Name: Makena Williamson Date: 10/20/2022

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. $T(n) = T\left(\frac{n}{2}\right) + n^2$
   Complexity: $\Theta(n^2)$

   $d=2$, $a=1$, $b=2$
   $a < b^d$
   $a < 2^2$ ✓ $\rightarrow \Theta(n^b)$

2. $T(n) = 4T\left(\frac{n}{2}\right) + n^2$
   Complexity: $\Theta(n^2 \lg n)$

   $d=2$, $a=4$, $b=2$
   $a = b^d$
   $4 = 2^2$ ✓ $\rightarrow \Theta(n^d \log_b n)$

3. $T(n) = 3T\left(\frac{n}{3}\right) + \sqrt{n}$
   Complexity: $\Theta(n)$

   $d=\frac{1}{2}$, $a=3$, $b=3$
   $a > b^d$
   $3 > \sqrt{3}$ ✓ $\rightarrow \Theta(n^{\log_b a})$
   $n^{\log_3(3)} = n^1 = n$

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

4.
```
int f(int arr[], int n) {
    if (n == 0) {
        return 0;
    }
    int sum = 0;
    for (int j = 0; j < n; ++j) { //n
        sum += arr[j];
    }
    return f(arr, n / 2) + sum + f(arr, n / 2);
}
```

$a=2$, $b=2$, $d=1$
$a = b^d$
$2 = 2^1$ ✓

Recurrence: $T(n) = 2T\left(\frac{n}{2}\right) + n$
Complexity: $\Theta(n \lg n)$

5.
```
void g(int n, int arrA[], int arrB[]) {
    if (n == 0) {
        return;
    }
    for (int i = 0; i < n; ++i) { //n
        for (int j = 0; j < n; ++j) { //n
            arrB[j] += arrA[i];
        }
    }
    g(n / 2, arrA, arrB);
}
```

$a=1$, $b=2$, $d=2$
$a < b^d$
$1 < 2^2$ ✓

Recurrence: $T(n) = T\left(\frac{n}{2}\right) + n^2$
Complexity: $\Theta(n^2)$