Question 1:

The output is : 1st column is the size, second is the average #operations for naive algorithm and the third is the average #operations for Karatsuba algorithm.

1,1,1
2,7,9
3,34,39
4,34,39
5,145,135
6,145,135
7,148,141
8,148,141
9,595,435
10,595,435
11,598,441
12,598,441
13,613,465
14,613,465
15,616,471

The karatsuba algorithm make less recursive calls than the naive algorithm however it does make twice addition and subtraction operations, hence why it is less efficient to calculate the multiplication of numbers with small size. As for big sizes (larger than 4), the Karatsuba algorithm is more efficient as the gap between the #operations keep increasing as the size increase.

Question 2:

a)

$T(n) = 25\,T\left(\frac{n}{5}\right) + n.$

$a = 25, \quad b = 5, \quad K = \log_5(25) = 2; \quad f(n) = n.$

we can apply the 1st case of the master's thm.

for $\varepsilon = 1$; $\quad f(n) = \cancel{f(n^{2-\varepsilon})} = O(n^{2-\varepsilon}) = O(n).$

$\longrightarrow T(n)$ is $\Theta(n^2)$

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

$a = 2, \; b = 3, \; K = \log_3 2 \simeq 0.63 < 1, \; f(n) = n\log n$

we apply the 3rd case of master theorem.

$n\log n$ is $\Omega(n)$.

i $\quad \Omega(n^{\log_3 2 + \varepsilon}) = \Omega(n)$ for $\varepsilon \simeq 0.73$

ii $\quad \frac{n}{3}\log\left(\frac{n}{3}\right) \leq \frac{1}{3}\, n\log n \quad C = \frac{1}{3}.$ (graphing) \\ last page

$\Longrightarrow T(n)$ is $\Theta(n\log n).$

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

we can apply the second case of the master theorem.

$a = 1, \; b = \frac{3}{4}, \; f(n) = 1, \; K = \log_{\frac{4}{3}} 1 = 0.$

$f(n)$ is $\Theta(n^0 \log^0 n)$, $\cancel{B} = \Theta(1)$ $(P = 0)$

$\Rightarrow T(n)$ is $\Theta(\log^{P+1} n) = \Theta(\log n).$

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

$a = 7,\ b = 3,\ k = \log_3 7 \approx 1.77$

we can apply the 3rd case of The masters thm.

i) for $\varepsilon \approx 1.23$; $f(n)$ is $= \Omega(n^{1.77+\varepsilon}) = \Omega(n^3)$.

ii) $7\left(\frac{n}{3}\right)^3 = \frac{7}{27}n^3 \leq \frac{7}{27}n^3 .\ C = \frac{7}{27} < 1.$

$\Rightarrow T(n)$ is $\Theta(n^3)$.

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

$a = 1,\ b = 2,\ k = \log_2 1 = 0.\ \ f(n) = n(2 - \cos n)$.

None of the cases apply.

i) for the 1st case,

$f(n)$ can't be $O(n^{\log_2 1 - \varepsilon})$ for any $\varepsilon > 0$ because $\log_2 1 = 0$.

ii) for the second case there is no p S.T. $n(2 - \cos n)$ is $\Theta(\log^p n)$.

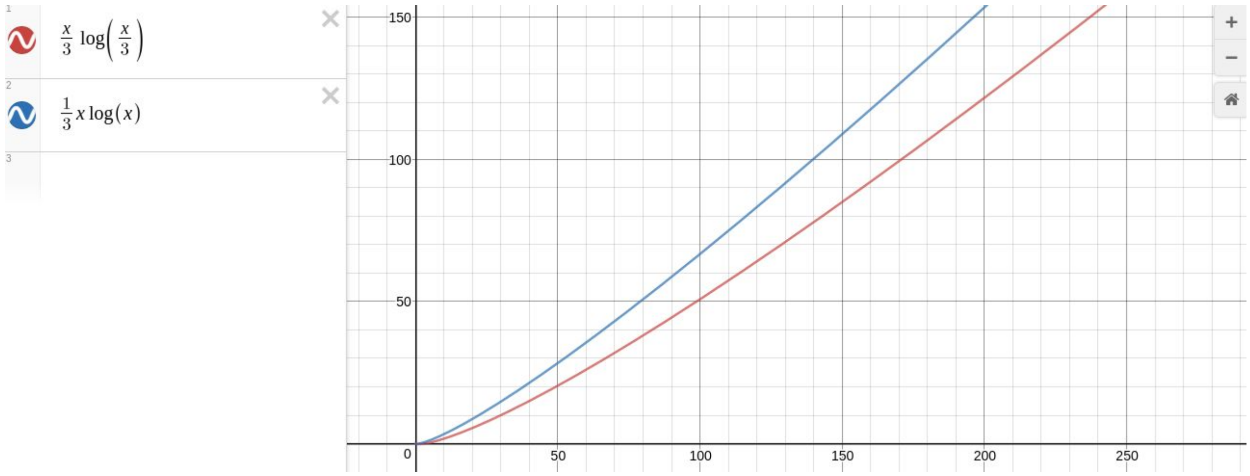
iii) the first condition ~~case~~ can be satisfied, however, the second condition.

$f\left(\frac{n}{2}(2 - \cos\frac{n}{2})\right) \leq C n(2 - \cos n)$ can't be satisfied with $C < 1$. ~~(graph later~~

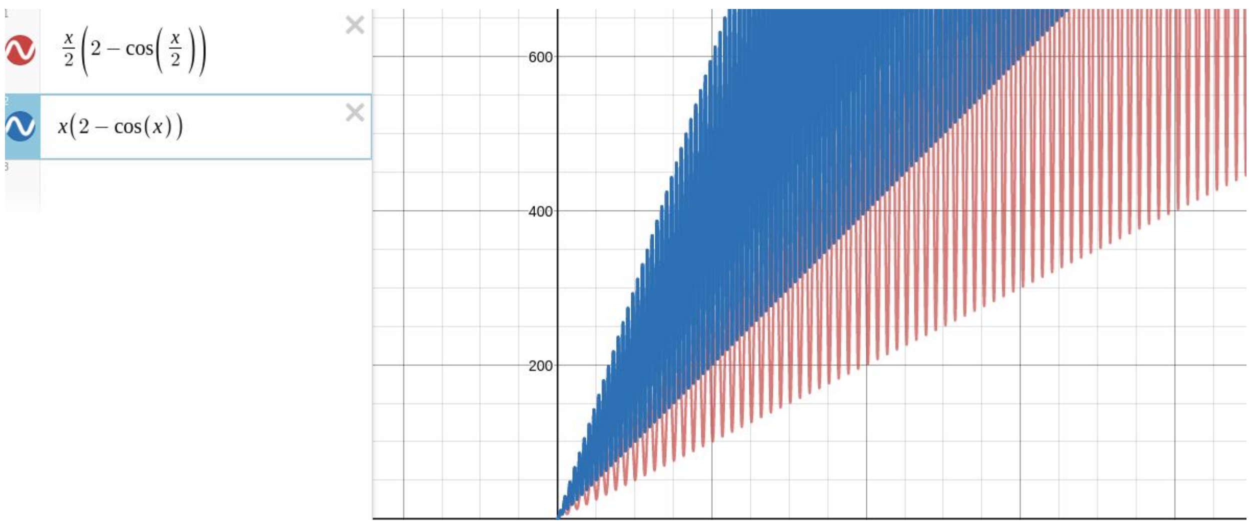
~~because for n, cos n~~

because for some values for n, where n is near its max value, $\cos\frac{n}{2}$ is near its ~~max~~ minimum. hence the multiplier C will be larger > 1.

(graph after this page).



This graph is for the 3rd recursion, (n/3) log (n/3) =< ⅓ (n logn) , C = ⅓ < 1



As for the 5th recursion;

As shown, both graphs intersect even when c =1 so in order to get, n/2(2-cos(n/2)) < C.n(2-cos(n) , C has to be greater than 1.

Question 3:

$$T_A = 7T_A\left(\frac{n}{2}\right) + n^2 \; ; \; T_B(n) = \alpha T_B\left(\frac{n}{4}\right) + n^2$$

first we sölve the time complexity of $T_A(n)$.

$a = 7,\ b = 2,\ K = \log_2 7 \simeq 2.81,\ f(n) = n^2.$

we can apply the first case of master's thm.

for $\varepsilon \simeq 0.81 > 0$; $f(n)$ is $O(n^{\log_2 7 - \varepsilon}) = O(n^2)$.

$\Longrightarrow T_A(n)$ is $\Theta(n^{\log_b a}) = \Theta(n^{2.81})$.

To find the largest α for T_B that is assymptically faster

we set the values equal to each other, we solve α

then we deduct 1 from α.

$$K_A = \log_7 2 = \log_4 \alpha'$$

$$\Longrightarrow \alpha' = 49 \rightarrow \alpha = 49 - 1 = 48.$$

hence The algorithm B runs asymptotically faster than T_A for $\alpha \leq 48$.