Problem 2:

T(n)=T(n/2)+3 if n>1 and T(1)=2

To solve this recurrence, I will assume that all division is integer division.

Now, suupose n is a power of 2, say, $n = 2^m$.

If so, what we will have is:

$$T(2^m) = T(\frac{2^m}{2})$$

The above expression, however, can be re-written recursively as:

$$T(2^{m-1}) + 3$$
$$T(2^{m-2}) + 3 + 3$$
$$T(2^{m-3}) + 3 + 3 + 3$$

textSo, for all k, we can claim that:

$$T(2^{m-k}) + 3k$$

textAnd if we apply this reasoning to $k = m$, we get:

$$T(2^{m-m}) + 3m$$

And this results in:

$$2 + 3m$$

Now, since I have substituted 2^m for n, we must take the log of both sides of the equation to get the true value of m:

$$n = 2^m$$

$$log(n) = log(2^m)$$

$$log(n) = m$$

Therefore,

$$T(n) = 2 + 3log_2 n$$

To determine the complexity, we can use the master theorem:

For this function, we have:

$$a = 1, b = 2, c = 0$$

$$log_b a = log_2 1 = 0$$

Therefore,

$$T(n) = \theta(n^0 logn)$$

Or,

$$\theta(logn)$$