CSC236 tutorial exercises, Week #8
best before Thursday evening

These exercises are intended to give you some practice proving bounds on recurrences, and proving correctness of recursive programs.

1. Examine the recurrence $R(n)$ below.

$$R(n) = \begin{cases} 0 & \text{if } n = 1 \\ n + 3R(\lceil n/3 \rceil) & \text{if } n > 1 \end{cases}$$

   Assume that for all $k \in \mathbb{N}, R(3^k) = k3^k$. Prove that $R \in \Theta(n \lg n)$.

2. Read over the code for `decimal_to_binary` below:

```
def decimal_to_binary(n: int) -> str:
    """
    Return binary string representing n.

    precondition: n is a natural number.

    >>> decimal_to_binary(0)
    '0'
    >>> decimal_to_binary(5)
    '101'

    postcondition: returns binary string representing
    n with no leading zeros (except if n == 0).
    """
    if n < 2:
        return str(n)
    else:
        return decimal_to_binary(n // 2) + decimal_to_binary(n % 2)
```

   Use the technique from week 7 notes to prove that the precondition implies termination and the postcondition, **or** find a counter-example.