Question 3

3. Analyze the big-oh run time for each function with respect to parameter n:

```
void A (int n)   // O(1)
{
        if (n < 10)          // O(1)
                time++;
        else
        {
                for(int i = 0; i < n; i++) // O(n) = O(n) * 1
                        time++;
                A(n – 2);  → T(n-2)
        }
}
```

$O(n^2)$

$T(n) = T(n-2) + cn$ ← got from for loop

$T(n)$

$T(n-2) = T(n-2-2) + c(n-2)$
$T(n-4)$

$T(n-4) = T(n-6) + c(n-4)$

n	$i=0$	$c(n-2i)$
$c(n-2)$	$i=1$	$c(n-2i)$
$c(n-4)$	$i=2$	$c(n-2i)$
⋮		
1 // base case		

Work done → i represents the level we're on

$T(n-2i) = 1$

How many, levels?

$n - 2i = 1$

$2i = 1 + n$

$i = \frac{1}{2} + \frac{n}{2}$

Solve the runtime

Work per level × length of tree → $n * \frac{n}{2} = \frac{1}{2}n^2$

$O(n^2)$

Proof w/ summation:

$$\sum_{i=0}^{n/2} c(n-2i) = c[n + (n-2) + (n-4) \dots 1]$$

$n/2$