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.

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

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

2. Read over the code for \texttt{decimal_to_binary} below:

```python
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'
    """
    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.