These problems concern divisibility in base-10. Let $N$ be an $n$-digit base-10 number,

$$N = D_n D_{n-1} \ldots D_2 D_1. \quad (1)$$

- Argue that $N$ can be checked for divisibility by 2 in constant time. Do you even have to read all the digits in?
- Argue that for any integer $k \geq 0$, the following holds:

$$10^k \equiv 1 \pmod{3}. \quad (2)$$

- Argue from this that if $N$ is divisible by 3, the sum of the digits $D_n + D_{n-1} + \ldots + D_2 + D_1$ must also be divisible by 3, and vice versa.
- Use this fact to construct an algorithm for checking divisibility by 3 (in base-10). What are the base cases?
- Estimate the big O complexity of summing the digits of an $n$-digit number. For an $n$-digit number $N$, how big can the digit-sum be?
- Write a recurrence bounding the complexity of checking for divisibility by 3 according to this algorithm.
- How does this algorithm compare, in terms of complexity, to simply dividing $N$ by 3 and checking the resulting remainder?
- Show that for any $k$, 

$$2^k \pmod{3} = \begin{cases} 
1 & \text{if } k \text{ is even} \\
2 & \text{if } k \text{ is odd}
\end{cases}, \quad (3)$$

and use this to suggest an algorithm for checking divisibility by 3 in base-2. What is its complexity?