Part 1:

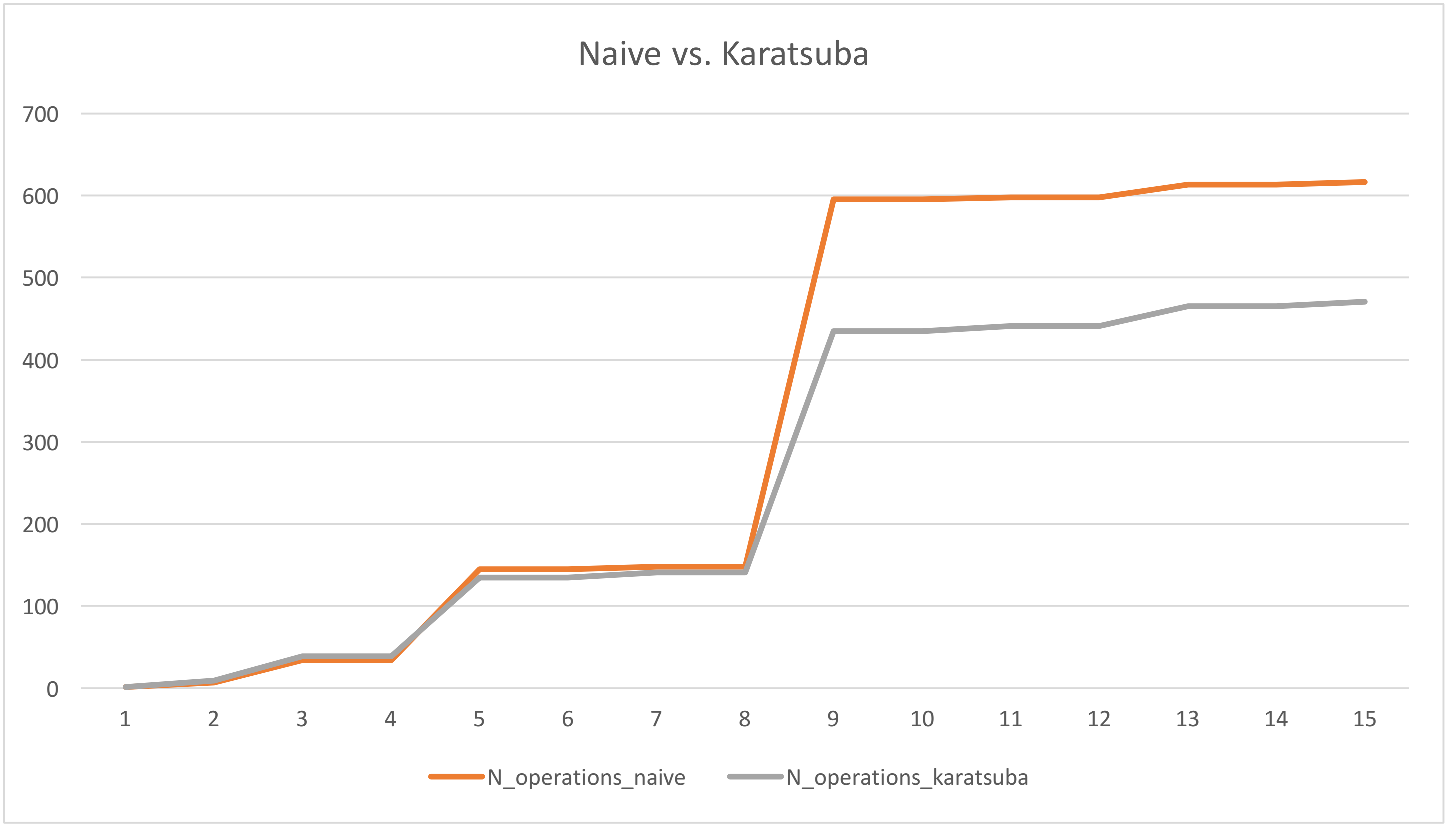

Figure 1: Comparison of Cost between Naïve and Karatsuba Multiplication Method

The above graph illustrates the grow in cost as the size of the input increases between naïve and Karatsuba multiplication methods. While the size is less than 4, the average cost for Karatsuba is higher than the naïve; however, Karatsuba's cost became less than the counterpart after. Furthermore, the gap between the two methods increases after size is equal to 8. As mentioned in the given guideline, Karatsuba algorithm requires 6 arithmetic operation while naïve algorithm only requires 3 arithmetic operation of integer of size m per recursive call. This is due to the fact that for Karatsuba, we have $2^{2m}e + 2^m(e + f - g) + f$. As the bitwise operation associated with the 2, the multiplication becomes negligible. Since there are 3 additions and 1 subtractions. Moreover, there are an additional 2 subtractions when we are providing the arguments to compute the variable g. Thus, there are 4 + 2 = 6 operations. On the contrary, in the naïve method, we have $2^{2m}e + 2^m(g + h) + f$, which contains 3 additions with some multiplications. Similarly, we omit the multiplication due to its fast computation associated with 2.

Part 2:

When possible, apply the master theorem to find the asymptotic behaviour of T (n) from the following recurrences. Show your work and justify your answer for each recurrence in the PDF file. Answer in the MyCourses quiz.

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

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

$$\boldsymbol{a} = 25,$$

$$\boldsymbol{b} = 5,$$

$$\boldsymbol{f(n)} = n,$$

$$\boldsymbol{k} = \log_5 25 = 2$$

Since $n^{\log_5 25} > f(n)$, we choose:

Case 1: If $\boldsymbol{f(n) = O(n^{k-\varepsilon})}$ for some constant $\boldsymbol{\varepsilon > 0}$, then $\boldsymbol{T(n) = \Theta(n^k)}$

$$f(n) = O(n^{k-\varepsilon}) = n = O(n^{2-\varepsilon})$$

$$let\ \varepsilon = 1\ and\ since\ 1 > 0,$$

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

$$\boldsymbol{T(n) = \Theta(n^k) = \Theta(n^2)}$$

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

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

$$\boldsymbol{a} = 2,$$

$$\boldsymbol{b} = 3,$$

$$\boldsymbol{f(n)} = n \cdot \log n,$$

$$\boldsymbol{k} = \log_3 2$$

Since $n^{\log_3 2} < f(n)$, we choose:

Case 3: If $\boldsymbol{f(n) = \Omega(n^{k+\varepsilon})}$ for some constant $\boldsymbol{\varepsilon > 0}$ and if $\boldsymbol{af\left(\frac{n}{b}\right) \le cf(n)}$ for some constant $\boldsymbol{c < 1}$ and all sufficiently large $\boldsymbol{n}$, then $\boldsymbol{T(n) = \Theta(f(n))}$.

Property 1: Let $\varepsilon = 1 - \log_3 2$ which is greater than 0, then
$f(n) = n \cdot \log n = \Omega(n^{\log_3 2 + (1 - \log_3 2)})$

$$\text{Property 2: let } c = \frac{2}{3}$$

$$2 \cdot \left(\frac{n}{3} \cdot \log \frac{n}{3}\right) \le c \cdot n \cdot \log n$$

$$\frac{2}{3} \cdot n \cdot \log \frac{n}{3} \le \frac{2}{3} \cdot n \cdot \log n$$

$$\boldsymbol{T(n) = \Theta(f(n)) = \Theta(n \cdot \log(n))}$$

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

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

$$\boldsymbol{a} = 1,$$

$$\boldsymbol{b} = \frac{4}{3},$$

$$\boldsymbol{f(n)} = 1,$$

$$\boldsymbol{k} = \log_{4/3} 1 = 0$$

Since $n^{\log_{4/3} 1} = f(n)$, we choose

Case 2: If $f(n) = \Theta(n^k \log^p n)$ then $T(n) = \Theta(n^k \log^{p+1} n)$

Since $k = 0$, by substituting k into $\Theta(n^k \log^p n)$, it yields $\Theta(n^0 \log^p n) = \log^p n$.

Then, we can make the equation $f(n) = \Theta(n^k \log^p n)$ holds by making $p = 0 \rightarrow \log^0 n = 1 = f(n)$

Thus, $\boldsymbol{T(n) = \Theta(n^k \log^{p+1} n) = \Theta(n^0 \log^{0+1} n) = \Theta(\log n)}$

(d) $\boldsymbol{T(n) = 7 \cdot T\left(\frac{n}{3}\right) + n^3}$

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

$$\boldsymbol{a} = 7,$$

$$\boldsymbol{b} = 3,$$

$$\boldsymbol{f(n)} = n^3,$$

$$\boldsymbol{k} = \log_3 7$$

Since $n^{\log_3 7} < f(n)$, we choose

Case 3: If $f(n) = \Omega(n^{k+\varepsilon})$ for some constant $\varepsilon > 0$ and if $af\left(\frac{n}{b}\right) \le cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$.

Property 1: Let $\varepsilon = 1 - \log_3 7$ which is greater than 0, then

$f(n) = n^3 = \Omega(n^{\log_3 7 + (2 - \log_3 7)})$

Property 2: let $c = \frac{7}{27}$, and $\frac{7}{27} < 1$

$$7 \cdot \left(\frac{n}{3}\right)^3 \le c \cdot n^3$$

$$\frac{7}{27} n^3 \le \frac{7}{27} n^3$$

$$\boldsymbol{T(n) = \Theta(f(n)) = \Theta(n^3)}$$

(e) $\boldsymbol{T(n) = T\left(\frac{n}{2}\right) + n(2 - \cos n)}$

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

$$\boldsymbol{a} = 1,$$

$$\boldsymbol{b} = 2,$$

$$\boldsymbol{f(n)} = n(2 - \cos n) = 2n - n\cos n,$$

$$\boldsymbol{k} = \log_2 1 = 0$$

Master Theorem does not apply. Periodicity of cos(n) invalidates the regularity condition.

Part 3:

Let T_A and T_B be two functions returning the running time of algorithms A and B, defined by the recursions $T_A(n) = 7T\left(\frac{n}{2}\right) + n^2$ and $T_B(n) = \alpha T_B\left(\frac{n}{4}\right) + n^2$. Find the largest integer value of α for which algorithm B is asymptotically faster than A. Show your work and justify your answer in the PDF file. Answer in the MyCourses quiz.

For $T_A = 7T\left(\frac{n}{2}\right) + n^2$, a = 7, b = 2, k $= \log_2 7$, $f(n) = n^2$

$$\textbf{Case 1: If } f(n) = O\left(n^{k-\varepsilon}\right) \textbf{ for some constant } \varepsilon > 0, \textbf{then } T(n) = \Theta\left(n^k\right).$$

$$f(n) = O\left(n^{k-\varepsilon}\right) = n^2 = O\left(n^{\log_2 7-\varepsilon}\right)$$

$$let\ \varepsilon = \log_2 7 - 2\ and\ since\ \log_2 7 - 2 > 0,$$

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

$$T(n) = \Theta\left(n^k\right) = \Theta\left(n^{\log_2 7}\right)$$

For $T_B(n) = \alpha T_B\left(\frac{n}{4}\right) + n^2$, a = α, b = 4, k $= \log_4 \alpha$, $f(n) = n^2$. In order to make algorithm B is asymptotically faster than A, algorithm B must have a runtime greater than $\Theta(n^2)$. To achieve this runtime, we will attempt to make the equation satisfy case 1 where:

$$\textbf{Case 1: If } f(n) = O\left(n^{k-\varepsilon}\right) \textbf{ for some constant } \varepsilon > 0, \textbf{then } T(n) = \Theta\left(n^k\right).$$

$$T(n) = \Theta\left(n^k\right) \leq \Theta(\log_2 7),$$

$$k < \log_2 7 = \log_4 \alpha$$

$$\alpha < 49$$

The largest integer value of α for which algorithm B is asymptotically faster than A is when $\boldsymbol{\alpha = 48}$.

Proof:

Substituting $\alpha = 48$ for $T_B(n) = 48T_B\left(\frac{n}{4}\right) + n^2$

$$k = \log_4 48 \text{ and let } \varepsilon > 0.$$

$$T(n) = \Theta\left(n^{\log_4 48}\right) < \Theta\left(n^{\log_4 49}\right)$$