Grade-school Algorithm for Integer Multiplication

Srijan R Shetty
11727

September 3, 2012

Abstract

The following report contains a brief description and analysis of the Grade-school algorithm for integer multiplication

Contents

1 Grade-school Algorithm

1.1 Problem

To design an algorithm to perform multiplication of two numbers stored as arrays of size n and m respectively using grade-school algorithm.

1.2 Solution

The Grade-school algorithm for the multiplication for two numbers is a trivial, bruteforce algorithm for the multiplication of two integers. In this algorithm, the each digit of the one of the numbers is multiplied with each digit of the other number. This product is then multiplied by a certain exponent raised to the base of the system and added to the final sum.

1.3 Data Structures Used

No extra data structures were used in the implementation of this algorithm

1.4 Algorithm

The psuedo code for the Grade-school algorithm is:

Data: Two arrays A, B of size n and m respectively, storing their digits
Output: The product A × B

```
1.  long int multiply(A[ ],B[ ])
2.      carry ← 0;
3.      for i=n-1 to 0 do
4.          for j=m-1 to 0 do
5.              prod ← A[i] × B[j];
6.              if j=0 then
7.                  digit ← prod+carry;
8.                  carry ← 0;
9.              else
10.                 digit ← (prod+carry)%10;
11.                 carry ← (prod+carry)/10;
12.             sum ← digit ×10^j;
13.         end
14.         total ← total + (sum ×10^i);
15.     end
16. print total
```

1.5 Analysis of Time and Space Complexity

In the above algorithm, the first for loop runs n times while the second one runs for m times. So the overall time complexity is $\mathbf{O(n)} \times \mathbf{O(m)}=\mathbf{O(mn)}$.

As the above algorithm does not use any additional data structures, so the space complexity is $\mathbf{O(1)}$

1.6 Summary

Data	Two arrays A[ ] and B[ ] of size n and m respectively
Output	The product A[ ] $\times$ B[ ]
Time Complexity	$\mathbf{O(mn)}$
Space Complexity	$\mathbf{O(1)}$