Activity: Big-O Definitions

For a non-negative function $T(n)$, $T(n) \in O(f(n))$ iff there exist positive constants c and n_0 such that $T(n) \leq c * f(n)$ for all $n > n_0$	$if \lim_{n\to\infty} \frac{f(n)}{g(n)} \begin{cases} \in [0,\infty), & f(n) \in O(g(n)) \\ \in (0,\infty), & f(n) \in \Theta(g(n)) \\ \in (0,\infty], & f(n) \in \Omega(g(n)) \end{cases}$	
$T(n) \in \Theta(f(n))$ iff $T(n) \in O(f(n))$ and $T(n) \in \Omega(f(n))$	**L'Hôpital's Rule** If $\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = 0 \; or \pm\infty$ and $f'(n)$ and $g'(n)$ exist, then $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(n)}{g'(n)}$	
For a non-negative function $T(n)$, $T(n) \in \Omega(f(n))$ iff there exist positive constants c and n_0 such that $T(n) \geq c * f(n)$ for all $n > n_0$		

Function	Derivative
c	0
n	1
n^k	$k * n^{k-1}$
$\log_2(n)$	$\frac{1}{n * \ln(2)}$
2^n	$2^n * \ln(n)$
$\ln(n)$	$\frac{1}{n}$
e^n	e^n

1. For the following questions, fill in the **most appropriate** Big-O relation (O/Ω/Θ):

$25n + 3 \in$ ____(n)

$n^2 + 3n \in$ ____(n)

$n * \log(n) \in$ ____(n^2)

$2^n \in$ ____(1)

$n^2 + \log(n) \in$ ____$(\log(n))$

$n^2 + \log(n) \in$ ____(n^2)

2. Use the **non-limit** definition (find constants *c* and n_0) to demonstrate that:

$$5n + 2 \in \Omega(n)$$

3. Use the **non-limit** definition (find constants *c* and n_0) to demonstrate that:

$$5n^3 + 4n^2 \in \Omega(n^2)$$

4. Use the **limit definition** to demonstrate that:

$$5n^3 + 4n^2 \in \Omega(n^2)$$

5. Use the **limit definition** to determine the relationship between f(n) and g(n) (show every step of your derivation):

$$f(n) = 5n^2 + n, \qquad g(n) = n^2 + \log(n)$$

Circle all that are true: $f(n) \in \Omega(g(n))$ $f(n) \in \Theta(g(n))$ $f(n) \in O(g(n))$

6. Analyze the following algorithm and answer the questions. Count the **underlined code** as your basic operation.

```
public static boolean mystery(int[] numbers) {
    for (int i = 0; i < numbers.length - 1; i++) {
        for (int j = i + 1; j < numbers.length; j++) {
            if (numbers[i] == numbers[j]) {
                return true;
            }
        }
    }
    return false;
}
```

In English, what does this method do or accomplish? ____

Give an input for n = 5 that would result in a best case runtime: `numbers = {   ,   ,   ,   ,   }`

Give an input for n = 5 that would result in a worst case runtime: `numbers = {   ,   ,   ,   ,   }`

Fill in the blanks with the most appropriate function:

Best Case: Θ(_______)

Worst Case: Θ(_______)

Average Case: Θ(_______) ← Take a guess here!