depends on the computer platform and context. As a rule of thumb, Karatsuba is usually faster when the multiplicands are longer than 320–640 bits.[6]

Pseudocode

```
procedure karatsuba(num1, num2)
  if (num1 < 10) or (num2 < 10)
    return num1*num2
  /* calculates the size of the numbers */
  m = max(size_base10(num1), size_base10(num2))
  m2 = m/2
  /* split the digit sequences about the middle */
  high1, low1 = split_at(num1, m2)
  high2, low2 = split_at(num2, m2)
  /* 3 calls made to numbers approximately half the size */
  z0 = karatsuba(low1,low2)
  z1 = karatsuba((low1+high1),(low2+high2))
  z2 = karatsuba(high1,high2)
  return (z2*10^(2*m2))+((z1-z2-z0)*10^(m2))+(z0)
```

References

1. A. Karatsuba and Yu. Ofman (1962). "Multiplication of Many-Digital Numbers by Automatic Computers". *Proceedings of the USSR Academy of Sciences* **145**: 293–294. Translation in the academic journal *Physics-Doklady*, **7** (1963), pp. 595–596
2. A. A. Karatsuba (1995). "The Complexity of Computations" (http://www.ccas.ru/personal/karatsuba/divcen.pdf) (PDF). *Proceedings of the Steklov Institute of Mathematics* **211**: 169–183. Translation from Trudy Mat. Inst. Steklova, 211, 186–202 (1995)
3. Knuth D.E. (1969) *The art of computer programming. v.2.* Addison-Wesley Publ.Co., 724 pp.
4. Charles Babbage, Chapter VIII – Of the Analytical Engine, Larger Numbers Treated, Passages from the Life of a Philosopher (http://books.google.com/books?id=Fa1JAAAAMAAJ&pg=PA125), Longman Green, London, 1864; page 125.
5. Torbjörn Granlund and the GMP development team, *The GNU Multiple Precision Arithmetic Library Manual,*

version 6.0.0, Free Software Foundation, Inc., March 2014.

6. [1] (http://gmplib.org/manual/Karatsuba-Multiplication.html)[2] (http://ozark.hendrix.edu/~burch/proj/karat/comment1.html)

External links

- Karatsuba's Algorithm for Polynomial Multiplication (http://www.cs.pitt.edu/~kirk/cs1501/animations/Karatsuba.html)
- Weisstein, Eric W., "Karatsuba Multiplication" (http://mathworld.wolfram.com/KaratsubaMultiplication.html), *MathWorld*.
- Karatsuba multiplication Algorithm – Web Based Calculator (GPL) (http://utilitymill.com/utility/Karatsuba_Multiplication)
- Bernstein, D. J., "Multidigit multiplication for mathematicians (http://cr.yp.to/papers/m3.pdf)". Covers Karatsuba and many other multiplication algorithms.
- Karatsuba Multiplication on Fast Algorithms and the FEE (http://www.ccas.ru/personal/karatsuba/divcen.htm)
- Karatsuba multiplication in C++ (https://github.com/yaojingguo/cpp-code/blob/master/kmul.cc)

Retrieved from "http://en.wikipedia.org/w/index.php?title=Karatsuba_algorithm&oldid=660616367"

Categories: Computer arithmetic algorithms | Multiplication