The code that compares and eventually swaps two elements is inside two loops.
The outer one executes from 1 to n, the inner one goes from 1 to n on the first iteration, from 1 to $n-1$ on the second and so on.

The code inside the loops is thus executed $n + (n-1) + (n-2) + ... + 2 + 1$ times. That is $\frac{n(n+1)}{2} = \frac{n^2-n}{2}$. Consequently, the code has $O(n^2)$ time complexity.