CS 325 Week 2- Practice Problems

Problem 1: How many times as a function of n (in Θ form), does the following PHP function echo "Print"? Write a recurrence and solve it.

```
function foo( $n )   {
        if ($n > 1) {
            foo($n/2);
            foo($n/2);
            foo($n/2);
            foo($n/2);
            for ($i = 1; $i <= $n; $i += 1) {
                    echo " Print  ".$i." <br> ";
            }
            echo " <br>";
         }  else {
             return 1;
         }
}
```

Problem 2: Give the asymptotic bounds for T(n) in each of the following recurrences. Make your bounds as tight as possible and justify your answers.

a) $T(n) = 2T\left(\frac{n}{4}\right) + n$

b) $T(n) = T(n-1) + n^2$

CS 325 Week 2- Practice Problems

Problem 3:
Complete the following divide-and-conquer algorithm to determine if all integers in an array A are equal. The initial call would be allEqual(A,0,A.length-1).

(Yes, there is an easy iterative algorithm for this problem. The goal here is to provide practice with the design and analysis of divide-and-conquer algorithms.)

```
boolean allEqual ( int A[], int p, int r){
        if (p == r)
                return true;
        if (A[p] != A[r])
                return false;

        //take it from here




}
```

Write a recurrence relation for your algorithm and then solve it to obtain the worst-case asymptotic time complexity for your algorithm.

Problem 4: For the following program fragment compute the worst-case asymptotic time complexity (as a function of n).

```
for (i=0; i<=n-1; i++)
        loop body

for (i=0; i<=n-1; i++)
        for (j=i; j<= n-1; j++)
                loop body
```