BLG 335E
ANALYSIS OF ALGORITHMS 2
Project-2

Umut Cem Avin
150140018

1. Introduction

In this project, 2 different binary multiplication calculations are made. First of them is “Classic Binary Multiplication Method”. It is also named as Shift and Add Method. And the second one is Karatsuba algorithm. A program which gives decimal result of these multiplication algorithms and also a text file which includes results, running times and also number of digits, was written. This report includes necessary information about the implementation about these algorithms and the outputs.

2. Development Environment

OS: MacOS Mojave 10.14
IDE: XCode 10.0
Compile command: g++ 150140018.cpp -o project2
Execution command: ./project2 output.txt

3. Analysis

Implementation

The program consists of 5 different functions and 1 main function. There are “equalizeLength(), bitAddition(), convertToDecimal(), karatsubaMethod() and classicMethod() functions.

equalizeLength: This function makes the lengths of string equal by placing zeros to the left of strings.

bitAddition: This function makes additions operation between bits one by one.

karatsubaMethod: Makes the binary multiplication according to Karatsuba Algorithm. Details about this algorithm will be given in the next section of report.

classicMethod: Makes the binary multiplication according to classic method(Shift and add). Details about this algorithm will be given in the next section of report.

convertToDecimal: Results from classical method comes as a string binary number. This functions converts them to the decimal numbers.

Pseudocode for Karatsuba algorithm was already given in the homework description file as:

Algortihm 1:

```
function multiply(x, y)

Input: Positive integers x and y, in binary
Output: Their product

n = max(size of x, size of y)
if n = 1: return xy

xL, xR = leftmost [n/2], rightmost [n/2] bits of x
yL, yR = leftmost [n/2], rightmost [n/2] bits of y

P1 = multiply(xL, yL)
P2 = multiply(xR, yR)
P3 = multiply(xL + xR, yL + yR)

return P1 × 2^n + (P3 − P1 − P2) × 2^(n/2) + P
```

Figure 1: Pseudocode for Karatsuba

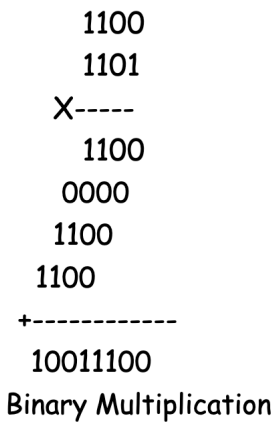

Figure 2: Binary multiplication in classic method

Running Times For Functions

Briefly, it was seem that Karatsuba algorithm works faster while the number of digits are increasing.

Karatsuba	Classic	Number of digits
2.3e-05	1.4e-05	2
1.3e-05	1.1e-05	4
5.5e-05	4.3e-05	8
0.000181	0.000188	16
0.000423	0.000671	32
0.001186	0.003464	64

$$T(n) = \sum_{k=0}^{\log_2 n} n \left(\frac{3}{2}\right)^k = \frac{\left(\frac{3}{2}\right)^{1+\log_2 n} - 1}{\frac{3}{2} - 1} = 3n^{\log_2 3} - 2$$