Name: **Christopher Cousillas** Date: 10/9/2019

Pledge: I pledge my honor that I have abided by the Stevens Honor System.

Use the Master Theorem to find the complexity of each recurrence relation listed below.

1. a = 1 b = 2 d=2 a\_\_b^d 1 < 4  
   Complexity:
2. Complexity: **)** a = 4 b = 2 d =2 a\_\_b^d 4=4
3. Complexity**:** a = 3 b = 3 d=1/2 a\_\_b^d 3>3^(1/2)

For each function below, write the recurrence relation for its running time and then use the Master Theorem to find its complexity.

1. **int** f(**int** arr[], **int** n) {

**if** (n == 0) {

**return** 0;

}

**int** sum = 0;  
 **for** (**int** j = 0; j < n; ++j) {  
 sum += arr[j];  
 }  
 **return** f(arr, n / 2) + sum + f(arr, n / 2);  
}  
  
Recurrence:

Complexity:

1. **void** g(**int** n, **int** arrA[], **int** arrB[]) {

**if** (n == 0) {

**return**;

}

**for** (**int** i = 0; i < n; ++i) {  
 **for** (**int** j = 0; j < n; ++j) {  
 arrB[j] += arrA[i];

}

}  
 g(n / 2, arrA, arrB);  
}

Recurrence:

Complexity: