Problem: Give a time-complexity analysis of an algorithm whose recursive equation is

\[ T(n) = T(n - 1) + 2n - 1 \]

where \( T(0) = 0 \).

Solution (Method 1-Don’t simplify at each iteration): It may be helpful to write down the first several iterations independently before tackling the top-level iteration. In particular, notice that to evaluate the above expression, we need to determine \( T(n - 1) \):

\[ T(n - 1) = T(n - 1 - 1) + 2(n - 1) - 1 = T(n - 2) + 2(n - 1) - 1 \]

and to evaluate this expression we need to determine \( T(n - 2) \):

\[ T(n - 2) = T(n - 2 - 1) + 2(n - 2) - 1 = T(n - 3) + 2(n - 2) - 1. \]

Now we consider the top-level equation, with the often-times tricky goal being to determine an expression for \( T(n) \) at the \( k^{th} \) iteration of the recursion. Using the above we get

\[
T(n) = T(n - 1) + 2n - 1 \\
= [T(n - 2) + 2(n - 1) - 1] + 2n - 1 \\
= T(n - 2) + 2(n - 1) + 2(n - 0) - 2 \\
= [T(n - 3) + 2(n - 2) - 1] + 2(n - 1) + 2(n - 0) - 2 \\
= T(n - 3) + 2(n - 2) + 2(n - 1) + 2(n - 0) - 3 \\
= \cdots \\
= T(n - k) + 2(n - k + 1) + 2(n - k + 2) + \cdots + 2(n - 0) - k
\]

Given this expression, we need to determine when this recursion will terminate. Notice that from the given we have \( T(0) = 0 \), and this implies that the recursion will stop when \( k = n \) (and thus when we evaluate \( T(0) = 0 \) in the above). Substitute \( n = k \) into the above expression to get

\[
T(n) = T(n - k) + 2(n - k + 1) + 2(n - k + 2) + \cdots + 2(n - 0) - k \\
= T(0) + 2(1) + 2(2) + \cdots + 2(n) - n \\
= 0 + 2(1 + 2 + \cdots + n) - n \\
= 2 \left( \frac{n(n + 1)}{2} \right) - n \\
= n^2 - n \\
= O(n^2).
\]

where we have used that \( \sum_{k=1}^{n} k = \frac{n(n + 1)}{2} \).
Solution (Method 2-Simplify at each iteration): Although you must be careful, sometimes it is possible to find a closed-form expression for $T(n)$. Notice in the sequence of iterations in the previous solution, we could have simplified at each step to obtain:

\[ T(n) = T(n - 1) + 2n - 1 \]
\[ = [T(n - 2) + 2(n - 1) - 1] + 2n - 1 \]
\[ = T(n - 2) + 4n - 4 \]
\[ = [T(n - 3) + 2(n - 2) - 1] + 4n - 4 \]
\[ = T(n - 3) + 6n - 9 \]
\[ = T(n - 4) + 2(n - 3) - 1 + 6n + 9 \]
\[ = T(n - 4) + 8n - 16 \]
\[ = \ldots \]
\[ = T(n - k) + (2k)n - k^2 \]

In the same way, notice that this iteration stops when $n = k$, so substituting this into the previous equation gives

\[ T(n) = T(0) + 2n^2 - n^2 = n^2 = O(n^2) \]

which is the result that we obtained using the previous method.