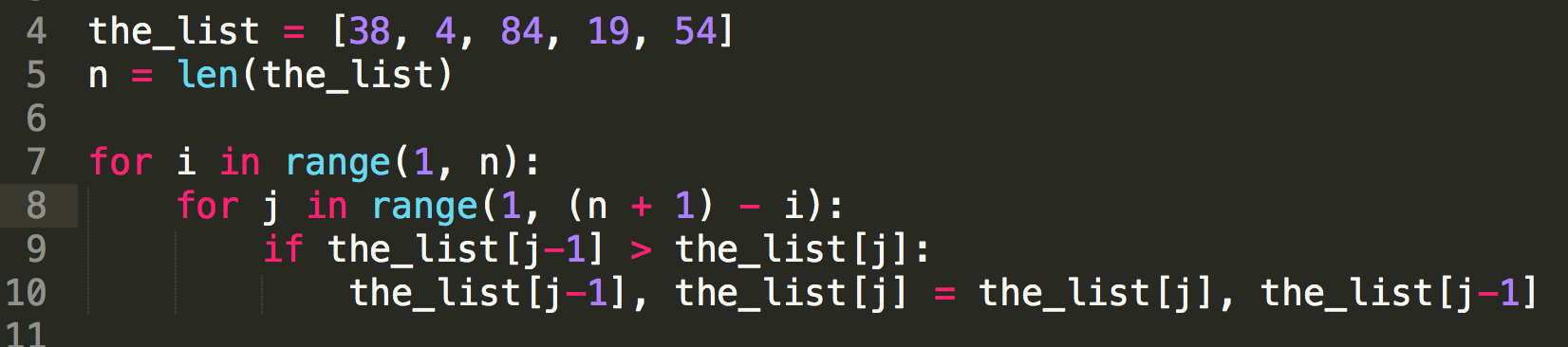
Here is a picture of our code.



Let’s start by analyzing the if block. Let’s assume that the if block all together is 1 operation. This is not accurate, but it is a permissible estimation.

We know that the if block run is inside of the for-loop, for j in range(1, (n + 1 ) – i ) times. Therefore, we run 1 operation from j = 1 to n – i times. Why to n – i and not to (n + 1) - i as indicated by range(1, (n + 1) - i)? Remember that when you use the range function with arguments range(a, b), it will only give you a list of numbers from a UP UNTIL b. So, to get a list of numbers from a to b, we need to add 1 to b. Therefore, to use the range() function to properly get a range from 1 to n – 1, we needed to add 1 to n before subtracting i.

Let’s use sigma notation to represent the summation of 1 operation from j = 1 to n – i times.

Because we know that n equals 5, the total amount of elements in our list, we can replace n – i with 5 – i.

Next, we know that the for-loop mentioned above, for j in range(1, n – I + 1), is inside of another for-loop, for I j in range(1, n). This for-loop runs from i = 1 to n – 1 times, again remembering than range(1, n) will give up from 1 up until n, or in other words from 1 to n – 1. And again, because our n = 5, it will can substitute i = 1 to n – 1 with i = 1 to 5 – 1 = 4 times.

Nesting our first summation representing the inner for-loop, inside of the second summation representing our outer for-loop, we get

Calculating the second sigma notation, we get Leaving us with

Using equivalences, we know that

and

**Therefore, the total number of steps is 10.**