Here’s a quick summary of the way to get an exact statement count for a piece of code:

1. Start by computing the number of times the body of each loop is executed. For simplicity, don’t try to sum across multiple levels of loop immediately; rather, write the count for each loop in terms of the next outermost loop’s counter variable.

2. Each simple statement in the body of a loop executes as often as the loop body. The loop test itself executes one more time than the body.

3. Assign statement counts from the innermost loop outward.

4. Finally, sum up the counts for all statements. Counts that involve an induction variable (i.e. the counter for some loop) should be summed over all values of that variable, so that the result is given only in terms of the input size.

The following example is our CheckAll procedure for closest pair.
Let $P[0 \ldots n-1]$ be an array of points.

$$\text{CHECKALL}(P)$$

- $\text{minDist} \leftarrow \infty$
- $j \leftarrow 0$
- while $j \leq n - 2$ do
  - $k \leftarrow j + 1$
  - while $k \leq n - 1$ do
    - $d \leftarrow \text{distance}(P[j], P[k])$
    - if $d < \text{minDist}$
      - $\text{minDist} \leftarrow d$
    - $k++$
  - $j++$
- return $\text{minDist}$

Note that the inner loop runs for $k = j + 1 \ldots n - 1$, or $(n - 1) - (j + 1) + 1 = n - j - 1$ iterations, while the outer loop runs for $j = 0 \ldots n - 2$, or $(n - 2) - 0 + 1 = n - 1$ iterations.

Adding up all the statements to get the total count $S(n)$, we have

$$S(n) = 3 + 3n - 2 + \sum_{j=0}^{n-2} (n - j) + 4 \sum_{j=0}^{n-2} (n - j - 1)$$

$$= 3n + 1 + (n + n - 1 + \ldots + 2) + 4(n - 1 + n - 2 + \ldots + 1)$$

$$= 3n + 1 + \left[ \frac{n(n + 1)}{2} - 1 \right] + 4 \left[ \frac{(n - 1)n}{2} \right]$$

$$= \frac{5}{2} n^2 + \frac{3}{2} n.$$

The most important thing to remember to make the summing easier is the formulas for sums of arithmetic and related progressions.