k.) a.) $f(x)+g(x)$ is $o(f(x) \times g(x))$ - True

$\therefore \lim_{x \to \infty} \left(\frac{f(x)+g(x)}{f(x)g(x)} \right)$

$= \lim_{x \to \infty} \left(\frac{1}{g(x)} + \frac{1}{f(x)} \right)$

$\therefore f(x)$ and $g(x)$ are monotonically increasing so

$\lim_{x \to \infty} \left(\frac{1}{g(x)} + \frac{1}{f(x)} \right) = 0 + 0 = 0$

b.) $2^x \cdot x^2$ is $o(2.1^x)$ - True

$\therefore \lim_{x \to \infty} \left(\frac{2^x \cdot x^2}{2.1^x} \right) = \lim_{x \to \infty} \left(\left(\frac{20}{21} \right)^x x^2 \right) = 0$

c.) $x^2 \cdot \log x$ is $O(x^2)$ - False

$\therefore$ Assume $x^2 \log x \leq C \cdot x^2$, for $x \geq k$

$\therefore \log x \leq C$, so false since $\log x$ grows monotonically

d.) $x^2 \cdot \log x$ is $O(x^3)$ - True

$\therefore$ Assume $x^2 \cdot \log x \leq C \cdot x^3$, for $x \geq k$

$\therefore \frac{\log x}{x} \leq C$, so true for $k = 1$, $C = 1$

e.) $7x^5$ is $O(12x^4 + 5x^3 + 8)$ - False

$\therefore 12x^4 + 5x^3 + 8$ is $O(25x^4)$

$7x^5 \leq C(25x^4)$, for $x \geq 4$

$\frac{7x}{25} \leq C$, so false since x grows monotonically

5.) a.) $64T(\frac{n}{8}) - n^2 \log n$

$\therefore$ not possible since $-n^2 \log n$ is not asymptotically positive

b.) $T(n) = 4T(\frac{n}{2}) + \frac{n}{\log n}$

$\therefore n^{\log_2(4)} = n^2$

$\therefore \frac{n}{\log n} = O(n^{\log_2(4) - \epsilon})$

$\therefore$ case 1 applies

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

c.) $2^n T(\frac{n}{2}) + n^n$

$\therefore$ not possible since a is not constant

d.) $T(n) = 3T(\frac{n}{4}) + n \log n$

$\therefore n^{\log_4(3)} \approx n^{0.792...}$

$\therefore n \log n = \Omega(n^{\log_4(3) + \epsilon})$

$\therefore \frac{3}{4} n \log(\frac{n}{4}) \leq c n \log n, \quad c = \frac{3}{4}$

$\therefore$ case 3 applies

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

e.) $3T(\frac{n}{3}) + \sqrt{n}$

$\therefore n^{\log_3(3)} = n$

$\therefore \sqrt{n} = O(n^{\log_3(3) - \epsilon})$

$\therefore$ case 1 applies

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

6.) b.) Worst case occurs when the array is already sorted. My function chooses k many elements from the left of the array. This means that every remaining element has to be compared with every pivot such that they all end up in the array to the left of the last pivot. This array is also sorted so the same situation will occur when it is passed back into the quicksort function. (Each array has length $n-k$)