Data Structures and TAKE-HOME ASSIGNMENT 3

Index No: 210670N
Name: Vidushini.A.O

Katsuba Algorithm

- ## Introduction

Katsuba algorithm is a fast multiplication algorithm that can multiply two large integers without directly using the traditional multiplication algorithm, the algorithm has a recursive approach.

- ## Pseudocode for the algorithm

```
Procedure karatsuba(x, y)
   size1 → number of digits in x
   size2 → number of digits in
   if(size1 == 1 or size == 2)then,
      return (x *y)
  n → max(size1, size2)
  m → n/2
   x_0 → floor_division(x, 10^m)
   x_1 → x%(10^m)
   y_0 → floor_division(y, 10^m)
   y_1 → y % (10^m)
   p_0 → karatsuba(x_1, y_1)
   p_1 → karatsuba(x_0, y_0)
   p_2 → karatsuba(x_0 + x_1, y_0 + y_1)
   return(p_2*10^(2*m)) + ((p_1 -p_2-p_0 )* 10^m) + p_0
```

- ## Time complexity analysis

For this algorithm,

In each recursive step the problem is divided to 3 sub problems of T(n/2) ;(since each x and y are divided into two integers

$$T(n) = 3(T(n/2)) + c * n; for n \geq 1 and some c > 0 and T(n) = 1 which is the base case$$

Here is the time complexity of the Karatsuba algorithm using the master theorem.

$T(\mathrm{n}) = 3(T(n/2)) + c * n$

$a = 3, b = 2\ \mathrm{f(n)} = \mathrm{n}$

$\log_b a = \log_2 3 = 1.585$

$for\ c = 1\ and\ \varepsilon = 0.5 > 0,\ ;\ \ \mathrm{f(n)} = \mathrm{n}$ and $\mathrm{n} = O(\mathrm{n}^{\log_2 3 - \varepsilon})$

Then, by case 1 of the master theorem,

$T(n) = \Theta(\mathrm{n}^{\log_2 3})$

$\Rightarrow \mathrm{T(n)} = O(\mathrm{n}^{\log_2 3})$