**Maximum Sub Array Iterative**:

Fundamental principle of this algorithm is that, only a sum that is positive contributes to the maximum sum of sub array. There can be negative values in that sum but as long as the sum of the sub array is increasing we can add them. So every time a sum goes less than zero we discard the sub array and start a new sub array. At every time we check if the current sub array has the maximum sum and if it has then we replace the maximum sub array with the new one. In the implemented code the first for loop does this. But this fails when all the numbers of the array is negative. For this reason a second for loop is executed only when all the numbers in the array are negative.

* In best case, where array has at least one positive number only one for loop executes. So T(n) = O(n).
* In worst case, where all the numbers are negative in the array both the loops will execute. So T(n) = O(n) + O(n) = O(n)

Therefore from the above two points, T(n) = O(n).