P5.2.

a) The brute force implementation of multiplying two large integers with n bits each is a time complexity of $\Theta(n^2)$. The algorithm works by multiplying each bit of "a" with each bit of "b". Each time there is a multiplication with a bit from "a" with each bit from "b", the result is bit-shifted by the position of the bit in a. The result is calculated by summing up all the results from each bit multiplication.

$$
\begin{array}{r}
\times \quad 1011 \\
1110 \\
\hline
0000 \\
1011 \\
+ \quad 1011 \\
1011 \\
\hline
10011010
\end{array}
$$

Since each addition and bit shift is considered to be done in linear time, the overall time complexity is $\Theta(n^2)$. Multiplying each bit from "a" with each bit from "b" is $T(n) = \Theta(n^2)$. Addition of n n-bit numbers takes $\Theta(n^2)$ time

$$T(n) = \Theta(n^2) + \Theta(n^2) = 2\,\Theta(n^2) = \Theta(n^2)$$

b) Suppose $x = 1234$ then

$x = 1234 =$

$= 12 \times 10^2 + 34$

$= 1200 + 34$

$= 1234$

From this we can derive that $x = a \times 10^{?} + b$, where a = left half, b = right half and n = total digits.

Now for another term "Y" it would be

$y = c \times 10^{\frac{n}{2}} + d$

So $x \cdot y =$

$= (a \cdot 10^{\frac{n}{2}} + b)(c \cdot 10^{\frac{n}{2}} + d)$

$= ac \cdot 10^{n} + (ad + bc) \cdot 10^{\frac{n}{2}} + bd$

~~We will simplify~~ $\cancel{(ad+bc)}$

$ad + bc = (a+b)(c+d) - ac - bd =$

$= ac + ad + bc + bd - ac - bd =$

$= ad + bc$

Now implementing algorithm

```
float multiply(x, y)
n = max(bits in x, bits in y)
IF (n == 1)
    return x*y
else
    a = left half of x
    b = righ half of x
    c = left half of y
```

d= right half of y

AC= multiply (a, c)
BD = multiply (b, d)
ADBC = multiply (a+b, c+d)

return $(AC * 10^n + (ADBC - AC - BD) \cdot 10^{\frac{n}{2}}) + BD)$.

c) ~~So~~ Now the multiplication is just bit-shifting in the formula so from that we can easily say our complexity is.

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

e) $T(n) = 3T(n/2) + \theta(n)$

$a = 3$
$b = 2$

$$n^{\log_2 3} = n^{1.58}$$

~~$f(n) = \Theta(n^{...})$~~

$f(n) = O(n^{\log_2 3 - \epsilon}) = O(n^{1.58 - \epsilon})$ where $\epsilon \approx 0.58$

$$T(n) = \Theta(n^{1.58})$$