1. **f(x) = 3\*N + log3(N)**

**3\*N + log3(N) <= 4N**

log3(N) <= N  
 **n0 = 1**, since log3(N) = 0 (meaning 3^0 = 1) and N = 1, when **c = 4**  
 **g(x) = N**

> Critical concepts: Does the student make an *appropriately* close guess? Does the student *solve* for a reasonably low n0? As long as a reasonable attempt is made, credit will be given. (If not, we will clearly indicate why)

2. **f(x) = 6(N2\*(log2(N) + 1))**

**6N2log2(N) + 6N2 <= 12\*N2\*(log2(N))** > Trick: Use c as 2x the factor so it cleanly cancels!

6N2 <= 6\*N2\*(log2(N)) > Notice how this dividing out 6N2 easier!

1 <= log2(N)

**n0 = 2**, since log2(2) = 1 (meaning 2^1 = 2) and N = 2, when **c = 6**

**g(x) = N2\*(log2(N))**

> Critical concepts: Does the student make an *appropriately* close guess? Does the student *solve* for a reasonably low n0? As long as a reasonable attempt is made, credit will be given. (If not, we will clearly indicate why)

3.

4.

5.

Prob5.h

Prob5( const std::string& netID\_in, const int& nd\_ID\_in, const double& GPA\_in, const std::string& dorm\_in);

Prob5.cpp

Prob::Prob5( const std::string& netID\_in, const int& nd\_ID\_in, const double& GPA\_in, const std::string& dorm\_in)

: netID(netID\_in), nd\_ID(nd\_ID\_in), GPA(GPA\_in), dorm(dorm\_in) {}

6.

void point\_private\_vals() const;

int get\_int() const;

double add\_and\_return( const double& double\_in ) const; // Note the use of *only* in the problem statement