Part II: Analysis Questions

For this, you will solve problems related to efficiency, and the Big O asymptotic notation. **SHOW YOUR WORK.**

The definition of Big O

Let f(n) and g(n) be functions mapping positive integers to positive real numbers.
We say that f(n) is O(g(n)) if there is a real constant $c > 0$ and an integer constant $N \geq 1$ such that:

$$\mathbf{f(n) \leq c * g(n), \ \ for\ n \geq N}$$

Note that the constants c and n_0 are *not unique*. You just have to find a *c* and an *N* that satisfies the definition of Big O

3.1 Given a time function, $T(n) = 5n^2 + n + 3$, find constants **c** and **N** that prove that the big O of the growth function T(n) is n^2

$T(n) = 5n^2 + n + 3$

$5n^2 + n + 3 <= 5n^2 + n^2 + 3n^3$

$5n^2 + n + 3 \leq= 9n^2$

This is on the form $c * g(n)$ proves $O(n^2)$: $c = 9$

It's true for : $n_0 = 1$

3.2 Find the Big O of $T(n) = 2n^4 + n^3 + 5n + 5$. Justify your answer by finding constants c and N

$T(n) = 2n^4 + n^3 + 5n + 5$

$2n^4 + n^3 + 5n + 5 <= 2n^4 + n^4 + 5n^4 + 5n^4$

$2n^4 + n^3 + 5n + 5 <= 13n^4$

We guess $O(n^4)$

$c = 13$ and $n_0 = 1$

3.3 Find the Big O of the following code:

```
for(int i = 0; i < n/5; i++)
{
     for(int j = 0; j < n/2; j++)
     {
          System.out.println("Hello!");
     }
}
```

Complexity of this code is $O(n^2)$

Two loops that iterate n/5 times and n/2

$\frac{n}{5} * \frac{n}{2} = \frac{n^2}{10}$

Space complexity is $O(1)$ there's no variable that grow with input

You do not have to find a T(n), or c and N. Just give the Big O, and informally justify your answer.