# Latex Documentation

Mohit Sharma

November 5, 2012

# 1 Multiplication of two long numbers

## 1.1 Problem:

Multiplying two numbers given as array of digits forming them. Maximum number of digits in each number is n.

## 1.2 Solution:

We solve the multiplication problem through the approach adopted in school. We save the first number as a string in array $A$, second number as string in array $B$. We store result of multiplication in an array $C$. We start from last element (before '/0') of array $B$. Keep multiplying it with digits of number $A$ starting again from the last element and going till the first digit of $A$. At each step we store the result of multiplication $(mod10)$ in same cell and carry (result/10) to next step.
Note: The result C is in reverse order.
Then we do the same with the second-last element of B and this time, we start adding result from secondlast element of C. and at each step we start from one element ahead than previous time. In the end, we will be left with answer to multiplication in reverse order.

## 1.3 Analysis of Time Complexity of the Algorithm

Algorithm 1 executes "for" loop on $i$, $b - 1$ times, and in each iteration it runs another for loop on $j$, $a - 1$ times doing some constant number of operations.

**Data**: Input will be two arrays $A[0..n-1]$ and $B[0..n-1]$. We will use an extra array $C[0..(2n-1)]$ to store the result at each step and hence the result of multiplication after final step.

**Result**: The result will be multiplication of two given number in digit form in array $C$

**1** Set all elements of $C$ to be zero.

**2** Get number of digits of $A$ and $B$, say it equals $a$ and $b$ respectively.

**3** **for** $i \leftarrow (b-1)$ **to** 0 **do**

**4**     k=$(b-1)$ - $i$;

**5**     **for** $j \leftarrow (a-1)$ **to** 0 **do**

**6**        $result= B[i] * A[j]$;

**7**        $C[k]= (C[k]+carry + result)$ % 10;

**8**        $carry= (C[k]+carry + result)/10$;

**9**        $k++$;

**10**     **end**

**11**     C[k]=C[k] + carry;

**12** **end**

**13** Reverse array $C$;

**14** return array$C$

**Algorithm 1:** Algorithm for multiplying two long numbers

Hence the total number of operations executed are $k * a * b$ where $k$ is a constant. Maximum value of a and b can be $n$ and hence worst case time complexity is $k * (n^2)$ or $\mathrm{O}(n^2)$.
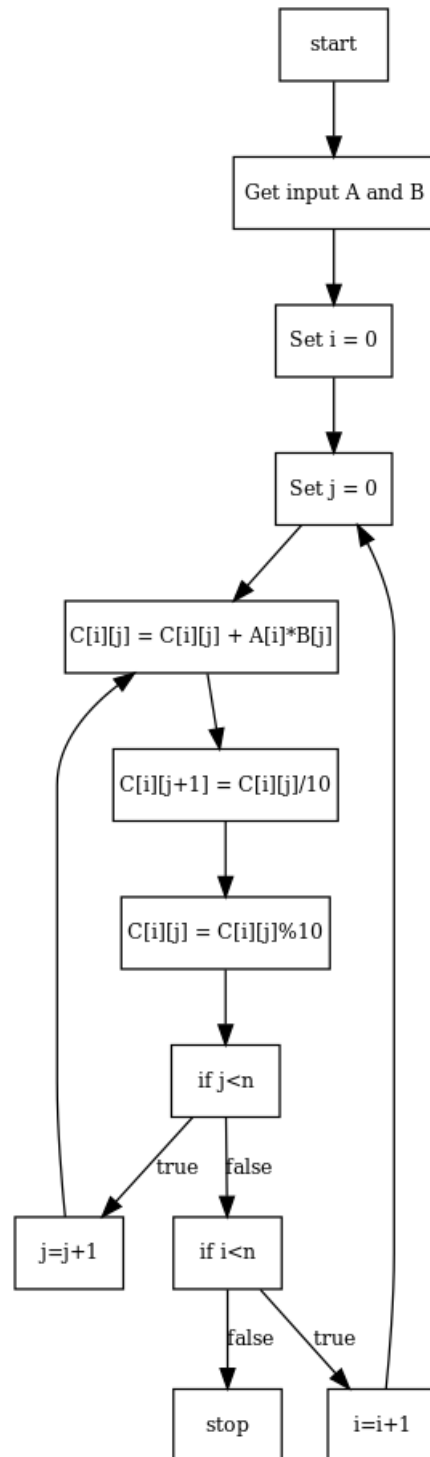
```
start
  ↓
Get input A and B
  ↓
Set i = 0
  ↓
Set j = 0
  ↓
C[i][j] = C[i][j] + A[i]*B[j]
  ↓
C[i][j+1] = C[i][j]/10
  ↓
C[i][j] = C[i][j]%10
  ↓
if j<n
  ├ true → j=j+1 → C[i][j] = C[i][j] + A[i]*B[j]
  └ false → if i<n
              ├ true → i=i+1 → Set j = 0
              └ false → stop
```

Figure 1: flowchart