usepackagealgorithm usepackagealgorithmic

CS251–ASSIGNMENT 3

DOCUMENT ON LaTeX

Algorithm for long integer multiplication

AUTHOR:
GORANK DUDEJA(11283)
G4

Algorithm for Long Integer Multiplication

November 5, 2012

TABLE OF CONTENTS

Contents

1 Explanation of the algorithm

This algorithm is really very simple.The main points are listed below

- Let us suppose we have to multiply a and b where both a and b are integers.

- Let the smaller one of them be a and so a is multiplier and b is multiplicand.

- First copy a and b into two other variables,say m and n.

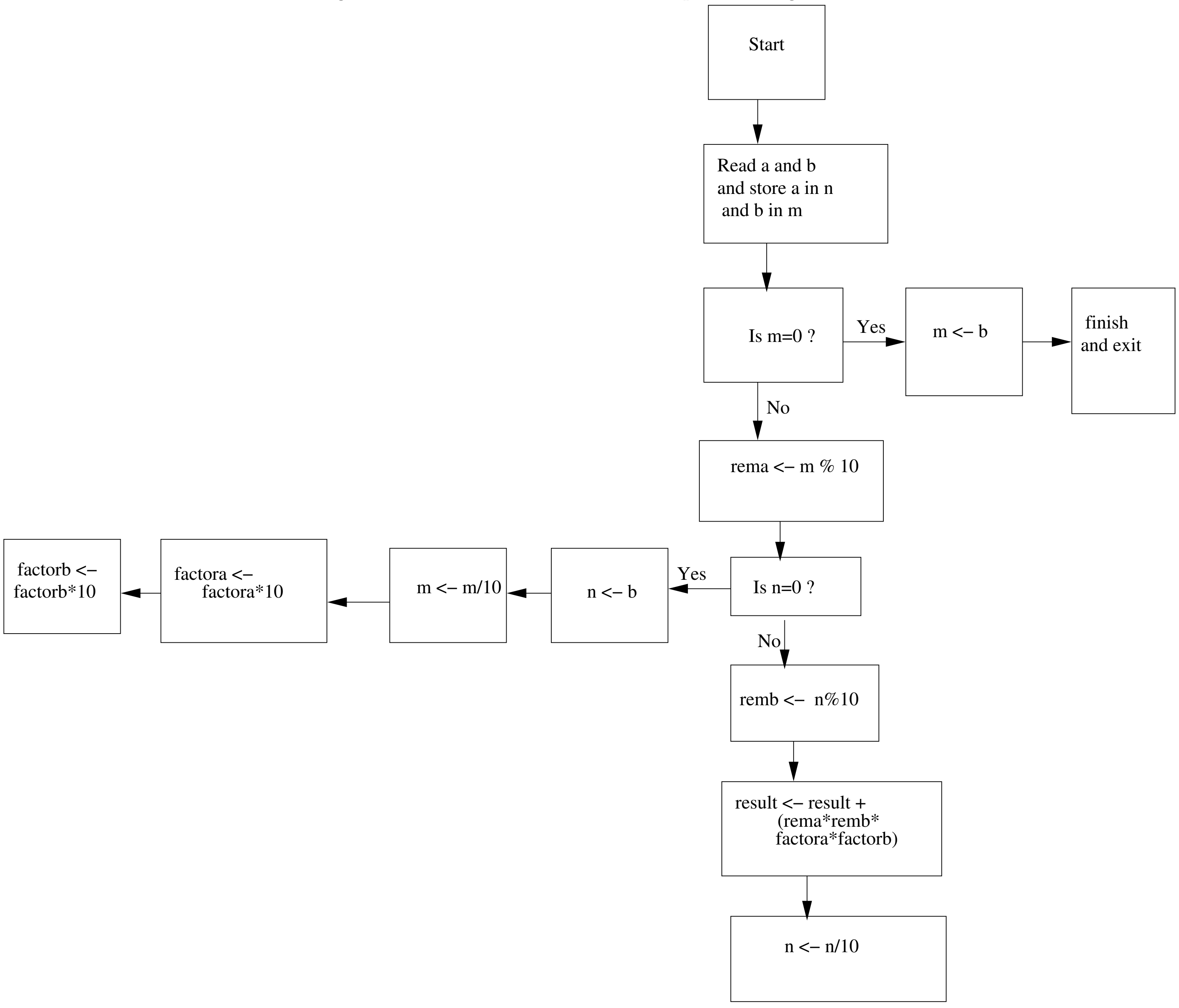
- Divide m and n into its digits considering the units place,tens place,hundreds place and so on. for example :- 324=300+20+4

- Now,multiply both the digits of m and n taking into account the corresponding factors of 10 and then add up all the products to get the result.

2 Pseudocode

Let a be the multiplier and b be the multiplier. Let factora and factorb be the factor of multiplication for a andb respectively Let rema be the last digit in a and remb be the last digit in b factora=1,factorb=1,result=0 $m \leftarrow a$ $n \leftarrow b$ 1 $rema \leftarrow m\%10$ 1 $remb \leftarrow n\%10$ $result \leftarrow$ result+(rema*remb*factora*factorb) $n \leftarrow n/10$ $n == 0$ $n \leftarrow b$ break $m \leftarrow m/10$ $factora \leftarrow factora * 10$ $factorb \leftarrow factorb * 10$ $m == 0$ $m \leftarrow b$ break

Figure 1: Flowchart to illustrate the steps of the algorithm



3 Flowchart

Figure 2: Example of multiplication of two integers

```
    283
  x  42
 ______________
    566          ( = 283  x 2 )
  11320            ( = 283  x 40 )

_________________
   11886
_________________
```

4 Example to illustrate the algorithm

5 Analysis of time complexity

This algorithm has **two loops** which are nested into each other.

The worst case that will be faced by the algorithm is when both the loops run upto n times where n is the number of digits in the two numbers.

Some other operations are also executed in the algorithm like modulus operator,initialisation etc.
but they all take constant time and so, the worst number of instructions executed by the algorithm is proportional to n^2 +k where k is a constant.

And so, the time complexity of the algorithm is O(n^2) where O stands for Big-O notation.