Problem Set 1

Assigned: May 27
Due: June 3

Problem 1

For each of the following pairs of functions \( f(n) \) and \( g(n) \), state whether \( f \) is \( o(g) \), \( f \) is \( \Theta(g) \); or \( g \) is \( o(f) \). (Exactly one of these is true in all cases.)

a. \( f(n) = n^{10}; g(n) = 2^{n/2} \).

b. \( f(n) = n^{3/2}; g(n) = n \log^2(n) \).

c. \( f(n) = \log(n^3); g(n) = \log(n) \).

d. \( f(n) = \log(3^n); g(n) = \log(2^n) \).

e. \( f(n) = (\log(n))^3; g(n) = (\log(n)) \).

f. \( f(n) = 2^n; g(n) = 2^{n/2} \).

g. \( f(n) = n^2; g(n) = (n/2)^2 \).

h. \( f(n) = n^2; g(n) = (n + 2)^2 \).

i. \( f(n) = 2^n; g(n) = 2^{n+2} \).

j. \( f(n) = n!; g(n) = (n + 2)! \).

Problem 2

The following two functions both compute the same thing. They take as arguments two arrays \( A \) and \( B \) and return TRUE if every element of \( A \) is less than every element of \( B \).

```cpp
AllLessThan1(int[] A, B) return bool {
    for (i=1 to |A|)
        for (j=1 to |B|)
            if (A[i] >= B[j]) return FALSE;
    return TRUE;
}

AllLessThan2(int[] A, B) return bool {
    largeA = A[1]
    for (i = 2 to |A|)
        if (A[i] > largeA) largeA = A[i];
    for (j = 1 to |B|)
        if (largeA >= B[j]) return FALSE;
    return TRUE;
}
```
A. Give the asymptotic worst-case running time of each algorithm as a function of $|A|$ and $|B|$. When does the worst case occur?

B. Give the asymptotic best-case running time of each algorithm as a function of $|A|$ and $|B|$. When does the best case occur?

C. Design an algorithm whose best-case running time is as good as the best-case for both of these algorithms, and whose worst-case running time is as good as the worst-case both of these algorithms.