Correctness of MSS Divide & Conquer Algorithm

* The algorithm uses divide & conquer approach to find the maximum subarray.

* So an array of size N is divided into 2 sub arrays as per the concept, however, if the size is one then the mid value is not calculated and in first if condition ie; when low_index = high-index the only element is returned as the maximum sub array.

* The looping statement written always makes sure that the array is scanned from the left index to the right index.

* The algorithm here uses two loops to calculate the maximum sum. The whole array of size 'N' is divided into left sub array [0...m] & right sub array [m+1, n] when m is the mid value in the array. The left sub array is scanned from 0 to m-1 and the second loop right sub array scans the list from m+1 to n (ie; highest index)

* The loop invariants in these cases are maximum sum of the array which are calculated while dividing the problem. The sum is calculated for each of the elements in the

array and they are compared with previous sum, right sub array sum, left sub array sum and the maximum sum of the array is returned.

* The termination condition for both the loops is when the left or the current index reaches the m-1 position and the right index (iteration operator) reaches i^{th} position of the array.

* As the algorithm shouldn't terminate when high negative numbers are used we use -inf or $-\infty$ as the range for the right & left sums.