

Algorithm for long multiplication

Author:Shefali Garg

11678

CSE

3 Sept 2012

Contents

1	Problem Statement	1
2	Algorithm	1
3	Analysing time complexity	2

1 Problem Statement

Write an algorithm to do long integer multiplication which was developed by Leslie Lamport[3],i.e., an algorithm to multiply two integers given as a list of their digits.

2 Algorithm

Given two integers,assign one with greater number of digits as primary number and other one as secondary number.If both have equal number of digits,consider any of them to be primary number.

Then take the last digit of the secondary number and do simple multiplication of that digit with the primary number.Write the obtained result just below the secondary number.Let us take an example from Donald Knuth's algo system[2] For ex-multiplication of 678 and 42 can be done as follows-

$$\begin{array}{r} 678 \\ \times 42 \\ \hline 1356 \\ 2712 \\ \hline 28476 \end{array}$$

Figure 1:
Example

678 - primary number
42 - secondary number

$$\begin{array}{r} 678 \\ * 42 \\ \hline 1356 \end{array}$$

Then take the second last digit of the secondary number and do simple multiplication with the primary number. Multiply the result obtained with 10 and write it just below the number obtained in the previous step. For example -

$$\begin{array}{r} 678 \\ * 42 \\ \hline 1356 \\ 27120 \end{array}$$

Then do simple addition on all the numbers got above. The result is the required answer.

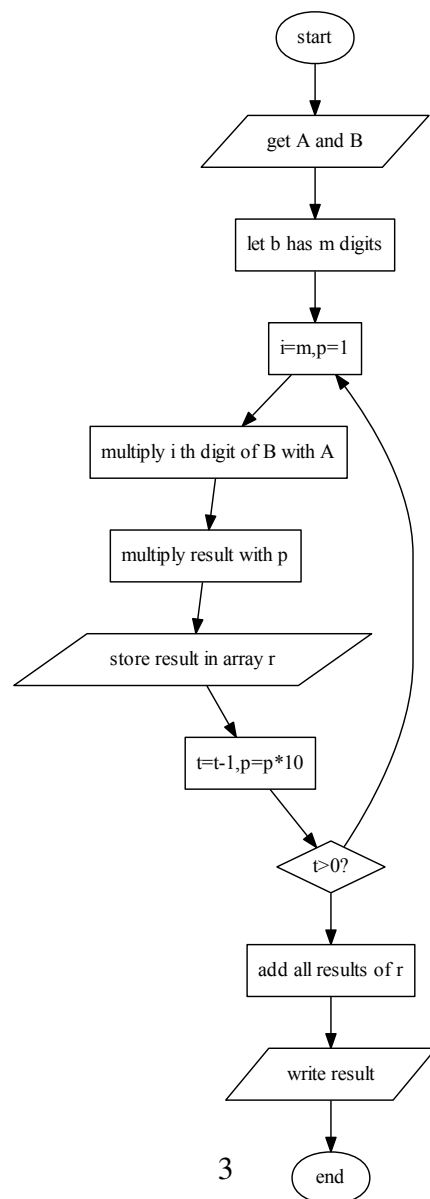
$$\begin{array}{r} 678 \\ * 42 \\ \hline 1356 \\ 27120 \\ \hline 28476 \end{array}$$

28476 is the required result(product) of the multiplication of 678 and 42.

3 Analysing time complexity

Let the two given numbers are of size n . In this algorithm, each digit of the secondary number is multiplied with each digit of the primary number once. So, the total number of steps taken here is n^2 , i.e., of $O(n^2)$. The addition of the two n digit numbers has the time complexity of order n . Here, the addition of n numbers is done so the time complexity of addition process is of order n^2 . So, the overall time complexity of the algorithm is $O(n^2)$ and the time taken by the algorithm is proportional to n^2 . There is a detailed explanation of complexity in Companion[1] and Rahtz's[4]

Figure 2: flowchart



References

- [1] Michel Goossens, Frank Mittlebach, and Alexander Samarin. *The Latex Companion*. Addison-Wesley, Reading, Massachusetts, 1993.
- [2] Donald E. Knuth. *Tex and Metafont, New Directions in Typesetting*. American Mathematical Society and Digital Press, Stanford, 1979.
- [3] Leslie Lamport. *Latex: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.
- [4] Sebastian Rahtz. A survey of Tex and graphics. Technical Report CSTR 89-7, Department of Electronics and Computer Science, University of Southampton, UK, 1989.