Integer Multiplication

George E. Collins

September 15, 1998

When we perform decimal multiplication, we must be able to multiply any two decimal digits and obtain a product of two decimal digits. Likewise, in order to carry out the classical multiplication algorithm on SACLIB integers in base β-digit representation, we must be able to multiply any two nonnegative β-digits and obtain the product in the form of two nonnegative β-digits. DPRNN is an assembly language program for the UltraSPARC V9 architecture of ren that does this. The time for one execution of DPRNN is approximately 0.12 microseconds. Since SACLIB integers represented by lists may contain negative B-digits, it is convenient to have a similar program, DPR, that takes arbitrary β-digits as inputs. An execution of DPR requires about 0.14 microseconds.

The SACLIB program IPRODA uses the classical method to multiply any two non-negative integers that are stored in arrays, producing the product in a third array supplied by the user. This is a simple two-step program that is easy to understand. Table 1 shows times T in microseconds for applying IPRODA to two integers that are both n β-digits long.

IPROD multiplies two list-represented integers, producing the product in list representation. IPROD uses a hybrid algorithm, using the classical method whenever n, the maximum of the β-lengths of the two inputs, is less than or equal to an adjustable constant MAXCLASSICAL, a SACLIB global constant,

Table 1: IPRODA Times

n	T
1	0.3
2	1.0
3	2.2
4	3.8
5	5.6
10	20.4
20	78.0
40	303.2

and otherwise using the Karatsuba integer multiplication method. When the classical method is used, it is carried out by one of various subprograms, depending on input sizes. If either input has β-length 1, the multiplication is performed by IDPR (integer-digit product) in Steps 1 to 3. Otherwise, if neither β-length exceeds 3, Step 4 utilizes one of three very special-purpose subprograms that are intended to be extremely efficient. If both β-lengths are greater than 3, Step 5 converts the inputs from list representation to array representation using a subprogram IFLTA. Step 6 applies either IPRODA or KARATSUBA. Before applying KARATSUBA, Step 6 must first compute the size of a working array required by KARATSUBA and allocate this array.It must also allocate an array for the product and set all of elements of this array to 0. It is beyond the scope of this course to study this particular implementation of the Karatsuba method, due to Werner Krandick, which is a variation of the scheme discussed in Exercise 18 of Section 4.3.3 of Section 4.3.3 of [1]. That scheme is due to R. Maeder, 1993. However we may readily observe from the specification of KARATSUBA that the size of the working array that is required is less than or equal to $2n + 6\log_2 n$.

The computing time of IPRODA is, of course, dominated by mn, where m and n are the β-lengths of the two inputs. Moreover, its computing time, for non-zero inputs, is codominant with mn. This is a convenient way of asserting that (1) the maximum computing time is dominated by mn, and (2) the minimum computing time dominates mn. The reader should test his ability to prove this. As a corollary, the computing time of IPRODA is dominated by n^2, the larger of the lengths of the two inputs. Knuth gives a brief proof in Section 4.3.3, page 295, of [1] that the computing time of the Karatsuba method is dominated by $n^{\lg 3}$. Exercise 13 of Section 4.3.3 shows that a particular version of the Karatsuba method has a computing time dominated by $(m/n)m^{\lg 3}$ where m and n are the lengths of the two inputs and $m \geq n$. Does KARATSUBA have this property also?

The current value of MAXCLASSICAL is 14. Table 2 gives the computing times T in microseconds for applying IPROD to two integers that are both n β-digits long. Notice that, as one should expect, when n is doubled from 20 to 40 the computing time T is multiplied by approximately 3.

References

[1] Donald E. Knuth The Art of Computer Programming, Vol. 2, 3rd Ed. Addison-Wesley, 1997

Table 2: IPROD Times

n	T
1	0.54
2	1.82
3	3.56
4	6.22
5	8.45
10	25.5
20	81.1
40	247.2
80	758