Name: Florin Kuran
Date: 11th March 2023

Week # 5

(a) Brute-Force Time Complexity

Since this involves each bit potentially getting multiplied by each bit of the other integer, This can lead to a nested loop so the time complexity would be $O(n^2)$. This is due to brute-force going through potentially every bit.

(b) Divide & Conquer Algorithm

Karatsuba Algorithm

For a number x with n bits, we can represent it as $x = x_1 \cdot 2^{n/2} + x_0$ where x_1 is the upper half of the bits x and x_0 is the lower half. Similarly, $b = b_1 \cdot 2^{n/2} + b_0$.

The product: $(2^{\frac{n}{2}} a_1 + a_0)(2^{n/2} b_1 + b_0) = 2^n \cdot a_1 b_1 + 2^{\frac{n}{2}}(a_1 b_0 + a_0 b_1) + a_0 b_0$.

This is [redacted] compler compared to the Karatsuba Algorithm which lowers it to 3 multiplications instead of 4.

Karatsuba's insight: $(a_1 b_0 + a_0 b_1) = (a_1 + a_0)(b_1 + b_0) - a_1 b_1 - a_0 b_0$.

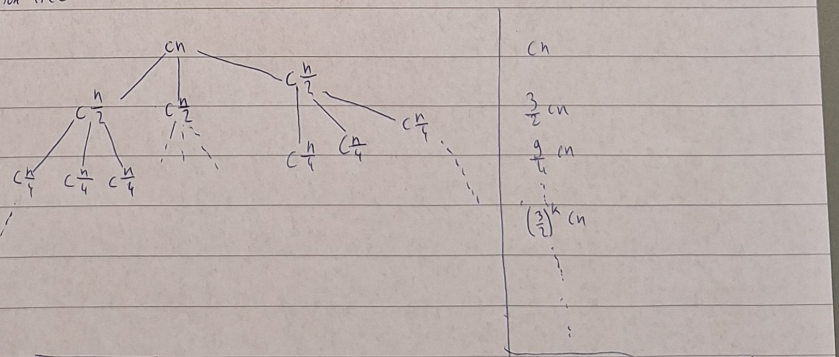
Hence only the products of $a_1 b_1$, $a_0 b_0$, and $(a_1 + a_0)(b_1 + b_0)$ are needed then we use them to calculate $a_1 b_0 + a_0 b_1$.

(c) Recurrence for Divide & Conquer Algorithm:

For Karatsuba's Algorithm it is known that the following recurrence relationship best describes it:

$$T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n)$$

(d) Recursion Tree



$O(n^{\log_2 3})$

$(\frac{3}{2})^{\log_3 n} cn = cn^{\log_2 3} \approx cn^{1.59}$

(e) Master Theorem

$T(n) = aT\left(\frac{n}{b}\right) + f(n)$ $\quad T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n)$

This fits Case 1 as we have $a=3, b=2$ so $f(n)$ is $O(n^{\log_b a - \varepsilon})$

In this case we find that $\boxed{T(n) = \Theta(n^{\log_2(3)})}$.

Since this is the same result that was found in the Recursive Tree we can confirm the result.

THE END