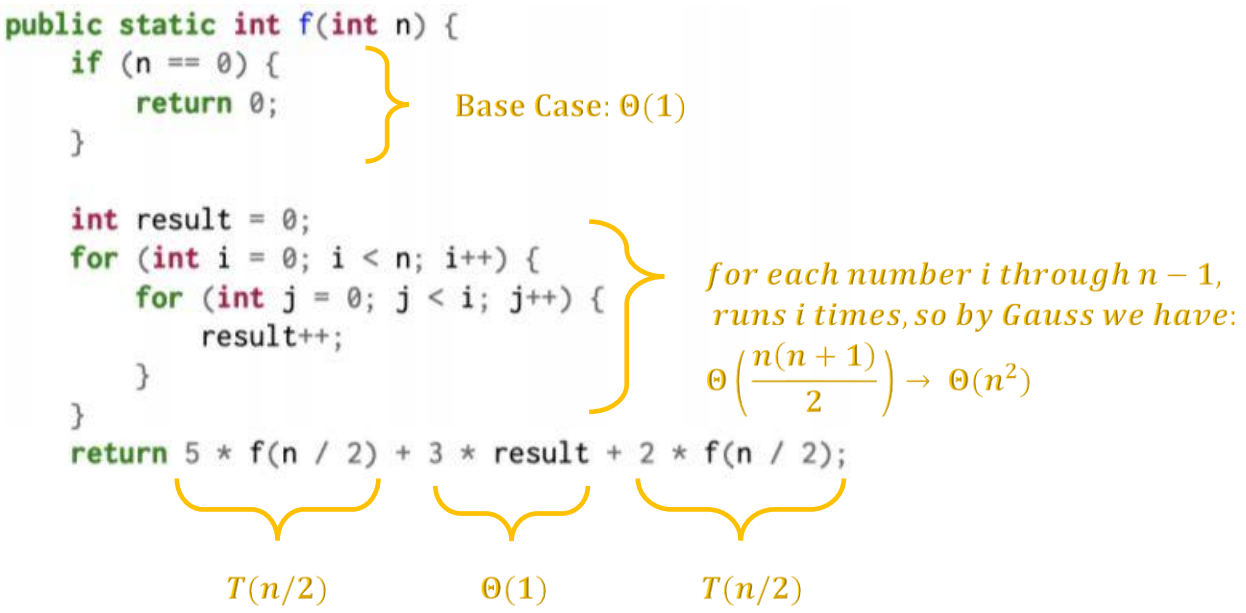
Jason Caplan
Design and Analysis of Algorithms
Professor Leff
HW3

6)



1. $$T(n) = \begin{cases} \Theta(1) & n = 0 \\ 2T\left(\frac{n}{2}\right) + \Theta(n^2) + \Theta(1) & n \geq 1 \end{cases}$$

2. $$W(n) = \begin{cases} 0 & n = 0 \\ 3\left(\frac{n(n+1)}{2}\right) + 7W\left(\frac{n}{2}\right) & n \geq 1 \end{cases}$$

7)

```
public int test(int n) {
    IDictionary<Integer, Integer> dict = new AvlDictionary<>();
    populate(n, dict);
    int counter = 0;
    for (int i = 0; i < n; i++) {
        counter += dict.get(i);
    }
    return counter;
}

private void populate(int k, IDictionary<Integer, Integer> dict) {
    if (k == 0) {
        dict.put(0, k);
    } else {
        for (int i = 0; i < k; i++) {
            dict.put(i, i);
        }
        populate(k / 2, dict);
    }
}
```

$P(n)$

$O(n \times log(n))$

Base Case: $O(log(n))$

$O(k \times log(n))$

$P(k/2)$

1. $T(n) = O(n \times \log(n)) + P(n)$

2. $$P(k) = \begin{cases} O(\log(n)) & n = 0 \\ O(k \times \log(n)) + P\left(\frac{k}{2}\right) & n \geq 1 \end{cases}$$