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 ≥ \_1 such that:  **f(n) ≤ \_c \* g(n), for n ≥ \_N**  **Note that the constants c and n0 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) = 3n2 + 2n + 1, find constants **c** and **N** that prove that the big O of the growth function T(n) is n2

The derivative proves it

3.2 Find the Big O of T(n) = 2n4 + n2 + 8n + 2. Justify your answer by finding constants c and N

10n^7

3.3 Find the Big O of the following code:

for(int i = 0; i < n/4; i++) {

for(int j = 0; j < n/2; j++) {

System.out.println(“Hello!”);

}

}

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

log4(n^2) + log(n^2) + 1