

Analysis: Reversort

The pseudocode of the solution is almost given in the statement. We just need to give it the formal shape in the programming language of our choice. There are two non-straightforward statements in the provided pseudocode. We assume we store L in an array, so we have quick access to any index within it.

The first one is to figure out the index of the minimum element in a contiguous subarray. There may be some library functions to perform this task. For example, in C++ we can use [`min_element`](#), in Python we can use [`index`](#) and [`min`](#) to figure out the minimum element. We can also run another loop to find out the minimum element in the subarray. Please note, the input numbers are all different so the minimum in each iteration is unique.

The second one is to reverse a subarray. Again, there may be some library functions to perform this task. For example, we can use the [`reverse`](#) STL library function for C++, [`reversed`](#) or [`reverse`](#) or simply slicing in Python. We can also run a loop to reverse the subarray.

The length of the subarray we are reversing in the second step above is the cost of the reversal. Accumulating these costs will give our final answer.

This solution has the time complexity of $O(N^2)$. We are running an outer loop from 1 to $N - 1$. Inside the loop we are performing two steps that take linear time each: minimum finding and array reversal. Hence the time complexity is $O(N^2)$.

One final note, there are solutions to this problem that run in $O(N \log N)$ time, but they are a lot more complex. Do you want to try to find one?