**DIVIDE AND CONQUER Design Strategy:**

Divide the given array into 2 halves, recursively, and chose one that yields, the maximum subarray from the below:

1. **From Left Half Array: (Recursive Call)**
   1. Find Maximum subarray sum from left half
   2. Start index for max subarray sum
   3. End index for the maximum sum
2. **From Right Half Array: (Recursive Call)**
   1. Find Maximum subarray sum from right half
   2. Start index for max subarray sum
   3. End index for the maximum sum
3. **Crossing Sub Array:**
   1. Find Maximum subarray sum such that, it’s crossing the midpoint of given array from left half array to right half array
   2. Start Index for the max sum
   3. End Index for the max sum

**Points 1 & 2** are recursive calls, where we split the arrays further into left and right halves, until single element, and then start finding the max subarray sums along with starting and ending indexes, based on the above discussed Divide and Conquer strategy.

**Crossing sub array,** is the case when we will have a max subarray, that crosses from left array to right array connecting the mid-point.

In this case

1. we start calculating max sum from mid index to low index, and
2. For right array, we start calculating sum starting mid+1 to end index of the array.
3. And, finally combine the result from above points, 1 & 2 and return.

LINEAR APPPROACH FOR THE SAME PROBLEM IS DISCUSSED IN BELOW PAGE

**Linear Approach:**

We use two loops, in linear approach, to find max subarray, that is resulting max profit, with below thought process.

* With outer loop, iterating all the elements of the input array, one after the other
* Inner loop, finding accumulated sum, one index after the other and finding the maximum sub array sum with chosen index in outerloop.

**Below steps, explain the loop in more detail:**

1. Start from first index value.
2. Calculate the cumulative sum for the rest of the array index values, noting the index returning max sum.
3. Repeat steps 1 & 2, choosing next index value for next run, and then calculating sum with rest of the array values and identifying indexes returning max sum.
4. Finally, return the array indexes resulting maximum subarray from given array.