Approach

The code below simply loops through the array taking an element "x" as the pivot. Using this pivot, we count separately the number of values:

1. **Lower** than "x" to the **left** (lo_L)
2. **Higher** than "x" to the **left** (hi_L)
3. **Lower** than "x" to the **right** (lo_R)
4. **Higher** than "x" to the **right** (hi_R)

These values are then combined to update our result variable:

- result += lo_L * hi_R + hi_L * lo_R

Detailed explanation of the formula lower_left * higher_right + higher_left * lower_right :

The formula comes from the fact that we need to create triplets with the form a<b<c ora>b>c. Since b can be regarded as the pivotal point dividing the sequence, we notice that we can form triplets by choosing:

1. For a<b<c, choose any number a (to the left) lower than b, and any number c to the right which is higher. The total number of combinations for this case is lower_left*higher_right.
2. For a>b>c, reverse the previous process. Choose any number a to the left higher than b, and any subsequent number c which is lower. This creates a total number of combinations higher_left*lower_right.

As you can see, summing the combinations for both alternatives gives us the formula total = lower_left*higher_right + higher_left*lower_right. The rest of the code is just book-keeping, and choosing good Data Structures to handle the problem easily.