Question 4

a)

False: Assume that $2x^4 is\ O(x^3 + 3x + 2)$. Therefore, there are constants C and k such that:

$$2x^4 \leq C(x^3 + 3x + 2), where\ x \geq k.$$

Dividing by $x^3 + 3x + 2$:

$$\frac{2x^4}{x^3 + 3x + 2} \leq C$$

Dividing LHS by $\frac{x}{x} = 1$:

$$\frac{2x}{1 + 3x^{-2} + 2x^3} \leq C$$

Given that there are no positive powers of LHS' denominator, it will tend towards 1 as x increases. However, the numerator $2x$ will continuously increase as x does meaning LHS will increase continuously as x does. Therefore, there can be no constant C that binds it and $2x^4 is\ not\ O(x^3 + 3x + 2)$.

b)

True: Given that $1 \leq log(x) \leq x \leq x^2 \leq x^3$ for all $x \geq 1$:

$$4x^3 + 2x^2 \cdot log(x) + 1 \leq 4x^3 + 2x^2 \cdot x + x^3$$
$$4x^3 + 2x^2 \cdot log(x) + 1 \leq 4x^3 + 2x^3 + x^3$$
$$4x^3 + 2x^2 \cdot log(x) + 1 \leq 7x^3 \ \ where\ (x \geq 1)$$

So, using the witnesses $C = 6$ and $k = 1$ we can deduce that $4x^3 + 2x^2 \cdot log(x) + 1$ is $O(x^3)$.

c)

True: For $3x^2 + 7x + 1$ to be $\omega(x \cdot log(x))$:

$$3x^2 + 7x + 1 > Cx \cdot log(x), where\ x \geq k\ (k > 0), and\ for\ all\ C > 0$$

Dividing by $x \cdot log(x)$:

$$\frac{3x + 7 + x^{-1}}{log(x)} > C$$

$$\frac{3x}{log(x)} + \frac{7}{log(x)} + \frac{1}{x \cdot log(x)} > C$$

Given that 7 is constant, and x and log(x) are increasing: $\frac{7}{log(x)} + \frac{1}{x \cdot log(x)}$ will tend towards zero as x increases. However, as powers beat logarithms, $\frac{3x}{log(x)}$ will increase continuously as x increases and therefore, there can always be a value for k such that $3x^2 + 7x + 1$is greater than a constant C. So $3x^2 + 7x + 1\ is\ \omega(x \cdot log(x))$.

d)

True: Given that $1 \leq log(x) \leq x \leq x^2$ for all $x \geq 1$:

$$x \cdot log(x) \leq x \cdot x$$
$$x \cdot log(x) \leq x^2$$

Seeing as $x^2 + 4x > x^2$ where $x \geq 0$. It can easily be seen that:

$$3x^2 + 7x + 1 > x \cdot log(x)\ where\ (x \geq 2)$$

So using witnesses $C = 1\ and\ k = 1$ it can be deduced that $x^2 + 4x$ is $\Omega(x \cdot log(x))$.

e)

False: Assume $f(x) + g(x)\ is\ \Theta(\mathrm{f(x)} \cdot \mathrm{g(x)})$. This would imply that $f(x) + g(x)\ is\ \Omega(f(x) \cdot g(x))$, further implying that: $f(x) + g(x) \geq C \cdot f(x) \cdot g(x), where\ x \geq k$. Where C and k are constants such that C > 0 and k > 0 and for all functions f and g.

Let $f(x) = x^3$ and $g(x) = x^2$. The assumption implies:

$$x^3 + x^2 \geq Cx^6$$

As $x \geq k \geq 0$, we can divide by x^6. Dividing by x^6gives us:

$$x^{-3} + x^{-2} \geq C$$

As all the powers on the left hand side are negative, $x^{-3} + x^{-2}$ decreases continuously as x increases. Meaning $x^{-3} + x^{-2}$ cannot be bounded underneath by a constant > 0 so no C > 0 can exist and therefore, $x^3 + x^2$ cannot be $\Omega(x^6)$. Causing a contradiction with the original assumption and proving that $f(x) + g(x)$ is not always $\Theta(\mathrm{f(x)} \cdot \mathrm{g(x)})$.

Question 5

a)

T(n) can be solved with master theorem using the constants: $a = 9, b = 3$ and the function $f(n) = n^2$. Considering the second case of master theorem where $f(n) = \Theta\left(\mathrm{n}^{\log_b a}\right)$:

$$\mathrm{n}^{\log_b a} = n^{\log_3 9} = n^2$$

We can prove $f(n) = O(n^2)$ using the witnesses $C = 1, k = 1$ and $f(n) = \Omega(n^2)$ using the same witnesses as $f(n) = n^2 = 1 \cdot n^2$, and therefore:

$$f(n) = \Theta(\mathrm{n}^2) = \Theta\left(\mathrm{n}^{\log_b a}\right)$$

So, using the second case of master theorem:

$$T(n) = \Theta\left(\mathrm{n}^2 log(n)\right)$$

b)

T(n) can be solved with master theorem using the constants: $a = 4, b = 2$ and the function $f(n) = 100n$. Considering the first case of master theorem where $f(n) = O(n^{\log_b a - \epsilon})$, for some $\epsilon > 0$:

$$n^{\log_b a - \epsilon} = n^{\log_2 4 - \epsilon} = n^{2 - \epsilon}$$

If we take $\epsilon = 1$, then $n^{2 - \epsilon} = n^{2 - 1} = n$, so using the witnesses $C = 100, k = 1$, we can show that $f(n) = 100n = O(n)$, and therefore:

$$f(n) = O(n^{\log_b a - \epsilon})$$

So, using the first case of master theorem:

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

c)

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

In order for a recurrence to be resolved using master theorem, it must be of the form:

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

However, this recurrence cannot be written in this form as in place of the constant a, there is a non-constant expression: 2^n. Therefore, this recurrence cannot be resolved using master theorem.

d)

T(n) can be resolved with master theorem using the constants $a = 3, b = 3$ and the function $f(n) = cn$, where c is a constant.

Considering the second case of master theorem where $f(n) = \Theta\left(n^{\log_b a}\right)$:

$$n^{\log_b a} = n^{\log_3 3} = n$$

If we use as our witnesses, $C = c, k = 1$ then:

$$f(n) = cn = cn^{\log_3 3} = cn^{\log_b a}, for\ all\ x > 1$$

Given that $f(n) = cn^{\log_b a}$, it is clear that $f(n) = O(n^{\log_b a})$ and $f(n) = \Omega(n^{\log_b a})$, so $f(n) = \Theta\left(n^{\log_b a}\right)$ and, through master theorem:

$$T(n) = \Theta\left(n^{\log_b a} \cdot log(n)\right) = \Theta(n \cdot log(n))$$

e)

$$T(n) = 0.99T\left(\frac{n}{7}\right) + \frac{1}{n^2}$$

In order for a recurrence to be resolved using master theorem, it must be of the form:

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

Where $a \geq 1, b > 1$. However the value of $a = 0.99$ means $a < 1$, meaning $T(n)$ cannot be resolved using master theorem.

Question 6

b)

Given that on an input of size n selection sort will make $\frac{n(n-1)}{2}$ and be of time complexity $O(n^2)$ regardless of the contents of n, it can essentially be ignored when searching for a difficult input. Furthermore, the MergeSort algorithm makes no comparisons and divides the function in have until each section reaches a length of 4, again, regardless of the contents of the input.

The only way in which the contents of the input effects the efficiency is the Merge function. When given inputs where every item of left is smaller than every item in right it will compare each item from the left to the right and end up with an empty left list meaning it can simply concatenate right onto the end of result. This would best case input to Merge.

The worst case input for Merge would logically be the opposite: An input where the indexes of left and right alternate in size, for example: left = [1, 3, 5, 7], right = [2, 4, 6, 8]. This input would result in left and right decreasing in size at essentially the same rate resulting in the maximum amount of comparisons and the full iteration through left and right. By this logic, the worst case input for the merge-selection hybrid would be one where, when split into sections the n^{th} item in the second section is bigger than the n^{th} item in the smaller section but smaller than the $(n + 1)^{th}$ item like so: [1, 3, 5, 7, 2, 4, 6, 8].