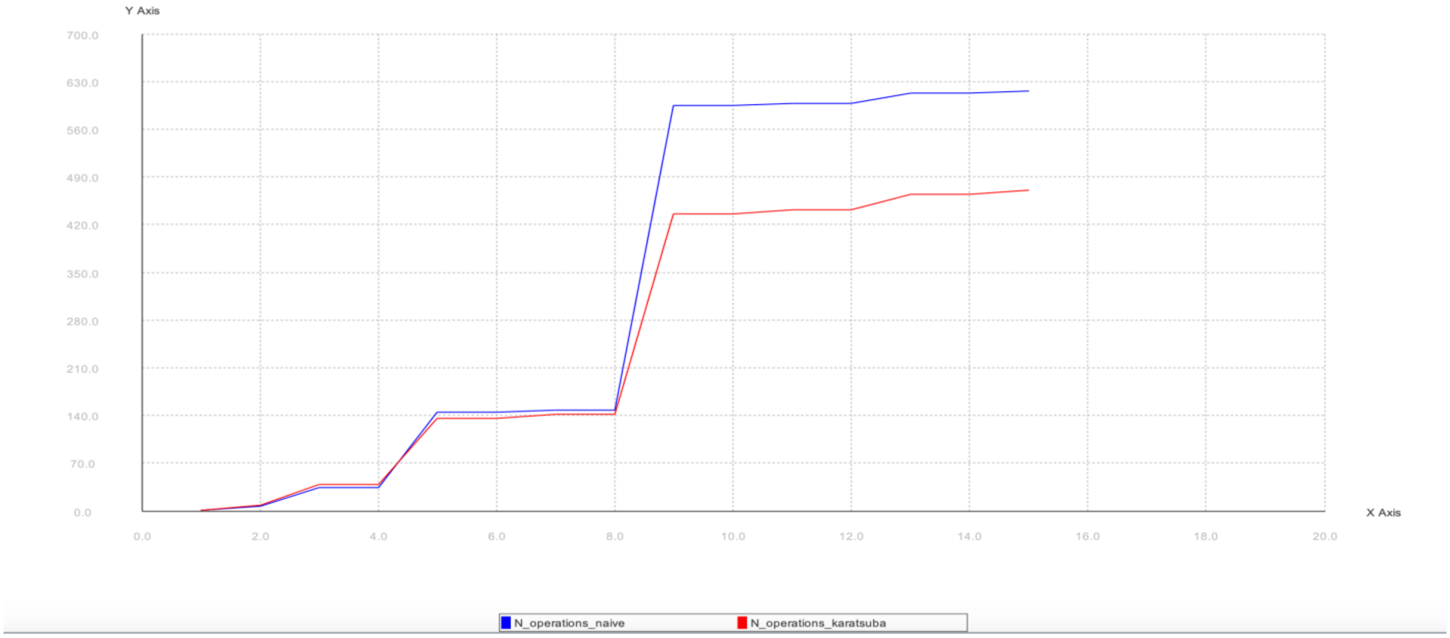

In the induction case, the naive method executes 3 arithmetic operations of integer of size m, in addition of the number of operations executed by each recursive call to the function. On the contrary, Karatsuba's algorithm requires 6 arithmetic operations of size m on top of the cost of the recursion.

But still, Karatsuba's algorithm has a lower time complexity.
This is because in the induction step, Karatsuba's algorithm calls itself 3 times, but the naïve algorithm calls itself 4 times.

As seen in the graph, Karatsuba is always either equal or below Naïve.
Karatsuba's algorithm requires $O(n^{1.585})$ bit operations, whereas Naïve algorithm requires $O(n^2)$ bit operations.

260737048

Assignment 4 Question 2

2 a) $T(n) = 25 \cdot T\left(\frac{n}{5}\right) + n$ $a = 25,\ b = 5$

$f(n) = n$

$f(n) = n = O\left(n^{2-(2-1)}\right)$

~~f(n)=n~~

$\therefore\ T(n) = \Theta(n^2)$

b) $T(n) = 2 \cdot T\left(\frac{n}{3}\right) + n \cdot \log(n)$ $a = 2$, $b = 3$

$f(n) = n \cdot \log(n)$

$f(n) = \Omega\left(n^{\log_3 2 + \epsilon}\right)$, case 03 applies if regularity condition holds

$a f\left(\frac{n}{b}\right) = 2 \cdot \frac{n}{3} \cdot \log\left(\frac{n}{3}\right) = \frac{2}{3} n \log(n) - \frac{2}{3} n \log\left(\frac{1}{3}\right) < \frac{2}{3} f(n)$

$\therefore\ c = \frac{2}{3}$

and $T(n) = \Theta(n \log n)$

c) $T(n) = T\left(\frac{3n}{4}\right) + 1$ $a = 1$ $b = \frac{4}{3}$

$f(n) = 1$ $f(n) = 1 = \Theta\left(n^{\log_{4/3} 1}\right)$

$T(n) = \cancel{\Theta(\log}\ \cancel{\Theta(\lg n)}\ \Theta(\lg n)$

d) $T(n) = 7 \cdot T\left(\frac{n}{3}\right) + n^3$, $f(n) = \Omega\left(n^{\log_3 7 + \epsilon}\right)$

$a = 7,\ b = 3$

$f(n) = n^3$ $7 f\left(\frac{n}{3}\right) = 7 \cdot \left(\frac{n}{3}\right)^3 = \frac{7}{27}(n^3)$

and $T(n) = \Theta n^3$ $\therefore\ c = \frac{7}{27}$

e) $T(n) = T\left(\frac{n}{2}\right) + n(2 - \cos n)$ ($n(2-\cos n)$ of the form $n^k \log^p n$)

so cannot appy Master Theorem.

Question 3

3) T_A returns running time of A. $T_A(n) = 7$

T_B returns running time of B

P

Question 3 (continued)

$T_A(n) = 7 \cdot T_n\left(\frac{n}{2}\right) + n^2$ $a = 7,\ b = 2,\ f(n) = n^2$

$f(n) = n^2 = O(n^{\log_2 7 - t})$

$t \approx 0.8$

Inorder for T_B to be faster, less time complexity is required than $T_A(n) = \Theta(n^{\lg 7})$

$T_B(n) = \alpha T_B(n/4) + n^2$ $a = \alpha$, $b = 4$, $f(n) = n^2$

$\log_2 7 = \log_4 49$

complexity should be less than $\alpha = 48$

So, $T_B(n) = \cancel{\Theta(n^{\lg 7})}\ \Theta\left(n^{\log_4 48}\right)$

$T_A(n) = \Theta\left(n^{\lg 7}\right)$