1. For each of the following recursive functions:

- Describe what the function computes (careful, some of these are tricky!)
- Give a recurrence relation that describes the running time of the function (Give both base and recursive cases)
- Solve the recurrence to get a \( \Theta \) running time for the function. Use either the repeated substitution method, or the recursion tree method (which is essentially the same as the repeated substitution method, just a little more graphical)
- Prove your bounds (upper and lower) are correct using the substitution method (you may need to refer to the “subtracting lower order terms” technique discussed in class on Monday)

(a) int recursive3(int n) {
    if (n > 1) {
        return 2 * recursive3(n-1);
    } else
        return n;
}

(b) int recursive4(int n) {
    int no_op;
    if (n > 1) {
        for (i=1; i<=n; i++)
            no_op++;
        return recursive4(n/2) * recursive4(n/2);
    } else
        return 1;
}

(c) int recursive5(int n) {
    int no_op;
    if (n > 0) {
        for (i=1; i<=n; i++)
            no_op++;
        return recursive4(n-1) + 2;
    } else
        return n;
}