**CS2023 - Data Structures and Algorithms**

**Take Home Assignment Week 4**

March 23, 2023

By: Sajeev Kugarajah (210554M) – Group IV

**Karatsuba algorithm to multiply two largest integers.**

Unlike traditional multiplication algorithm, Karatsuba algorithm uses divide and conquer method which reduces the time complexity by a significant amount which makes Karatsuba algorithm more efficient when multiplying larger integers.

For example, here’s an implementation of Karatsuba algorithm for 1234\*5678 in base 10

**Recurrence relation for time complexity**

**Solving the recurrence relation using master theorem**

From this, we can conclude that, Karatsuba algorithm with O(n1.585) time complexity is way efficient than the naïve multiplication algorithm with O(n2) time complexity for multiplying larger integers.

**Appendix: Source code for implemented Karatsuba algorithm in C++**

#include <iostream>

#include <string>

#include <cmath>

using namespace std;

int karatsuba(int x, int y)

{

// if x or y is less than 10, then return the product

if (x < 10 || y < 10)

return x \* y;

// find the maximum length of x and y

int n = max(to\_string(x).length(), to\_string(y).length());

// find the middle of the number

int m = n / 2;

// split the number into two parts

int a = x / pow(10, m);

int b = x % (int)pow(10, m);

int c = y / pow(10, m);

int d = y % (int)pow(10, m);

// calculate the product of a and c

int ac = karatsuba(a, c);

// calculate the product of b and d

int bd = karatsuba(b, d);

// calculate the product of (a+b) and (c+d)

int adbc = karatsuba(a + b, c + d) - ac - bd;

// return the product

return ac \* pow(10, 2 \* m) + (adbc \* pow(10, m)) + bd;

}

int main()

{

int x, y;

cout << "Enter two numbers: ";

cin >> x >> y;

cout << "Product: " << karatsuba(x, y);

return 0;

}