

6)

```

public static int f(int n) {
    if (n == 0) {
        return 0;
    }

    int result = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++) {
            result++;
        }
    }
    return 5 * f(n / 2) + 3 * result + 2 * f(n / 2);
}
    
```

Base Case: $\Theta(1)$

for each number i through $n - 1$, runs i times, so by Gauss we have:
 $\Theta\left(\frac{n(n+1)}{2}\right) \rightarrow \Theta(n^2)$

$T(n/2)$ $\Theta(1)$ $T(n/2)$

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

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

7)

```

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

private void populate(int k, IDictionary<Integer, Integer> dict) {
    if (k == 0) {
        dict.put(0, k); } Base Case: O(log(n))
    } else {
        for (int i = 0; i < k; i++) {
            dict.put(i, i); } O(k × log(n))
        }
        populate(k / 2, dict); } P(k/2)
    }
}

```

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

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